

А.Л. КОРОЛЕВ, Н.Б. ПАРШУКОВА

**КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ
ОБЪЕКТОВ, ПРОЦЕССОВ И СИСТЕМ**

УЧЕБНОЕ ПОСОБИЕ

Министерство просвещения РФ
Федеральное государственное бюджетное образовательное
учреждение высшего образования «Южно-Уральский
государственный гуманитарно-педагогический университет»

А.Л. КОРОЛЕВ, Н.Б. ПАРШУКОВА

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ОБЪЕКТОВ, ПРОЦЕССОВ И СИСТЕМ

УЧЕБНОЕ ПОСОБИЕ

Челябинск

2020

УДК 681.4(021)

ББК 32.973:2-018я73

К 68

Королев, А.Л. Компьютерное моделирование объектов, процессов и систем: учебное пособие / А.Л. Королев, Н.Б. Паршукова. – Челябинск: Изд-во Южно-Урал. гос. гуманитар.-пед. ун-та, 329 с.

ISBN 978-5-907409-15-6

Пособие посвящено одной из важных информационных технологий – компьютерному моделированию процессов, объектов и систем. Предназначено для студентов бакалавриата, обучающихся по следующим специальностям:

Направление: 44.03.05 «Педагогическое образование» по профилям: «Информатика-Английский язык», «Математика-Информатика», «Физика-Информатика», «Информатика-Математика» при изучении курсов «Компьютерное моделирование», «Виртуальные лаборатории в школьном курсе информатики».

Направление: 09.03.02 «Информационные системы и технологии», профиль: «Информационные технологии в образовании» при изучении курсов «Моделирование систем», «Моделирование объектов, процессов и систем», «Автоматизированные лабораторные комплексы».

Цель пособия – формирование у студентов способности разрабатывать компьютерные модели процессов и систем на основе современной методологии моделирования с использованием современных технологий и основных естественнонаучных законов, а также проводить модельные исследования и компьютерные эксперименты в области профессиональной деятельности. Пособие содержит теоретические положения моделирования, материалы для лабораторных работ по построению различных типов моделей. Практическая часть курса построена на доступном для образовательных целей программном обеспечении.

ISBN 978-5-907409-15-6

Рецензенты:

В.Л. Дильман, д-р физ.-мат. наук, доцент

Г.Б. Поднебесова, канд. пед. наук, доцент

© А.Л. Королев, Н.Б. Паршукова, 2020

© Издательство Южно-Уральского государственного гуманитарно-педагогического университета, 2020

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	(Королев А,Л,)	5
ГЛАВА 1.	ТЕОРЕТИЧЕСКИЕ ОСНОВЫ МОДЕЛИРОВАНИЯ (Королев А.Л.)	9
1.1.	Исторический обзор развития моделирования	9
1.2.	Основные понятия моделирования	13
1.3.	Роль моделирования в науке и технике	21
1.4.	Особенности компьютерного моделирования	25
1.5.	Системный подход в моделировании	39
1.6.	Устойчивость систем	45
1.7.	Общая схема построения модели	46
1.8.	Адекватность моделей	48
1.9.	Классификация моделей	51
1.10.	Информационные модели	53
1.11.	Контрольные вопросы к главе 1	56
ГЛАВА 2.	МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ (Королев А.Л.)	58
2.1.	Введение в математическое моделирование	58
2.2.	Примеры построения математических моделей	68
2.3.	Методы исследования моделей, численное моделирование	79
2.4.	Модели процессов с распределенными параметрами	90
2.5.	Компьютерный вычислительный эксперимент	102
2.6.	Формализация и моделирование	105
2.7.	Контрольные вопросы к главе 2	108
ГЛАВА 3.	МОДЕЛИРОВАНИЕ СИСТЕМ (Королев А.Л.)	109
3.1.	Моделирование сложных систем	109
3.2.	Структурные модели	115

3.3.	Имитационное моделирование	125
3.4.	Моделирование стохастических процессов	139
3.5.	Моделирование систем массового обслуживания	160
3.6.	Клеточные автоматы и агентное моделирование	166
3.7.	Регрессионные модели и корреляционный анализ	175
3.8.	Планирование экспериментов	183
3.9.	Оптимизационные модели	190
3.10.	Геометрические компьютерные модели и САПР	201
3.11.	Моделирование систем управления	229
3.12.	Контрольные вопросы к главе 3	238
ГЛАВА 4.	МОДЕЛИРОВАНИЕ В СРЕДЕ RAND MODEL DESIGNER И ANYLOGIC	240
4.1.	Моделирование с RMD (Королев А.Л.)	240
4.2.	Проект RMD «Непрерывная модель» (Королев А.Л.)	246
4.3.	Проект RMD «Гибридная модель» (Королев А.Л.)	264
4.4.	Инструментальная система моделирования AnyLogic (Королев А.Л.)	276
4.5.	AnyLogic модели системной динамики (Паршукова Н.Б.)	289
4.6.	AnyLogic дискретно-событийная модель (Паршукова Н.Б.)	320
4.7.	AnyLogic агентная модель (Паршукова Н.Б.)	334
4.8.	Контрольные вопросы к главе 4 (Паршукова Н.Б.)	346
	БИБЛИОГРАФИЧЕСКИЙ СПИСОК (Королев А.Л.)	323

ВВЕДЕНИЕ

Моделирование является общенаучным методом изучения законов окружающего мира, свойств объектов и систем самой различной природы или новых объектов и систем. Моделирование как метод – мощный инструмент науки и техники. Как показывает опыт, активное участие в моделировании вырабатывает более глубокое понимание сути протекающих процессов и наблюдаемых явлений.

Развитие компьютерных технологий предоставляет в профессиональной деятельности новые возможности с максимальной степенью наглядности и оперативности получить и представить информацию о свойствах объектов и характере протекающих в них процессов. Применение компьютерных методов моделирования, проведение компьютерных экспериментов способствуют углублению и расширению знаний о процессах, протекающих в конкретных системах или объектах, существующих или проектируемых.

Долгое время достаточно сильным препятствием в этом направлении была необходимость создания моделей средствами какой-либо системы программирования. В этом случае собственно моделирование отодвигалось на второй план, так как разработка модели путем непосредственного программирования с получением в итоге программного комплекса требует значительных затрат времени и средств.

В этом смысле компьютерное моделирование на основе специализированных инструментальных программных комплексов предоставляет возможность построить процесс моделирования, который будет принципиально отличаться тем, что модель создается средствами быстрой разработки, визуальными методами, с автоматическим выбором численных методов и генерацией программы. Это позволяет использовать технологию компьютерного моделирования в учебном процессе по ряду дисциплин профессионального и есте-

ственно-научного циклов. Таким образом, инструментальные программные комплексы моделирования, дающие возможность конструирования моделей с наглядным представлением результатов при минимальной потребности в программировании, имеют особую ценность.

Визуализация – уникальная возможность компьютерной технологии моделирования, так как показать невидимое явление способны только компьютерные модели. Моделирование составляет неотъемлемую часть современного образования, причем по важности моделирование приобретает первостепенное значение. Термин «моделирование» в большинстве случаев означает «компьютерное моделирование», так как применение компьютеров существенно расширило возможности и породило новые технологии.

Цель настоящего пособия – формирование представления о современных методах построения, реализации и исследования моделей объектов, процессов и систем разнообразной природы; расширение представления студентов о моделировании как о методе научного познания; знакомство с методологией моделирования; обучение применению компьютера как средства познания в различных областях практической деятельности; выработка умения применять методы моделирования для решения конкретных задач в области моделирования процессов и систем различной природы. Таким образом, пособие позволяет решить следующие задачи подготовки специалистов:

- познакомить с современными методами и технологиями построения моделей и проведения модельных экспериментов средствами специализированных программных комплексов;
- обучить эффективному применению моделирования и модельного эксперимента;
- развить творческий потенциал будущего специалиста, необходимый для дальнейшего самообучения в условиях непрерывного развития и совершенствования информационных технологий.

В результате изучения представленного в пособии материала, студенты должны быть способны использовать основные законы естественнонаучных дисциплин в профессиональной деятельности, применять методы математиче-

ского анализа и моделирования, теоретического и экспериментального исследования, проводить моделирование процессов и систем. Результатом формирования компетенции в области моделирования будет являться готовность к участию в постановке и проведению экспериментальных исследований; способность обосновывать правильность выбранной модели, сопоставляя результаты экспериментальных данных и полученных решений; способность использовать математические методы обработки, анализа и синтеза результатов профессиональных исследований; способность применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности, а так же применение математических моделей, методов и средств проектирования при моделировании систем.

Пособие будет полезно студентам направлений: 44.03.01 Педагогическое образование, профиль «Информатика»; 44.03.05 Педагогическое образование, профиль «Информатика (с дополнительной специальностью)»; 09.03.02 Информационные системы и технологии.

Настоящее пособие является дополненным и переработанным продолжением учебных пособий:

1. Королев, А.Л. Компьютерное моделирование. Москва: ЛБЗ-БИНОМ, 2010. – 232 с.
2. Королев, А.Л. Компьютерное моделирование технических систем. Челябинск: Изд-во ЧГПУ, 2009. – 170 с.

За время, прошедшее с момента написания данных пособий, изменились требования по составу компетенций в курсах «Компьютерное моделирование» и «Моделирование систем» и появились новые программные комплексы и технологии. Таким образом, переработка и обновление учебных пособий становятся актуальной задачей.

ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ МОДЕЛИРОВАНИЯ

1.1. ИСТОРИЧЕСКИЙ ОБЗОР РАЗВИТИЯ МОДЕЛИРОВАНИЯ

Моделирование имеет многовековую историю, это неотъемлемая сторона человеческой деятельности. По существу, история науки и техники – это история развития моделирования явлений, процессов объектов и систем. Моделирование как познавательный приём неотделимо от развития знаний. Таким образом, понятия «**моделирование**», «**модель**» играли жизненно важную роль в деятельности человечества с тех пор, как оно стало стремиться к пониманию и изменению окружающей среды, созданию новых объектов.

Моделирование как форма отражения действительности зарождается вместе с возникновением научного знания и развития технологий. Модели и моделирование начинают широко использоваться в эпоху Возрождения. Так итальянские архитекторы того времени пользовались моделями проектируемых сооружений. Уже в XIX–XX вв., а тем более в XXI в. трудно назвать область науки, техники и технологий или их приложений, где моделирование не имело бы существенного значения.

Естественно, что моделирование первоначально носило материальный и наглядный характер: рисунок, макет, схема, чертеж и т.п. Возможность и необходимость представлять знания в виде моделей была понята только с течением времени. В истории моделирования можно выделить три направления. **Первое направление** связано с технологией литья в формы. **Второе направление** связано с материальными макетами объектов, которые использовались при решении архитектурно-строительных и технических задач. Понятие масштабной материальной модели известно архитекторам еще до нашей эры. **Тре-**

тье направление образовано научными знаниями, для осознания модельного характера которых потребовалось двадцать с лишним веков развития науки.

Использование материальных моделей как инструмента для решения технических и технологических задач началось еще в глубокой древности. Создаваемые материальные модели служили средством для разработки новой и усовершенствования существующей техники. Они сыграли важную роль в переходе к машинному производству. В этот период моделирование становится средством и способом поиска рациональной конструкции устройств. Такого рода модели были основой творчества мастеров-изобретателей, усилиями которых создавалась техническая база перехода к более совершенным способам производства.

Научная составляющая подобных технических моделей была крайне малой в силу слабого развития науки и ее сословного отрыва от технической практики. Например, основой творчества Джеймса Уатта был, прежде всего, модельный эксперимент. Научных представлений о процессах в паровой машине еще не существовало.

По мере развития науки и расширения ее технологического применения модели постепенно наполнялись теоретическим содержанием и из наглядных пособий превращались в средство для применения научных методов в конкретных прикладных задачах. Начиная, с 70-х гг. XIX в. материальные модели становятся составной частью науки. В целом ряде областей научного знания и его практических приложений они сохраняют большое значение и в настоящее время.

Активное применение науки в технической практике в ходе промышленной революции привело к насыщению модельных экспериментов теоретическим содержанием. Выявленные наукой законы становились исходными данными при постановке конкретного модельного эксперимента, направленного на решение определенной технической задачи, и применялись при обработке результатов экспериментов.

Использование результатов научных исследований в технике и технологии обозначило **два класса моделей**:

- модели и теоретические зависимости, полученные на основе научных законов и представленные математическими формулами;

- модели, полученные опытным путем, представленные в форме таблиц, графиков и эмпирических зависимостей.

Теоретические и эмпирические модели дополняют друг друга, обеспечивая достижение главной цели научно-технической деятельности – создание новых технических средств и технологических процессов с заранее заданными свойствами. Совокупность таких моделей, относящихся к некоторому классу технических задач, совершенствовались, дополнялись инженерными методиками и приобретали статус самостоятельных инженерных **научных дисциплин**. Примерами являются такие науки, как сопротивление материалов (теория прочности), термодинамика, гидро-и аэродинамика и др.

Примечательно, что в практике кораблестроения моделирование впервые получило твердую научную основу в виде **теории подобия**. При проектировании пароходов одной из важных задач является снижение сопротивления движению судна. От решения этой задачи зависит количество необходимого кораблю запаса топлива, мощность, масса двигателя и скорость хода корабля. Прямое опытное решение этой задачи невозможно по очевидным экономическим причинам. Не поддается эта задача и теоретическому решению, несмотря на существование развитой гидродинамической теории. Подобные задачи с требуемой точностью могут быть решены только современными весьма мощными вычислительными комплексами на основе достаточно сложных численных методов и математических моделей.

Выход был найден путем соединения теории и экспериментального исследования малых геометрически подобных моделей кораблей. **Теория подобия** позволила выполнить перенос результатов испытаний моделей кораблей на проектируемые суда. До сих пор в Санкт-Петербурге существует и используется бассейн для исследования сопротивления при движении корпуса судна даже в условиях воздействия волн.

Экспериментальное исследование гидродинамических характеристик кораблей с помощью масштабных моделей стало первым в истории примером применения научной методологии моделирования и стимулировало развитие

теории подобия применительно к целому ряду явлений разнообразной физической природы.

Например, прочностной расчет сложных конструкций мостовых ферм был невозможен, т. к. вычислительной техники и соответствующих вычислительных методов еще не существовало. Однако можно было построить уменьшенную модель мостовой фермы и провести ее прочностные испытания. Теперь возникла другая проблема: как перенести результаты модельных исследований на реальный объект. Эту проблему успешно и решала **теория подобия**.

В последней четверти XIX в. возникло другое направление моделирования – **аналоговое моделирование**. В этом случае свойства и характеристики некоторого объекта воспроизводятся с помощью модели иной, чем у оригинала физической природы. По мере математизации естественных и инженерных наук стало очевидным, что целый ряд явлений различной природы описываются совпадающими (аналогичными) по форме математическими моделями. Это позволяет исследовать свойства моделей как самостоятельных абстрактных объектов, что привело к развитию прикладных направлений математики. Примером может быть математическая физика, теория устойчивости, теория колебаний и т.п.

С развитием науки и техники все большее значение приобретали **вероятностно-статистические** модели, которые оказывались более адекватными многим реальным явлениям. Развитие теории автоматического регулирования, теории информации и кибернетики позволило уточнить понятие «**модель**», независимо от того, реализована она материально или представляет собой некоторый идеальный объект. В любом случае смысл и суть существования модели заключается в том, что она несет в себе информацию о свойствах объекта-оригинала, существенных с точки зрения решаемой задачи.

В настоящее время от моделирования процессов и явлений происходит переход к моделированию **знаний**, т.е. к моделированию логического вывода новых знаний на базе уже имеющихся. Методология моделирования и формализации знаний, ориентированная на их компьютерную обработку, является одним из основных направлений развития искусственного интеллекта.

Даже небольшой экскурс в историю показывает, что моделирование прошло огромный путь развития, от сформулированных на естественном языке непосредственно наблюдаемых закономерностей реальности до сложнейших имитационных, математических и интеллектуальных систем, опирающихся на возможности современной вычислительной техники.

Таким образом, моделирование постепенно распространялось на все новые области научно-технических знаний: техническое конструирование, строительство и архитектуру, астрономию, физику, химию, биологию и общественные науки. Компьютерная техника существенно расширила сферу применения методов моделирования, породила принципиально новые возможности, виды моделей и целые технологии. Поэтому в настоящее время понятия «модель», «моделирование» неявно отождествляются с **компьютерными моделями** и **компьютерным моделированием**. Этим и объясняется название данной работы и содержание рассматриваемых в ней вопросов.

1.2. ОСНОВНЫЕ ПОНЯТИЯ МОДЕЛИРОВАНИЯ

Рассмотрим основные понятия моделирования.

Объект моделирования (объект-оригинал) – некоторая часть окружающего нас мира, реальной действительности (объект, процесс, явление), которая может быть рассмотрена как единое целое.

Под **моделью** мы будем понимать объект произвольной природы, отражающий свойства, характеристики и связи моделируемого объекта (объекта-оригинала), которые считаются существенными для решения поставленной задачи.

В идеальном случае модель отражает только главные свойства объекта моделирования и не отражает несущественные свойства (с точки зрения решаемой задачи). Однако модель, как самостоятельный объект, имеет и свои собственные свойства, которые никак не связаны с моделируемым объектом.

Главное назначение модели состоит в упрощении получения информации о свойствах объекта-оригинала для решения конкретных задач практической деятельности. Полное соответствие модели оригиналу невозможно по

определению (рис. 1.1). В этом случае теряются все преимущества моделей, так как мы будем иметь второй экземпляр объекта-оригинала с малодоступным для изучения бесконечным набором свойств. В ряде случаев требуется построить модель системы, существующей в единственном экземпляре.

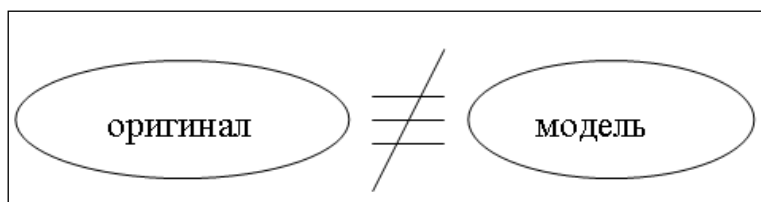


Рис. 1.1. Соотношение между моделью и оригиналом

Самое распространенное представление о модели связано с **материальной моделью**, например, с макетом. Однако моделью является экспериментальная установка, словесное описание объекта-оригинала, его мысленный образ и т.д.

Понятие модели включает в себя следующие основные компоненты: объект-оригинал; решаемая задача; характер отражаемых свойств объекта-оригинала; способ построения и способ реализации модели. Решаемая задача является одним из главных элементов, определяющих характер создаваемой модели. Именно задача определяет набор существенных свойств моделируемого объекта, вид и способ построения модели (рис. 1.2). Без связи с конкретной задачей понятие модели не имеет смысла.

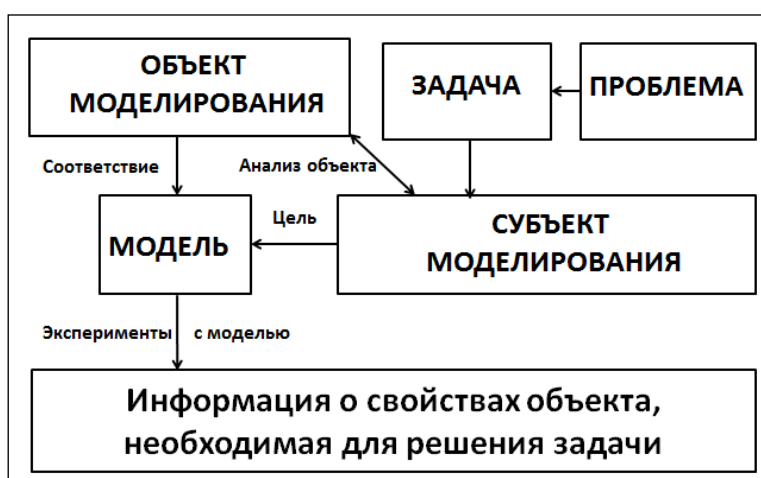


Рис. 1.2. Процесс моделирования

Ряд факторов определяет **множественность** моделей, используемых для описания и исследования объекта:

1. Любой объект имеет бесконечное количество свойств. Следовательно, для его разностороннего изучения необходимо построить множество моделей, каждая из которых будет отображать определённую группу его свойств.

2. Для одного и того же объекта, для отображения одних и тех же свойств можно построить множество моделей разными способами в зависимости от целей моделирования и доступных средств.

3. Так как созданием моделей занимается человек, то построенная модель зачастую существенно зависит от его субъективных предпочтений.

4. В зависимости от решаемой задачи для одного и того же объекта, одним и тем же способом, для отображения одних и тех же свойств можно построить множество моделей с разной степенью детализации описания свойств объекта.

В любой деятельности по созданию новых объектов, систем или технических устройств всегда имеется образ будущего объекта. Этот образ первоначально задается в виде технического задания на проектирование. Затем в ходе проектирования создается модель новой технической системы, которая позволяет получить информацию о свойствах еще не существующего объекта. Этим обеспечивается создание объекта с заданными (желаемыми) свойствами. Можно сказать, что моделирование является обязательным элементом во всякой целесообразной деятельности.

Условно можно разделить модели на две группы: познавательные и прагматические. **Познавательные модели** являются формой представления знаний. Поэтому при обнаружении расхождения между моделью и реальностью возникает задача устранения этого расхождения с помощью изменения модели.

Познавательная деятельность ориентирована в основном на приближение модели к реальности, которую модель отображает (рис. 1.3). Познавательные модели широко используются в естественных науках. Примером познавательной модели может служить модель солнечной системы.

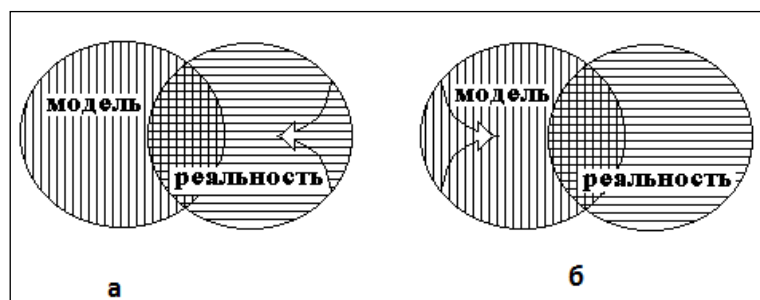


Рис. 1.3. а – познавательная модель; б – прагматическая модель

Прагматические модели являются средством организации практических действий, являются отображением цели. Примерами прагматических моделей могут служить проекты, рабочие чертежи и т.п. Прагматические модели носят нормативный характер, играют роль стандарта, образца, под который «подгоняется» результат деятельности. Действительно, при создании нового объекта (например, строительство здания) отклонение от проекта считается недопустимым.

Рассмотрим основные **функции** моделей в современной науке, технике и практической деятельности. Еще раз подчеркнем, что главное назначение моделей – упрощение получения информации об объекте моделирования. Вместе с тем модели выполняют и ряд других важных функций:

- познавательная функция (получение новых знаний, познание законов функционирования объекта);
- передача информации и формирование знаний;
- решение задач оптимизации и управления состоянием объекта;
- создание объектов с заданными свойствами;
- диагностика состояния объекта;
- прогнозирование поведения объекта или развития процесса;
- имитация систем и создание тренажеров;
- разработка игровых моделей и моделей обучения.

В целом моделирование – это общенаучный метод изучения свойств объектов и процессов по их моделям, используемый в целях познания, исследования, проектирования, принятия решений.

Процесс моделирования можно представить в виде нескольких этапов: реальный объект → построение модели → изучение модели → получение зна-

ний об объекте на основе модельных экспериментов → перенос результатов моделирования на реальный объект.

Термин «моделирование» используется также и для обозначения собственно процесса построения модели. В этом смысле он понимается как технология построения моделей. Первоначально, при создании модели, от нее в первую очередь требуется **отображение свойств** объекта моделирования, актуальных для решения задачи, впрочем, эти свойства необходимо еще установить. Далее, при экспериментах с моделью возможен **прогноз** поведения объекта-оригинала в иных ситуациях по сравнению с теми, на основе которых модель создавалась. При этом сведения, полученные в ходе моделирования, т.п. исследования модели, объективно представляют собой информацию о свойствах самой модели, поэтому далее они должны быть перенесены на оригинал на основе определенных правил перехода с целью предсказания свойств уже самого объекта моделирования.

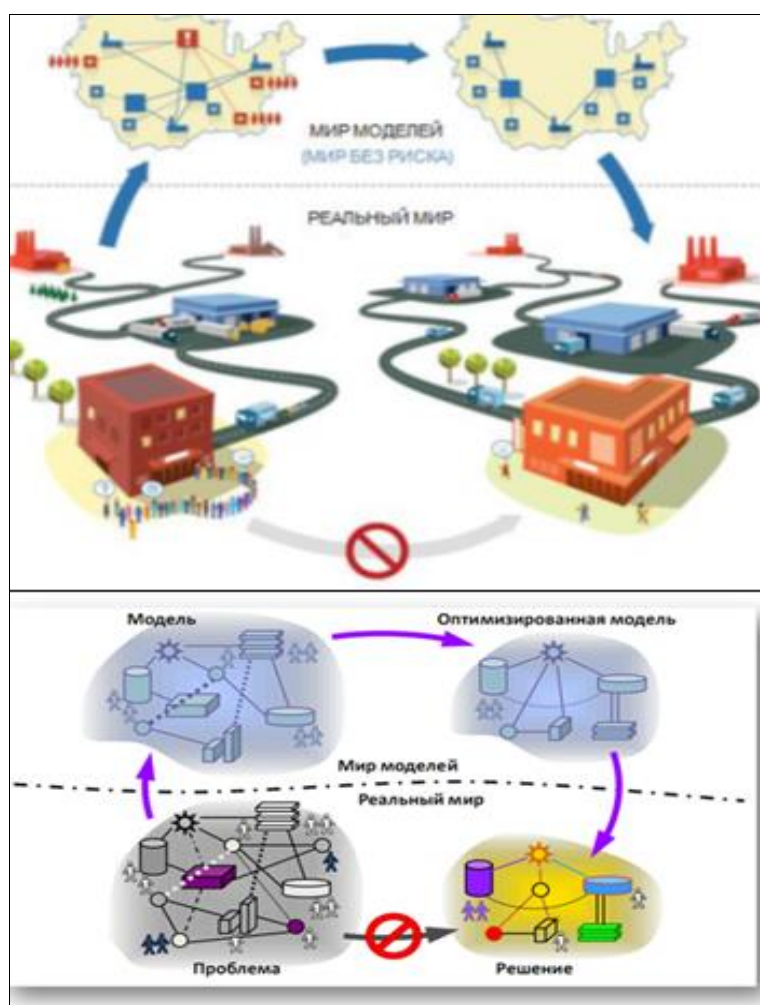


Рис. 1.4. Применение моделирования

Некоторые исследователи совершают методологическую ошибку, приписывая объекту свойства модели, адекватность которой еще требуется установить.

В принципе существует два пути исследования любого объекта или процесса: прямое изучение объекта-оригинала путем экспериментов с ним или изучение объекта по его модели. Естественно, что во многих случаях используется и моделирование, и прямое изучение объекта моделирования, а в некоторых случаях моделирование является единственным способом.

Моделирование особо актуально в следующих случаях:

1. Объект либо ещё, либо уже не существует, однако необходимо изучить его свойства. Так, например, методами компьютерного математического моделирования была реставрирована картина развития процессов при падении Тунгусского метеорита, а на основе анализа результатов моделирования были качественно объяснены все последствия падения космического тела и определены его параметры. Модель изменения климата вследствие ядерного конфликта убедительно доказала реальную возможность развития глобальной экологической катастрофы (ядерная зима) и неприемлемость применения ядерного оружия даже в ограниченном варианте. Как уже отмечалось ранее, любое производство предусматривает проектирование. Результатом проектирования является модель будущего объекта, который после создания должен обладать определенными свойствами.

2. Масштаб времени реальных процессов или геометрические размеры объекта несоизмеримы с возможностями нашего восприятия. Моделирование позволяет изучать и весьма быстрые (химические и ядерные реакции), и достаточно медленные процессы (старение материалов, экологические процессы, движение материков и т.п.), микро- и макроскопические объекты (модель атома, модель планеты Земля и модель солнечной системы).

3. Реальный объект или процесс недоступен для прямого изучения. Средства наблюдения (измерения) способны существенно исказить естественный ход событий, либо прямые измерения в принципе невозможны. Такие ситуации возможны и в технике, и в фундаментальных науках, и в социально-экономической сфере.

4. Эксперименты с реальным объектом дороги или слишком опасны.

Подобная ситуация имеет место в экономических и социальных науках, где эксперименты с реальными системами могут привести к необратимым катастрофическим последствиям. Всем известно, чем закончился натурный эксперимент на четвертом энергоблоке Чернобыльской АЭС.

Таким образом, модель служит для получения информации об объекте исследования, которую затруднительно или невозможно получить путем непосредственного исследования оригинала, во многих случаях моделирование – это единственно возможный путь изучения объектов и процессов. А в некоторых случаях испытание реального объекта обязательно, например, испытания новой авиационной техники.

Рассмотрим типовые задачи моделирования.

Прямая задача моделирования. Требуется определить реакцию объекта или его поведение в ответ на известное внешнее воздействие. Решение данной задачи выполняется с помощью проведения модельных экспериментов. Такие задачи возникают в ходе изучения свойств объектов или процессов.

Обратная задача моделирования. Реакция объекта известна или задана, требуется определить, какое внешнее воздействие способно вызвать подобную реакцию. Для решения этой задачи необходимо также иметь уже разработанную модель. Обратные задачи актуальны, например, в управлении: определить, как необходимо воздействовать на объект, чтобы он сохранял заданное состояние или выполнял необходимые действия.

Задача идентификации или синтез модели. Известно внешнее воздействие на объект, известна реакция объекта, требуется определить параметры модели или построить собственно модель. Подобная задача возникает в тех случаях, когда модель в целом разработана, но требуется по результатам испытаний или экспериментов с реальным объектом определить некоторые ее параметры. Если модель еще не разработана, то методами математической статистики и регрессионного анализа можно построить чисто формальную модель объекта типа «**черный ящик**».

Рассмотрев основные понятия моделирования, области его применения, решаемые задачи можно перейти к следующему разделу, посвященному особой роли моделей в науке и технике.

1.2. РОЛЬ МОДЕЛИРОВАНИЯ В НАУКЕ И ТЕХНИКЕ

Цель данного параграфа – показать роль моделей и моделирования в современной науке и технике. Моделирование – это один из основных методов научного познания. Суть этого метода заключается в том, что из сложного явления выделяются некоторые его главные части и замещаются моделями, более понятными, более простыми, более удобными для изучения и объяснения.

В основе любой науки особую роль имеют **концептуальные модели**, т.е. представления об объекте-оригинале, которые сложились в сознании человека. Основой для формирования таких моделей являются и результаты наблюдений свойств объекта-оригинала, и теоретический багаж исследователя, и его опыт, аналогии, логические выводы. Объединение всех этих представлений в концептуальную модель осуществляется неформально. Таким образом, построение концептуальной модели предполагает применение для описания объекта строгих научных понятий и выявление наиболее существенных факторов.

С помощью концептуальных моделей строится первичная система простейших абстрактных моделей, которые отражают свойства реальных объектов, представляющих интерес для данной науки, например: в физике – «материальная точка», «идеальный газ», «абсолютно черное тело»; в геометрии: – «плоскость», «прямая», «точка» и т.д. Концептуальная модель «материальная точка» отражает свойство инерции тела (материальная точка имеет массу) и способность занимать определенное положение в пространстве (материальная точка имеет координаты). Концептуальные модели занимают самый нижний уровень научного знания, но именно они связаны с практической деятельностью и рождаются в ходе ее.

В свою очередь, **научные законы** формулируются как описание связей и взаимодействий между концептуальными моделями. Примером могут служить законы механики Ньютона, законы Кирхгофа, закон Гука и т.п. Таким образом, научные законы также являются в определенном смысле моделями реальности.

Научные законы носят общий характер и занимают более высокий уровень по сравнению с концептуальными моделями. Законы имеют вполне конкретную объектную область. На базе концептуальных моделей и соответствующих законов строятся модели целых классов явлений и процессов, которые образуют **научные теории**.

Итак, с точки зрения моделирования научные теории являются системой концептуальных моделей и законов, описывающих взаимодействие между концептуальными моделями, например: теория относительности, квантовая теория, теория твердого тела, теория колебаний, теория устойчивости и т.п.

Науке давно известно понятие **гипотезы**, которую можно считать моделью реальности в условиях неполной изученности явлений. Наука допускает существование нескольких гипотез, поскольку одни и те же наблюдения могут одинаково хорошо объясняться с различных точек зрения. Как бы хорошо ни описывали процессы и явления, существующие модели, всегда возможно их дальнейшее уточнение, которое постоянно происходит в любой науке.

Проектирование и эксплуатация современных сложных технических систем все больше требует «поддержки» со стороны моделирования. Давно ушли в историю методы проектирования, основанные на опыте и интуиции инженеров и техников. Современные технологии проектирования подразумевают применение научных знаний, математических моделей, 3D-графики, методов оптимизации с целью получения объекта с наилучшими свойствами и т.п. Кроме того, техническая документация проекта (чертежи) создается программными средствами автоматизированного проектирования в электронном виде.

Как уже отмечалось, применение моделирования в ходе проектирования неизбежно, так как оно позволяет в итоге создать объект с требуемыми свойствами. Моделирование позволяет существенно сократить затраты на доработку спроектированного изделия по данным испытаний опытных образцов.

Однако полностью исключить испытания и натурные эксперименты в силу сложности современных технических систем не удастся. Профессия летчика-испытателя в обозримом будущем останется крайне необходимой.

Функционирование современных сложных технических систем требует управления и регулирования режимов их работы. Управление производится компьютерными системами (информационно-измерительными комплексами) на основе моделей объектов управления, которые позволяют учесть возможные взаимосвязи, ограничения, установить оптимальные режимы функционирования.

В ходе эксплуатации технических объектов возможны аварии и неисправности. Для обеспечения высокой надежности технических систем важно вовремя распознать приближение аварийной ситуации. Такая задача решается методами диагностики состояния объекта. Для подобных задач необходимо на основе моделирования аварийных ситуаций получить информацию о состояниях, предшествующих аварии, то есть получить картину динамики развития аварии. Теперь в случае распознавания предаварийного состояния, технический объект может быть своевременно выведен из эксплуатации для проведения ремонта. Например, при работе подшипников турбин или двигателей их состояние можно диагностировать по частоте и амплитуде вибраций, зависящих от степени износа подшипников.

Если авария все же произошла, то модель системы помогает установить ее причины и исключить их в будущем. Так при установлении причин аварии энергоблока на Чернобыльской АЭС была создана модель функционирования энергоблока и установлены возможные причины аварии.

Рассмотренные выше примеры применения моделей показывают, что роль моделирования в современной науке и технике трудно переоценить. Пройденный путь начат с применения макетов и материальных моделей на сегодня продолжен сложнейшими математическими и имитационными компьютерными моделями.

В настоящее время основные функции моделей в технике следующие:

- Получение знаний о закономерностях функционирования объектов или систем существующих или проектируемых.
- Решение задач оптимизации объектов и управления протеканием процессов в системах.

- Создание объектов с заранее заданными свойствами.
- Диагностика состояния, прогнозирование поведения объекта или прогнозирование развития процессов.

- Имитация сложных технических объектов и систем.

Цель моделирования в технике – обеспечение создания системы с заданными свойствами. Эта задача решается в ходе проектирования. **Проектирование** – это создание описания, необходимого для изготовления некоторого объекта или системы в заданных условиях на основе описания этого объекта в виде технического задания (**ТЗ**). Любая система проектируется постепенно, по частям. Возможны различные способы проектирования.

Нисходящее проектирование:

1. Система представляется состоящей из нескольких подсистем. Объект представляется в виде функциональной структуры, состоящей из подсистем.

2. Задачи проектирования решаются для объекта, который состоит из подсистем.

3. Далее формулируются технические задания для проектирования подсистем, выполняется решение аналогичных задач для подсистем (рис. 1.5), агрегатов и узлов.

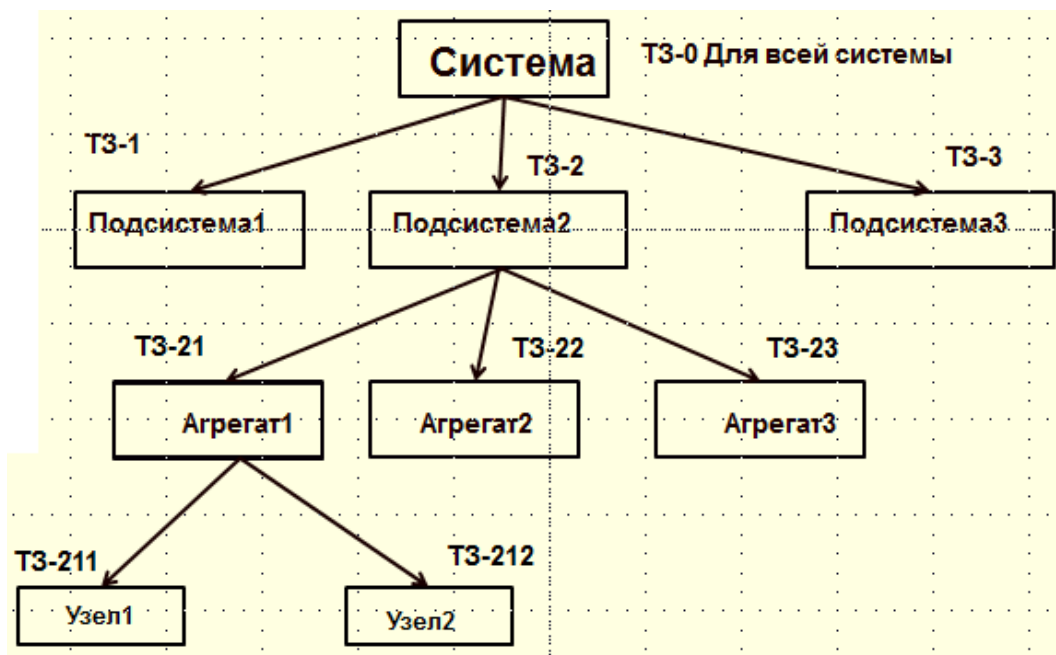


Рис. 1.5. Нисходящее проектирование

Недостаток: на очередном этапе декомпозиции могут быть получены такие технические требования, которые невозможно выполнить при данном развитии техники и технологии.

Восходящее проектирование – проектирование системы из набора типовых элементов (принцип детского конструктора). **Недостаток восходящего проектирования:** неизвестно что получится в итоге и будет ли результат удовлетворять требованиям технического задания.

Обычно оба подхода используются комплексно.

1.4. ОСОБЕННОСТИ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ

В настоящее время методы компьютерного моделирования прочно вошли в практику решения широкого круга теоретических и прикладных задач в различных сферах практической деятельности. Исторически первая роль компьютера – **вычислитель**, т.е. численное решение задач моделирования в первую очередь технических систем.

Сущность компьютерного моделирования состоит в построении модели, которая представляет собой некоторый программный комплекс, алгоритмически описывающий развитие процесса или поведение объекта. **Компьютерная модель** предназначена для проведения с ней экспериментов на вычислительной машине. Она имеет две составляющие: **программную и аппаратную**. Программная составляющая выполняется процессором компьютера и отображает свойства объекта моделирования.

Главной особенностью компьютерных моделей является относительная простота создания и модификации. Изменениям подвергается только сама программа, а аппаратная составляющая (компьютер) остается неизменной. Если добавить практически неограниченную функциональную и структурную сложность компьютерных моделей, высокую точность результатов, то становится ясно, почему в настоящее время под **моделированием** почти всегда понимают **компьютерное моделирование**.

Выделим ряд особенностей компьютерного моделирования:

1. Компьютер – мощный инструмент проведения вычислительных экспериментов, так как позволяет хранить и быстро обрабатывать большие объемы информации. Это существенно увеличило возможности численного решения задач математического моделирования и на порядки сократило время их решения.

2. Компьютерное моделирование позволяет исследовать модели высокой степени сложности, учитывать и анализировать влияние множества факторов. Только применение компьютерного моделирования дало возможность использовать технологии, опирающиеся на тонкие физико-химические эффекты.

3. Применение компьютера в моделировании привело к рождению новых направлений как в самом моделировании (имитационное и стохастическое моделирование, моделирование знаний), так и в различных прикладных науках (вычислительная физика, автоматизированное проектирование и т.п.).

4. Компьютерные модели стали основой математизации ряда областей научного знания и практической деятельности, которые ранее развивались как описательные и носили сугубо качественный характер. Это, в первую очередь, относится к моделированию знаний и технологиям искусственного интеллекта.

5. В ходе компьютерного моделирования возможна визуализация результатов моделирования, отображение их в наиболее обозримой и наглядной форме средствами виртуальной реальности. Такую возможность, например, дает компьютерная технология трёхмерного твердотельного моделирования и анимация.

6. Компьютер не только средство реализации модели и проведения экспериментов, но и инструмент создания самих моделей: предоставляется возможность автоматизированного построения модели, выбора численных методов и создания программы, реализующей вычислительную модель.

Далеко не полный перечень сфер применения компьютерного моделирования в технике, который постоянно расширяется, представляется следующим:

1. Анализ проектируемого объекта и выбор оптимальных решений.

2. Техническая диагностика состояния объекта – распознавание образа состояния объекта и прогноз динамики его изменения, например, с целью предупреждения аварий.

3. Автоматизированное управление объектом.

4. Создание имитаторов и тренажеров.

5. Автоматизированное проектирование технических систем.

Построение модели с использованием компьютерной техники содержит в себе несколько необходимых этапов. На первом формируется теоретическое представление об исследуемом объекте, строится его **концептуальная модель**.

На втором этапе концептуальная модель переводится на формальный язык математики и численных методов, язык описания алгоритмов: создаются **математическая и алгоритмическая модели** объекта.

Переложение модели на язык программирования дает **компьютерную модель**, позволяющую оперировать с цифровой информацией. Наконец, используя эмпирическую информацию о параметрах исследуемой системы, можно получить полноценную компьютерную модель, которая позволяет с той или иной степенью достоверности исследовать свойства объектов, прогнозировать последствия принятых решений, иными словами – проводить модельный **вычислительный эксперимент**.

В настоящее время актуален вопрос создания автоматизированных систем моделирования, которые позволят выполнять быстрое построение моделей, проведение модельных экспериментов, обработку и анализ их результатов. В связи с этим рассмотрим разделение ролей при разработке модели. Естественно, что представленное разделение несколько утрировано, но это не меняет его сути.

Традиционный путь создания реальной компьютерной модели начинается с выявления или синтеза структуры объекта, при этом проводится предварительный качественный анализ его свойств. Постановка всех задач моделирования осуществляется **специалистом в конкретной предметной области** в терминах соответствующего профессионального языка, строится концептуальная модель объекта, процесса или явления.

Далее необходима работа **математика**, который выполняет математическую постановку всех задач, создает описание (модель) объекта средствами языка математики. При этом в диалоге со специалистом-предметником происходит уточнение описания модели, т.к. многие очевидные для предметника факты и предположения, как правило, в его модели отсутствуют. Таким образом, на этом этапе модель формализуется и существенно уточняется математиком. В итоге его работы является логико-математическая модель объекта. Кроме того, математик должен выбрать надежные и эффективные численные методы ее реализации, т.е. преобразовать математическую модель в вычислительную.

На следующем этапе **программист** разрабатывает алгоритмы и программы, реализующие решение задачи моделирования, с учетом возможностей конкретной системы программирования и компьютера. Естественно, что программист должен выполнить отладку программы и предъявить первичные результаты моделирования математику, оценивающему их качество с математической точки зрения.

Общая оценка модели по данным предварительных расчетов должна быть выполнена специалистом-предметником, которому далее и предстоит использовать разработанную компьютерную модель в своей практике.

Если потребуются совершенствование и переработка модели, то весь путь должен быть пройден вновь. Ясно, что данная схема весьма затратна, громоздка и инертна. Следовательно, необходимо **приближение компьютера к специалисту**.

В свое время в XX веке для решения этой проблемы создавались модели знаний конкретной предметной области. На основе этих моделей знаний и проводилось построение модели конкретного объекта. Но такой подход оказался неприемлемым, т. к. необходимо построение модели знаний конкретной предметной области, а это является весьма сложной задачей.

Идея программного комплекса **Simulink**, который входит в состав математического пакета **MatLab**, состоит в том, что модель визуально конструируется в виде блок-схемы из стандартных функциональных блоков (рис. 1.6). Пакет позволяет реализовать модельно-ориентированное проектирование.

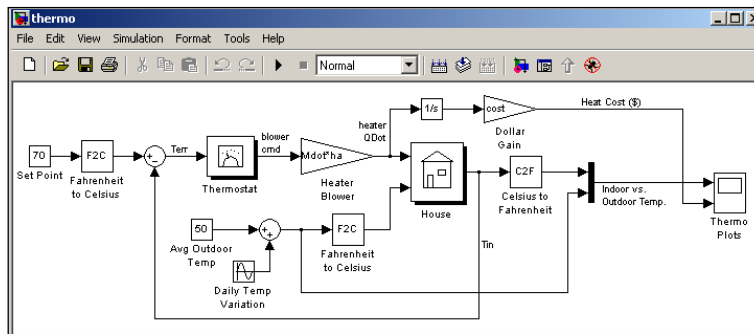


Рис. 1.6. Пример блок-схемы Simulink-модели

Каждый блок является объектом, параметры которого можно менять. Каждый блок реализует определенную функцию преобразования входного сигнала в выходной. Пакет Simulink имеет средства отображения результатов моделирования в графической форме (рис. 1.7). Моделирование сводится к построению **функциональной блок-схемы** объекта путем копирования блоков из библиотеки, установления (рисования) связей между ними и заданием параметров элементов системы.

В данном программном комплексе реализована идея аналоговых вычислительных машин (АВМ), в которых построение модели сводилось к подобным действиям с реальными электронными блоками.

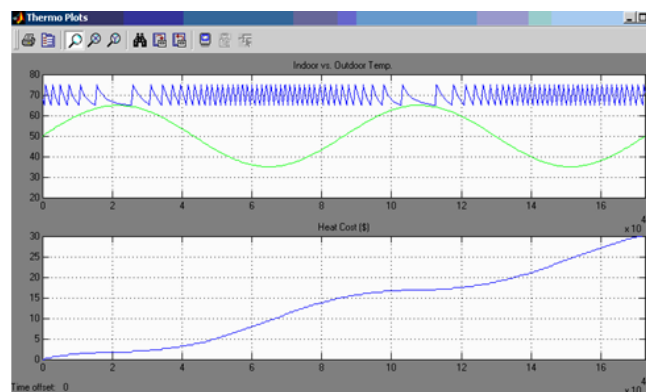


Рис. 1.7. Отображение результатов моделирования средствами Simulink

Пакет Simulink имеет развитую библиотеку стандартных функциональных блоков общего и специального назначения (рис. 1.8) для построения моделей в различных областях техники.

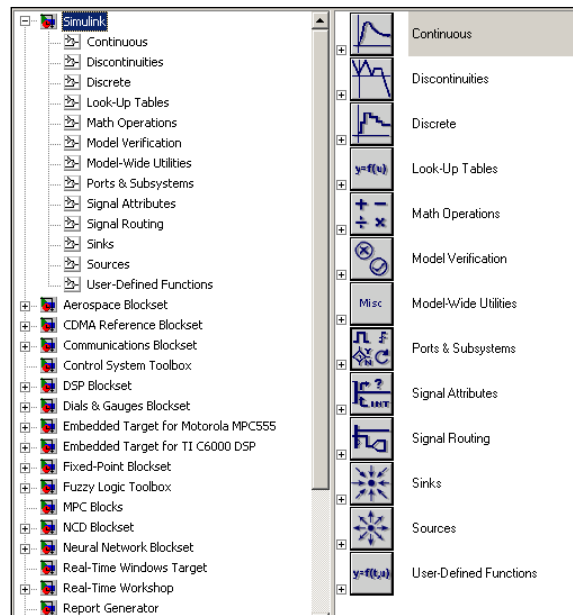


Рис. 1.8. Библиотека блоков Simulink

Кроме того, имеется возможность создавать собственные новые функциональные блоки. Моделирование систем возможно по блочно-иерархическому принципу.

В рамках этого пакета автоматизируются функции и математика, и программиста, а модель объекта в виде блок-схемы может быть построена специалистом в конкретной предметной области.

Программный комплекс **SolidWorks** позволяет построить трехмерную твердотельную модель объекта (рис. 1.9) и провести расчеты, например, температурных полей под действием заданных пользователем воздействий (рис.1.10). При этом пользователь строит только трехмерную модель объекта и определяет внешние воздействия. Построение математической и вычислительной моделей, получение и представление результатов исследования производится полностью автоматически.

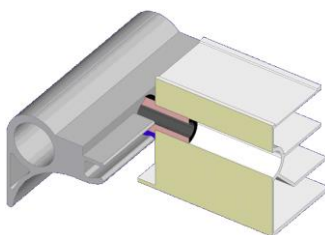


Рис. 1.9. Трехмерная модель узла конструкции

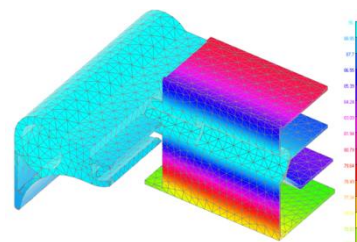


Рис. 1.10. Отображение результатов расчета температурных полей

Аналогичные функции выполняют системы автоматизированного проектирования КОМПАС, T-Flex, ADEM, AutoCAD и другие.

Одной из универсальных систем моделирования является система MVS (Model Vision Studium). MVS – это интегрированная система для быстрого создания интерактивных визуальных моделей сложных динамических систем и проведения с ними вычислительных экспериментов. Разработчики системы позиционируют ее как академическую систему, подобная современная профессиональная система моделирования называется RMD (Rand Model Designer»). Система RMD была разработана как результат развития MVS, вследствие большой популярности MVS.

Главными особенностями MVS и RMD являются использование технологии объектно-ориентированного моделирования; адекватное описание гибридных (дискретно-непрерывных) систем; обеспечение численного решения; визуализация результатов моделирования без традиционного программирования.

Пакеты позволяют выполнять моделирование гибридных систем, которые обладают одновременно непрерывными и дискретными свойствами. При создании MVS и RMD принято, что выбор и настройка математического метода решения должны выполняться системой моделирования. Пользователь имеет возможность активно вмешиваться в ход вычислительного эксперимента.

Rand Model Designer – это высокопроизводительная визуальная среда для разработки компонентных моделей сложных динамических систем. RMD позволяет разрабатывать непрерывные, дискретные и гибридные (непрерывно-дискретные) модели и проводить интерактивные вычислительные эксперименты с разработанными моделями. Пакет предназначен для пользователей, проводящих многочисленные сложные и затратные вычислительные эксперименты, или создающих новые программные приложения, в основе которых лежат математические модели. Продукт сочетает в себе интуитивно понятную среду объектно-ориентированного моделирования и мощные механизмы символьных вычислений, которые позволяют ускорить процесс подготовки качественных моделей.

Основными областями применения продукта являются:

- проведение научных вычислительных экспериментов, в том числе и с использованием внешних приборов и оборудования;
- проектирование технических систем;
- обучение;
- разработка математических моделей физических систем и процессов с последующим включением их во внешние программные приложения.

Любая модель, создаваемая в MVS (RMD), представляет собой проект, содержащий стандартные компоненты (рис. 1.11). Системы позволяют на основе карты поведения (рис. 1.13) наглядно представить и описать логику смены поведения объекта.

Основным элементом при построении модели является устройство (CDevice) – это некоторый объект, функционирующий параллельно и независимо от других объектов в непрерывном времени.

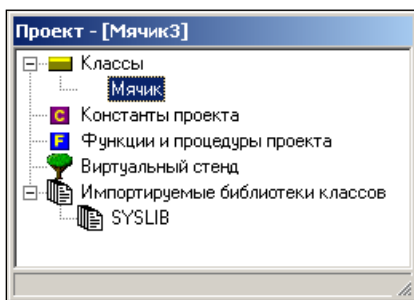


Рис. 1.11. Проект MVS

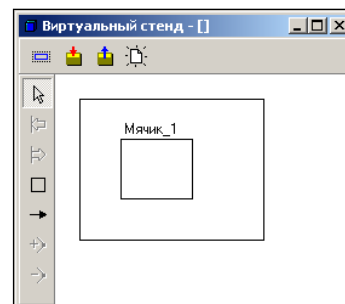


Рис. 1.12. Виртуальный стенд проекта

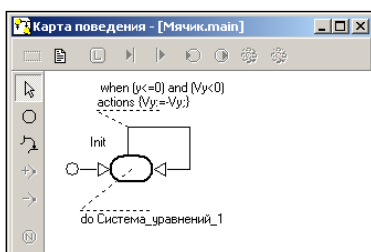


Рис. 1.13. Карта поведения

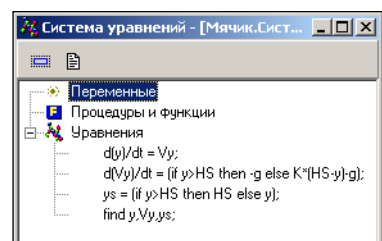


Рис. 1.14. Система уравнений

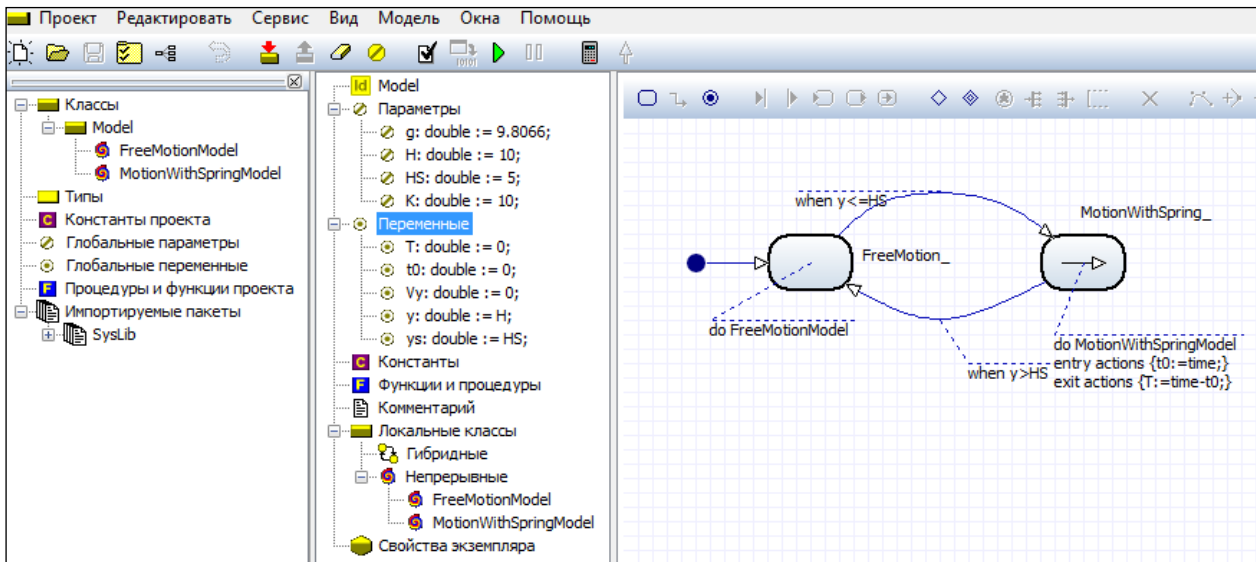


Рис. 1.15. Проект RMD с картой поведения

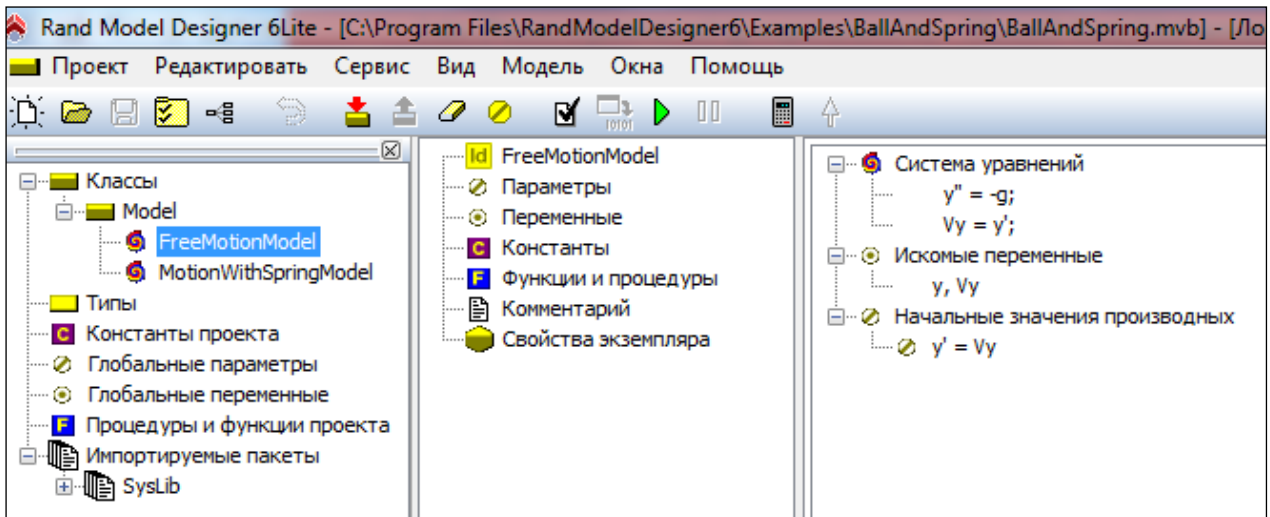


Рис. 1.16. Система уравнений проекта по рис. 1.15

Непрерывное поведение объекта в общем случае может быть задано системой обыкновенных дифференциальных уравнений вида $\frac{ds}{dt} = f(s, t)$, или совокупностью алгебраических уравнений вида $F(s, t) = 0$ или формул вида $s = \langle \text{выражение, не зависящее от } s \rangle$ (рис. 1.14).

Устройства могут объединяться в блоки, что позволяет строить модели систем по блочно-иерархическому принципу. Все взаимодействия блока осуществляются только через его входы и выходы, составляющие интерфейс блока

(рис. 1.18). Блок получает сигналы на входе, преобразует их и передает к выходу. Все остальные свойства блока инкапсулированы внутри него.

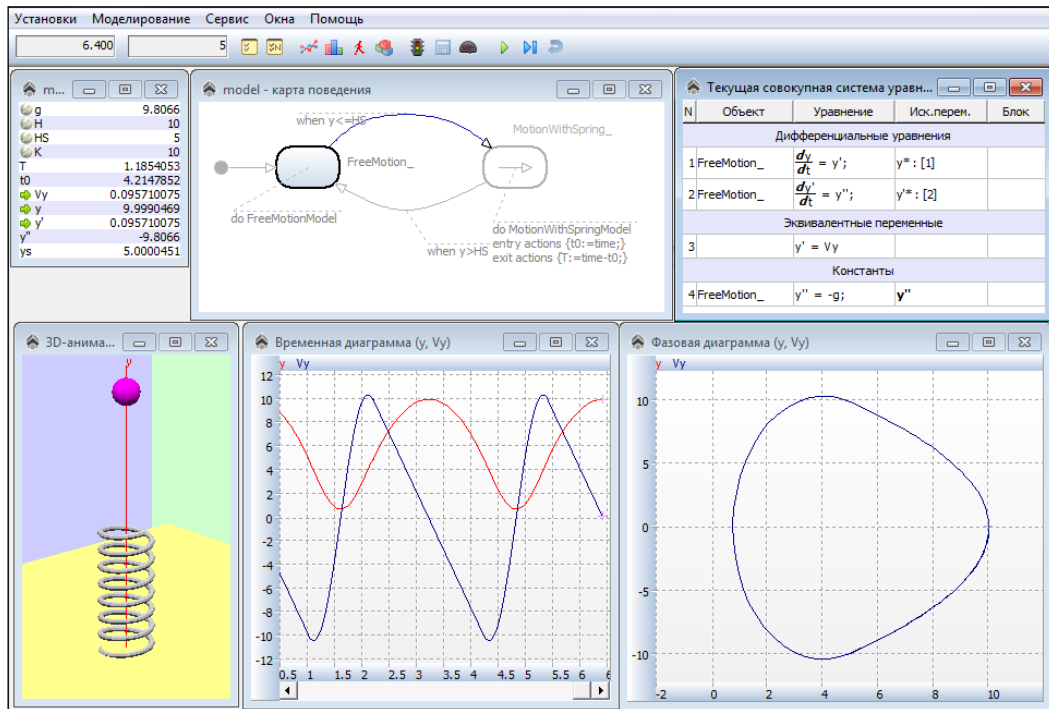


Рис. 1.17. Представление результатов моделирования в проекте по рис. 1.15

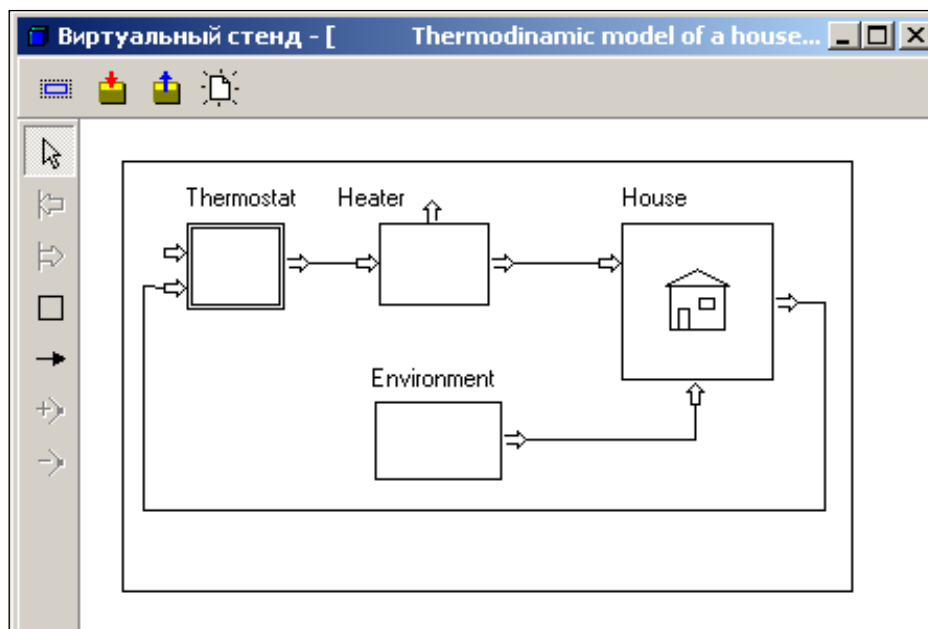


Рис. 1.18. Модель, построенная в виде блок-схемы

Блоки могут соединяться между собой функциональными связями и входить в состав других блоков, образуя определенную структуру. Конкретная

модель, с которой будет проводиться вычислительный эксперимент, собирается в окне виртуального стенда из экземпляров классов, определенных в данном проекте или из библиотечных классов. Она дополняется средствами графического отображения результатов моделирования (рис. 1.15).

Для отображения поведения объекта служит **карта поведения**. Это ориентированный граф, в котором узлам соответствует локальное непрерывное поведение, а дуги интерпретируются как переходы от одного поведения к другому. В каждый момент времени один из узлов графа является активным (рис. 1.13, 1.15, 1.19).



Рис. 1.19. Отображение смены состояний на карте поведения

Смена текущего узла происходит в результате срабатывания переходов. Для перехода между узлами должно быть задано запускающее событие: логическое условие, определяющее возможность срабатывания перехода; поступление внешнего сигнала; истечение заданного времени пребывания в текущем узле, а также действия (actions) в переходе (рис. 1.20).

Срабатывание перехода представляет собой следующую последовательность действий: данный узел перестает быть текущим; выполняется последовательность мгновенных действий в том порядке, как они записаны для перехода; текущим становится новый узел. Для каждого узла могут быть заданы входные действия (entry actions) и выходные действия (exit actions). Действия в переходах, входные и выходные действия, представляют собой программы, записанные на языке MVL (Model Visual Language). В MVS и RMD имеется богатая библиотека численных методов, предназначенных для воспроизведения поведения гибридных систем. Это программные реализации методов решения нелинейных алгебраических уравнений и систем обыкновенных дифференциаль-

ных уравнений. Для каждой группы задач имеется свой автоматический решатель, цель которого обеспечивать получение решения на заданном временном участке, любыми имеющимися в пакете методами.

К проекту могут быть присоединены любые ранее созданные библиотеки классов, таким образом, при создании своей модели пользователь может применять уже готовые классы устройств. Пакеты имеют стандартную библиотеку классов SysLib, содержащую набор наиболее типичных блоков и источников сигналов.

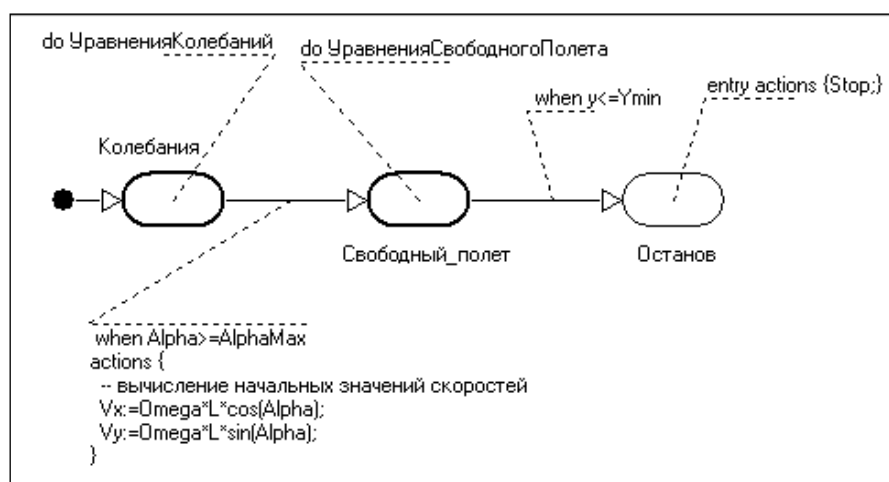


Рис. 1.20. Карта поведения с условиями срабатывания переходов

Система **AnyLogic** – профессиональный инструмент имитационного моделирования нового поколения, основанный на результатах, полученных в теории моделирования и в области информационных технологий в XXI веке. По сравнению с другими системами **AnyLogic** предоставляет существенно более широкий спектр возможностей (рис. 1.21) при меньших трудозатратах, поскольку позволяет:

- моделировать при помощи визуальных объектов;
- моделировать, применяя любые подходы, в любом сочетании;
- создавать интерактивные 2D и 3D анимации, визуально отображающие результаты работы модели в реальном времени;
- использовать мощный арсенал средств анализа и оптимизации непосредственно из среды разработки модели;

- просто и эффективно интегрировать модель с офисным и корпоративным программным обеспечением, включая электронные таблицы, БД, ERP и CRM системы.

В AnyLogic поддерживаются все существующие подходы дискретно-событийного и непрерывного моделирования: блок-схемы процессов, системная динамика, агентное моделирование, карты состояний, системы уравнений и т.д. (рис. 1.21, 1.23). AnyLogic позволяет строить стохастические и детерминированные модели и проводить анализ результатов моделирования.



Рис. 1.21. Возможности AnyLogic в моделировании

В AnyLogic встроен оптимизатор, который, используя эвристики, нейронные сети и математическую оптимизацию, позволяет находить значения дискретных и непрерывных параметров модели, соответствующие максимуму или минимуму целевой функции в условиях неопределённости и при наличии ограничений.

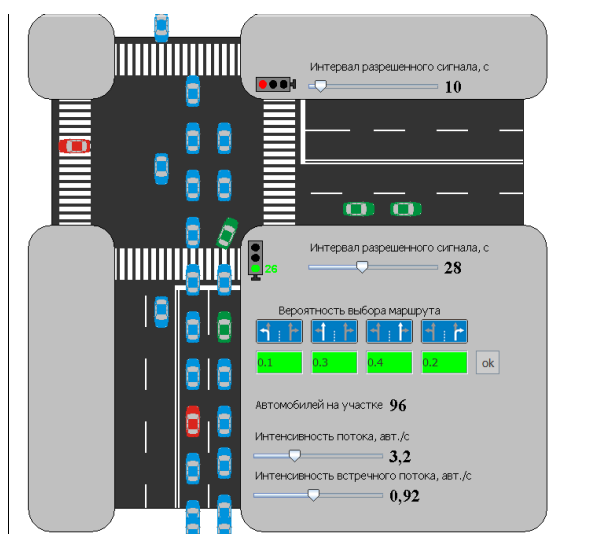


Рис. 1.22. Интерактивная AnyLogic-модель перекрестка с анимацией

С помощью технологии визуализации AnyLogic позволяет создавать интерактивные анимации произвольной сложности. Поскольку модели AnyLogic – Java-модели, их можно не только запускать на большинстве современных платформ, но и помещать на сайты в виде апплетов.



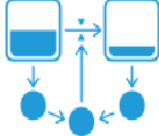
		
Агентное моделирование	Дискретно-событийное моделирование	Моделирование системной динамики

Рис. 1.23. Подходы к моделированию в AnyLogic

Это позволяет удалённым пользователям запускать интерактивные модели в браузере без необходимости устанавливать какое-либо программное обеспечение.

Следует отметить, что в данном разделе рассмотрены только некоторые системы моделирования общего назначения. Кроме них существует множество специализированных систем моделирования, которые применяются для решения узкоспециальных задач конкретной предметной области.

1.5. СИСТЕМНЫЙ ПОДХОД В МОДЕЛИРОВАНИИ

Системный подход – это подход к анализу объектов, в основе которого лежит исследование объектов как **систем**. Под **системой** будем понимать совокупность взаимодействующих между собой элементов, которая обладает определенной целостностью.

Еще в 1922 г. Александр Богданов, российский учёный, экономист, социальный философ, политический и общественный деятель, предложил идею создания новой научной дисциплины – тектологии, в которой предвосхитил некоторые положения кибернетики и системной теории. Людвиг фон Берталанфи –

австрийский биолог – в 1950 г. опубликовал работу «The Theory of Open Systems in Physics and Biology». В этой работе ученый утверждал, что «Физические системы отличаются от живых образований тем, что закрыты по отношению к внешней среде, тогда как живые организмы являются открытыми. Жизненный процесс организмов предполагает наличие входящего из окружающей среды потока материи, тип и объем которого определяется в соответствии с системными характеристиками организма. Так же осуществляется выход из системы в окружающую среду материи, как результата функционирования системы». В настоящее время Людвиг фон Берталанфи считается основателем обобщённой системной концепции под названием «Общая теория систем».

Актуальность системного подхода состоит в следующем:

- Системный подход актуален в связи с исследованиями больших реальных систем, когда недостаточно каких-либо частных решений.
- Большой объем исходных данных.
- Учет связей и взаимодействий с внешней средой.
- Необходимость изучения объекта во взаимодействии с внешней средой, а также в совокупности с другими системами.

Любой объект окружающего мира можно рассматривать как систему. Сам термин указывает на необходимость всестороннего исследования объекта, комплексно, в отличие от ранее принятого разделения исследований на физические, химические, биологические и т.д. Оказалось, что на этой основе можно получить более полные представления о реальных объектах, выявить их новые свойства, определить взаимоотношения с окружающей средой и т.д.

Применение методологии системного подхода позволило решать задачи проектирования, анализа целенаправленной деятельности, функционирования, планирования, управления и оптимизации для систем самой разнообразной природы и сложности. Таким образом, значение системного подхода для моделирования трудно переоценить.

Определить систему – значит выделить её из состава окружающей среды. **Элемент системы** – это некоторая относительно самостоятельная часть системы. В принципе разделить систему на элементы можно несколькими спосо-

бами в зависимости от характера решаемой задачи. Для любой системы существует **системообразующий фактор**. Это то, что объединяет элементы в систему и придаёт системе целостность. Системообразующий фактор позволяет выделить систему как самостоятельный объект окружающего мира.

В принципе, каждый элемент системы можно рассматривать как другую систему более низкого уровня – **подсистему**. С другой стороны, любую систему можно рассматривать как элемент более **общей системы**. Возможность деления системы на подсистемы связана с вычленением совокупности взаимодействующих элементов, способных выполнять относительно независимые функции.

Любая система имеет **структуру**. В структуре отображается состав элементов системы и связи между ними. Структура системы может быть отображена в виде графа или матрицы инцидентий (рис. 1.24). Связи между элементами обеспечивают сохранение структуры и свойств системы.

Функции и свойства системы зависят и от свойств элементов системы и от ее структуры. Свойства системы не являются простой суммой свойств её элементов. Соединение элементов в систему всегда дает новые свойства, так называемый **системный эффект**.

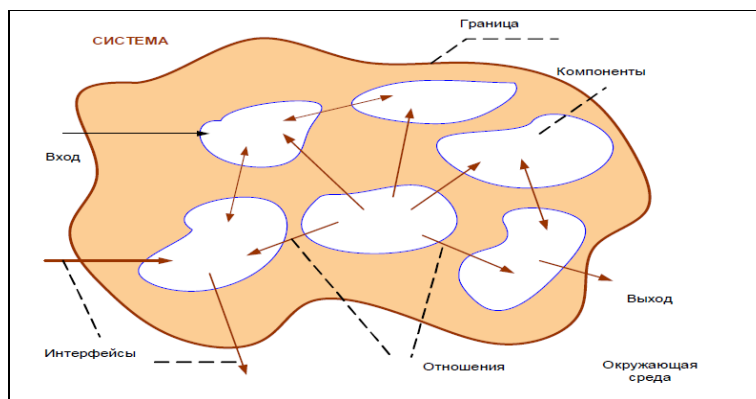


Рис. 1.24. Представление структуры системы

Основные понятия системного подхода представляются терминами:

- **Деталь (элемент системы)** – это неделимая часть системы.
- **Подсистема** – это относительно самостоятельная часть системы способная выполнять определенные функции.
- **Окружающая среда** – это факторы вне системы, но влияющие на нее.

- **Структура** – модель связей между элементами системы.
- **Связь** – это характеристика, имеющая такие признаки как направление, сила и другие.
- **Состояние** – это совокупность значений различных параметров в данный момент.
- **Поведение** – изменение состояния во времени.
- **Устойчивость** – это способность системы возвращаться в исходное состояние после прекращения действия возмущений.

Информация о свойствах системы в определенный момент времени определяет **состояние системы**. **Целенаправленное поведение** системы представляется как стабильная способность к определенным действиям. **Движение (поведение)** системы – это процесс перехода системы из одного состояния в другое и т.д. Если переход системы из одного состояния в другое происходит скачкообразно, то система называется **дискретной**. Если при переходе между любыми двумя состояниями система обязательно проходит через **промежуточное** состояние, то она называется **непрерывной**.

Следует упомянуть теорему Эшби, английского ученого и одного из крупных теоретиков кибернетики. Теорема Эшби формулируется так: у системы тем больше возможностей в выборе поведения, чем сильнее степень согласованности поведения ее частей.

Можно выделить следующие режимы поведения системы:

- **стационарный режим** (соответствует положению равновесия, система находится все время в одном и том же состоянии);
- **динамический режим** (состояние системы непрерывно меняется во времени);
- **периодический режим** (система через равные промежутки времени проходит одни и те же состояния);
- **переходный режим** (соответствует движению системы из одного стационарного состояния в другое).

Возможность отражения поведения системы зависит от того, каким образом мы представляем объект моделирования. Например, воздух в комнате мож-

но представить в виде системы молекул, каждая из которых имеет свои координаты и скорость. В этом случае мы получим сверхсложную систему с хаотическим движением молекул, которую исследовать вряд ли возможно. Если же воздух в комнате представить как систему, состоящую из одного элемента, параметрами которого являются давление, температура и влажность, то такую систему достаточно просто описать в соответствии с законами термодинамики. Для всех практических задач этот способ определения системы предпочтительнее.

Всё, что не входит в систему, но взаимодействует с ней, называется **окружающей средой** (рис. 1.24). Любая система всегда существует и функционирует, взаимодействуя с окружающей средой. Это взаимодействие происходит по принципу обратной связи (рис. 1.25).

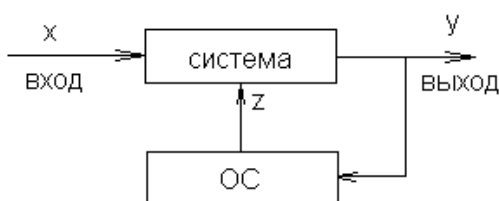


Рис. 1.25. Схема взаимодействия системы и окружающей среды

Реакция системы Y воздействует на окружающую среду. В результате окружающая среда (ОС) вырабатывает ответную реакцию Z , дополнительно воздействующую на систему.

Классификацию систем можно провести и по критериям: сложности, детерминированности и непрерывности. Системы по степени сложности можно разделить на **простые, сложные и сверхсложные**. К простым системам относятся системы, имеющие соответствующую простую структуру, достаточно легко поддающиеся математическому описанию.

Сложными являются системы, имеющие много элементов и внутренних связей и, соответственно, сложное математическое описание. Математические модели таких систем реализуются только с использованием вычислительной техники. Сверхсложные системы не поддаются математическому описанию, их исследование возможно исключительно методами имитационного моделиро-

вания. Сложность системы может определяться ее поведением. Примером являются гибридные системы, которые сочетают непрерывное и дискретное поведение.

Системы можно разделить на **детерминированные** и **стохастические**. На поведение стохастических систем оказывают влияние случайные факторы.

Возможные варианты систем получаются комбинированием указанных классов: от простых детерминированных моделей до сверхсложных вероятностных.

Например, **непрерывно-детерминированные** системы описываются моделями на основе дифференциальных уравнений (**D-схемы**). Искомые переменные являются непрерывными. Подобные модели, как правило, отражают динамику изменения состояния системы.

К **дискретно-детерминированным** системам (**F-схемы**) относятся так называемые конечные автоматы. Автомат – это некоторое устройство, на которое подаются входные сигналы и снимаются выходные. У конечного автомата множество входных сигналов и множество внутренних состояний конечно. Если состояние автомата имеет случайный характер, то такая система относится к классу **дискретно-стохастических** систем (**P-схемы**).

Изучение и описание любой системы неизбежно связано с моделированием. Моделирование систем осуществляется на двух уровнях. На **внешнем уровне** происходит выделение самой системы и ее связей с окружающей средой. На **внутреннем уровне** моделирование системы состоит в том, что производится выделение элементов системы и связей между ними. На практике при моделировании систем решается два типа задач: задача **анализа** – исследование свойств существующей системы; задача **синтеза** – построение новых систем с заранее заданными свойствами.

В заключение отметим, что системный подход позволяет сделать ряд простых, но весьма важных для моделирования выводов: при построении модели системы необходимо разработать не только модель **собственно системы**, но и модель её взаимодействия с **окружающей средой**.

При построении модели объекта требуется провести его **системный анализ**, при этом необходимо выявить:

- 1) элементы и подсистемы, из которых он состоит;
- 2) особенности взаимодействия элементов друг с другом;
- 3) особенности взаимодействия элементов с окружающей средой.

Рассмотренные положения системного подхода в определенном смысле составляют методологическую основу моделирования, что, в первую очередь, относится к моделированию сложных систем.

1.6. Устойчивость систем

Функционирование любой системы сопряжено с действием внешних и внутренних возмущений. Устойчивая система всегда возвращается в исходное состояние после прекращения действия возмущений.

Неустойчивая система либо переходит в новое состояние, либо совершает колебания вокруг исходного состояния (режим автоколебаний), либо в ней развивается хаотическое поведение.

Для некоторых нелинейных систем устойчивость зависит от величины возмущения. Различают устойчивость **«в малом»**, устойчивость **«в большом»** и устойчивость **«в целом»**.

Система считается устойчивой **«в малом»** (относительно малых возмущений), если ее устойчивость установлена без указания величины возмущения. Устойчивость **«в малом»** важное свойство. Во многих случаях величина внешних возмущений достаточно мала. Если величина возмущений, при которых система сохраняет устойчивость, установлена, то имеет место устойчивость **«в большом»**. Устойчивость **«в целом»** имеет место для любых возмущений.

Неустойчивые системы неспособны выполнять свои функции. Анализ устойчивости систем является главной задачей. Исследование устойчивости систем начинается с устойчивости **«в малом»**. Для этого проводится исследование линеаризованных моделей систем (рис. 1.25).

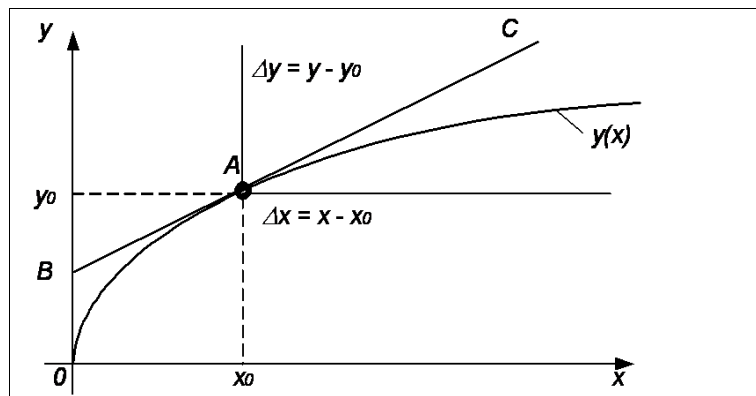


Рис. 1.25. Геометрический смысл линеаризации

$$\Delta y = \frac{\partial y}{\partial x_{x=x_0}} \Delta x, \quad y = \frac{\partial y}{\partial x_{x=x_0}} (x - x_0) + y_0$$

То есть, линеаризация – замена нелинейных зависимостей их линейными приближениями, которые справедливы при малых значениях Δx . Исследование устойчивости систем – предмет изучения специального научного направления «теория устойчивости».

1.7. ОБЩАЯ СХЕМА ПОСТРОЕНИЯ МОДЕЛИ

Долгое время построение моделей считалось искусством. Успех в этой деятельности определялся, практическим опытом, экспериментаторским мастерством, интуицией. Основой моделирования, особенно при создании новой техники, оставался метод проб и ошибок. В этом случае создание новой технической системы сопровождалось длительной отработкой объекта проектирования на основе испытаний опытных образцов. Эти факторы и сегодня играют достаточно большую роль. Вместе с тем опыт моделирования все более сложных технических систем убедительно показывает, что качество моделей решающим образом зависит от того, насколько целесообразно построен весь процесс их создания.

Общая логика моделирования получила развитие в рамках работ по теории планирования эксперимента, теории подобия, прикладному системному анализу и теории идентификации как инструмента построения моделей слабо изученных сложных систем. Достижения в данных дисциплинах составляют сегодня научный фундамент методологии моделирования.

В общем случае построение модели и моделирование актуально при возникновении какой-либо **проблемы**. Анализ проблемы приводит к формулированию **задач** исследования, которые содержат и описание объекта моделирования. В технике результаты данной работы фиксируются в виде технического задания на проектирование.

Следующим шагом в построении модели является **анализ** объекта моделирования с точки зрения поставленной задачи. В итоге формируется некий образ объекта, который можно назвать **когнитивной** моделью. При этом с целью получения компактного описания объекта сложная реальность упрощается путем отсечения всего второстепенного для решения конкретной задачи.

На основе когнитивной создается **концептуальная** модель, при построении которой используются научные положения, законы и понятия той предметной области, где решается задача. В технике на основе концептуальной модели удается выполнить математическое описание объекта.

Если объект достаточно изучен, применяется теоретический путь построения модели. Альтернативой является **идентификация** модели объекта, т. е. определение связей между свойствами объекта на основе статистической обработки результатов наблюдений.

Рассмотрим общую схему построения модели без учета особенностей конкретных видов моделей. Данная схема содержит этапы, характерные для моделирования в целом, они реализуются независимо от того, имеем ли мы представление об их существовании или нет. Содержание этапов построения модели представляется следующим:

1. На основе существующей **проблемы** формулируется **задача** (одна или несколько). Выбирается **объект**, действия с которым приведут к решению поставленной задачи.

2. Выполняется **системный анализ** объекта моделирования: устанавливается, из каких элементов состоит объект, как они взаимодействуют между собой. Устанавливаются **свойства** объекта актуальные для решения поставленной задачи. Выявляются **факторы**, определяющие эти свойства.

3. Выполняется **создание** собственно модели, при этом производится выбор вида модели и способа её построения. Естественно, что выбор существенно зависит от решаемой задачи и возможностей исследователя.

4. Решается вопрос об **интерпретации** результатов моделирования, если это необходимо. Каким образом результаты экспериментов с моделью будут перенесены на реальный объект. При этом следует учесть, что модель – это самостоятельный объект, обладающий рядом таких собственных свойств, которые не имеют никакого отношения к объекту моделирования.

5. Проводятся эксперименты с моделью, осуществляется проверка ее **адекватности**. Адекватность – это степень соответствия по моделируемым свойствам между моделью и объектом. Адекватность характеризует качество отображения моделью свойств реального объекта с точки зрения решаемой задачи.

6. Выполняется **корректировка** или переработка модели в случае ее слабой адекватности.

7. Модель **применяется** для решения поставленной задачи.

Следует отметить, что любые правила разработки моделей могут служить лишь в качестве определенного каркаса. Видимо основой успешной методики моделирования может быть последовательная разработка системы моделей. Начав с относительно простой модели, постепенно необходимо продвигаться к более сложной ее форме, отражающей ситуацию более точно. По мере проведения испытаний, получения дополнительных данных обычно возникает уточненный вариант модели. Однако, в ряде случаев простые модели в инженерной практике дают достаточно адекватные результаты.

1.8. АДЕКВАТНОСТЬ МОДЕЛЕЙ

Адекватность – это характеристика точности отражения моделью свойств объекта-оригинала, точности необходимой для решения поставленной задачи. Модель, с помощью которой успешно решается поставленная задача, будем называть адекватной. Таким образом, адекватность модели означает, что требования точности отражения свойств выполнены в той мере, которая достаточна для решения поставленной задачи.

Естественное различие между моделью и оригиналом вызвано тем, что в модели отображаются лишь самые важные для решения задачи свойства объекта. В результате такое свойство модели как приближенность неизбежно, но оказывается, что даже простых моделей достаточно для практической деятельности, особенно инженерной.

Следует помнить, что модель – это самостоятельный объект, имеющий и свои собственные свойства, которые могут быть не связаны со свойствами моделируемого объекта. Таким образом, проблема адекватности одна из важнейших. От адекватности модели зависит степень доверия к результатам моделирования. В ряде случаев удается ввести некоторую меру адекватности модели, например, коэффициент достоверности.

Рассмотрим некоторые причины неадекватности. При построении моделей всегда производится отбор свойств объекта, которые актуальны для решения поставленной задачи. Причем, в первую очередь, естественно, рассматриваются известные свойства и отношения. Вполне вероятно, что моделируемый объект имеет еще **неизвестные** свойства, которые также значимы и актуальны для данной задачи. Таким образом, причиной неадекватности может быть **неполный учет свойств**, существенно влияющих на поведение объекта. Кроме того, модель всегда есть продукт деятельности человека, и несет в себе определенную долю субъективизма.

Другой возможной причиной неадекватности может быть **неполный учет факторов**, которые определяют актуальные свойства объекта моделирования, вынужденное упрощение закономерностей, неполнота и неточность используемых при построении моделей данных наблюдений и экспериментов.

При построении любой модели неизбежно производится упрощение, схематизация, принимаются определенные **допущения**, которые ограничивают область применения модели. Использование модели в условиях, когда эти допущения нарушаются, с большой вероятностью даст неадекватные результаты.

Проверка адекватности возможна путем сравнения результатов моделирования и данных экспериментов с реальным объектом. Для доказательства адекватности модели можно провести ретроспективный анализ поведения объекта моделирования. Если данные о прошлой «жизни» объекта достаточно хорошо описываются моделью, то есть основания предполагать, что такой же результат будет в случае прогноза развития процессов в будущем.

Можно различать адекватность на качественном и количественном уровне. Модель, адекватная на **качественном уровне**, воспроизводит определенные эффекты, например, наличие резонанса, развитие автоколебаний и т.д. Если модель адекватна на качественном уровне, то после этого можно говорить об адекватности на количественном уровне. Адекватность на **количественном уровне** – это отражение с необходимой степенью точности изменения параметров системы.

Адекватность непосредственно связана со степенью детализации описания объекта. Естественно, что детализация описания протекающих в объекте моделирования процессов ограничена общим уровнем знаний и возможностью проведения экспериментов с самой моделью.

Слишком **простые модели** совершенно **не гарантируют** отражения свойств объекта во всей необходимой полноте. С другой стороны, слишком сложные модели лишь **потенциально** «богаче», чем простые модели. Для исследования сложных моделей требуется применение достаточно сложных методов, что неизбежно вносит определенную погрешность, требует намного больших затрат времени и средств. Кроме того, в любой модели используются эмпирические и полуэмпирические законы, низкая степень точности которых существенным образом влияет на качество результатов моделирования.

С практической точки зрения бессмысленно строить детальные модели и проводить слишком точные численные расчеты, если при разработке модели приняты весьма грубые исходные допущения. Многие крупнейшие инже-

неры-математики рассматривали излишнюю точность методов получения результатов как грубейшую ошибку. Следовательно, при неточности исходных данных и детальная модель не гарантирует высокой степени адекватности. Для разрешения данной проблемы обычно создают ряд моделей, начиная с самых простых.

Таким образом, любой результат модельного исследования должен восприниматься критически. Такие исследования всегда основаны на тех или иных гипотезах или допущениях, которые могут оказаться и ошибочными.

Результаты моделирования и результаты экспериментов с объектом всегда различаются. Это обусловлено не только свойствами модели, но и неустранимой погрешностью измерений в ходе экспериментов, и влиянием случайных факторов. Возникает задача установления соответствия модели реальному объекту на основе неточных результатов измерений. В ряде случаев решение подобной задачи возможно методами математической статистики.

В заключение отметим, что проверка адекватности модели является одним из важнейших этапов ее разработки.

1.9. КЛАССИФИКАЦИЯ МОДЕЛЕЙ

Задача классификации нетривиальна даже в тех случаях, когда признаки родства элементов совершенно объективны. Пример тому – классификация в биологии, которая достаточно долго является предметом научных дискуссий. Модели строятся для объектов с чрезвычайно широким спектром признаков, поэтому общепринятой классификации моделей на сегодня не существует.

Рассмотрим классификацию, которая опирается на определение модели, а также учитывает признаки, принятые в наиболее развитых областях моделирования.

Каждая модель характеризуется следующими основными признаками:

1. Принадлежность к определенному классу задач.
2. Принадлежность к определенному классу объектов.
3. Способ построения самой модели.

4. Характеристика основных свойств объекта, которые отражаются моделью.
5. Цель моделирования.

Именно эти признаки часто выносятся название модели. Например, экономико-математическая модель. Ясно, что такая модель построена средствами математики (способ реализации), а объект моделирования относится к сфере экономики.

1. **По классам решаемых задач** модели можно разделить на: проектные; оптимизационные; технологические; управленческие и т.п. Естественно, представленный здесь перечень не покрывает всех задач, решаемых методами моделирования, а каждый класс может быть разделен далее на множество подклассов.

2. **Классы объектов моделирования** – второй основной классификационный признак. Перечень объектов соответствует исторически сложившимся в науке представлениям: физические, химические, биологические, экономические и т.д.

3. **По способу реализации модели** разделяются на материальные модели и абстрактные.

4. **Характер отражаемых моделью свойств** объекта-оригинала часто выносятся в ее название. Например: **структурная модель** – такая модель отражает структуру системы. Если модель именуется **структурно-функциональной**, то ясно, что речь идет о взаимосвязи функций элементов системы. Геометрическая модель отражает геометрические свойства объекта: ориентацию в пространстве, взаимное расположение, размеры и форму элементов системы.

Например, при проектировании любого здания или сооружения создается ряд моделей, каждая из которых отражает свою группу специфических свойств объекта. Геометрическая модель отражает внешний вид, внутреннюю планировку и общий архитектурный замысел.

В ряде случаев модели разделяют по степени неопределенности на **детерминированные и вероятностные (стохастические)**, которые отражают влияние случайных факторов.

При разделении моделей по зависимости от фактора времени разделяют **динамические модели и статические модели**. Первые учитывают изменение

состояния объекта моделирования во времени. Статические модели отражают некоторое равновесное, которое не меняется во времени.

5. **По цели моделирования** модели делятся на следующие виды:

- Познавательные модели, которые создаются с целью научного познания объектов, процессов и явлений.
- Образовательные модели, предназначенные для обучения.
- Проектные модели, которые строятся с целью создания новых объектов с заданными свойствами.
- Модели управления, использующиеся с целью получения желаемого состояния или поведения объектов.
- Прогностические модели, которые способны предсказать будущее состояние объекта или развитие процесса;
- Модели диагностики состояния объекта предназначены для оценки текущего состояния объекта с целью выявления неисправностей. Представленная классификация не претендует на полноту (известно несколько десятков классификаций). Она затрагивает лишь те свойства, которые составляют основу понятия «модель».

1.10. ИНФОРМАЦИОННЫЕ МОДЕЛИ

В настоящее время термин **«информационная модель»** широко используется в курсе информатики при изучении темы «моделирование». Вместе с тем смысл этого термина многими авторами трактуется неоднозначно. Под моделью мы будем понимать объект произвольной природы, отражающий свойства, характеристики и связи моделируемого объекта (объекта-оригинала), которые считаются существенными для решения поставленной задачи.

Таким образом, определение модели включает следующие понятия: объект моделирования; свойства объекта моделирования, актуальные для решения поставленной задачи; модель, как самостоятельный объект; сама решаемая задача. Здесь же можно отметить и способ построения модели.

В итоге эти понятия зачастую выносятся в наименование модели, например: структурная модель, геометрическая модель, компьютерная модель, имитационная модель, материальная модель (макет), оптимизационная модель.

В пособии не преследуется цель дать однозначные ответы на все вопросы, связанные с понятием «информационная модель», мы рассмотрим разные точки зрения, которые представлены в современной учебной, методической и научной литературе по информатике и моделированию.

Итак, в зависимости от целей моделирования возможно несколько уровней формализованного описания объектов моделирования: символический уровень, теоретико-множественный уровень, абстрактно-алгебраический, топологический, логико-математический и информационный уровни. На информационном уровне описания систем предполагается, что в любом процессе управления или регулирования происходит переработка входных информационных сигналов в выходные сигналы. Таким образом, в основе информационных моделей лежит отражение информационных процессов в системах управления.

Согласно А.П. Ершову: Информационная модель – это модель, в которой изучаемое явление представляется в виде процессов передачи и обработки информации» и, следовательно, отображает закономерности протекания этих процессов.

Информационную модель характеризует формализованное описание информационных структур и операций над ними и параметрическое представление процесса циркуляции информации, подлежащей автоматизированной обработке в системе управления. Подобная модель – это вид знаковых моделей, описывающих информационные процессы (возникновение, передачу, преобразование и использование информации) в системах самой разнообразной природы.

Данные определения подчеркивает информационный аспект изучения объектов. Из этих определений следует, что при информационном моделировании отображаются, в первую очередь, процессы передачи и преобразования сигналов, например, в системах управления или регулирования. Подобное по-

нимание сущности информационной модели, по нашему мнению, наиболее приемлемо.

В литературе встречается множество других определений понятия «информационная модель». Например, можно встретить такие определения:

«Информационная модель объекта – это его описание».

«В самом общем виде информационная модель материального объекта, явления или процесса – это описание этого объекта на одном из языков кодирования».

«Модели можно отнести к одной из двух групп: натурные (материальные) и информационные (нематериальные)».

Иными словами, если построено описание объекта, то построена его информационная модель. Эти определения можно понимать и так, что описание объекта и есть его информационная модель, потому что оно содержит определенную информацию об объекте. Любая модель, в том числе и материальная, содержит информацию об объекте моделирования и его свойствах. Это основное назначение моделей – отражать свойства объекта моделирования, которые актуальны для решения поставленной задачи. Такое определение информационной модели имеет весьма общий характер, который, на наш взгляд, не отражает специфических особенностей моделей.

Из всего множества определений можно выделить такие, которые связывают понятие «информационная модель» с характером отражаемых моделью процессов. Вторая группа определений связывает понятие «информационная модель» со способом построения самой модели.

Таким образом, в настоящее время имеется множество разнообразных подходов к понятию «информационная модель», отличающихся как степенью обобщения, так и рядом принципиальных признаков. Следствием этого обстоятельства и является различие в составе класса информационных моделей.

Дополнительные сведения об основах моделирования можно найти в работах, представленных в библиографическом списке [6; 16; 15; 43; 45; 55].

1.11. КОНТРОЛЬНЫЕ ВОПРОСЫ К ГЛАВЕ 1

1. В каких областях научных знаний находит применение моделирование?
2. Раскройте суть понятий «модель» и «моделирование».
3. В каких случаях моделирование особо актуально?
4. Укажите причины необходимости наличия множества моделей.
5. Каковы основные функции моделей?
6. Какие типовые задачи решаются путем моделирования?
7. Какие модели составляют фундамент любой научной теории?
8. Каковы особенности имеет компьютерное моделирование?
9. В чем суть системного подхода?
10. Раскройте суть следующих понятий: система, подсистема, окружающая среда, структура системы.
11. Как происходит взаимодействие системы и окружающей среды?
12. Какие выводы для моделирования позволяет сделать системный подход?
13. Какие аспекты системного подхода используются при построении моделей?
14. Перечислите этапы построения модели.
15. Какие факторы должны учитываться при выборе вида модели?
16. Что такое адекватность модели?
17. Каковы причины неадекватности?
18. Что такое формализация?
19. Почему возникла потребность в формализации?
20. По каким основным признакам классифицируются модели?
21. Что такое информационная модель?
22. Что такое устойчивость системы?
23. Перечислите виды устойчивости систем.

ГЛАВА 2. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

2.1. ВВЕДЕНИЕ В МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

Элементы математического моделирования используются с момента появления точных наук. Собственно и само рождение науки математики связано с решением практических задач на основе вычислений и моделирования. Многие методы математического моделирования носят имена великих ученых-математиков. Второе рождение математического моделирования связано с появлением компьютеров, которые избавили ученых и инженеров от огромной рутинной работы по проведению расчетов и существенно расширили сферы приложения математического моделирования, без которого нельзя представить современную науку и технику. Актуальность математического моделирования связана со следующими обстоятельствами:

- Использование научных знаний для решения конкретных практических задач. Например, задача проектирования – установить параметры объекта, чтобы его свойства соответствовали требуемым.
- Общенаучное значение математического моделирования, т.е. возможность сравнения выводов теории с результатами наблюдений и экспериментов.

Теоретические выводы получают точный и практически значимый характер, если они выражены в виде количественных соотношений. Как уже отмечалось при обсуждении актуальности моделирования, прямой натурный эксперимент для многих объектов долог, дорог, опасен или невозможен. Многие из технических систем существуют в единственном экземпляре, а результаты экспериментов с реальными объектами могут привести к необратимым отрицательным последствиям. При создании новых технических объектов их необходимо сначала спроектировать, т.е. установить какие значения должны иметь

параметры объекта, чтобы его свойства и функции соответствовали требуемым. Иными словами, параметры нового объекта, прежде чем он будет создан, нужно рассчитать и необходимо провести анализ его свойств.

Совершенно очевидно, что результаты теоретических исследований в любой области науки будут иметь наибольшее практическое значение, если они будут выражены в виде конкретных **количественных зависимостей**, или, попросту говоря, в виде математических формул или вычислительных алгоритмов. Это залог эффективного применения теории в практических целях. Кроме того, теоретические результаты необходимо сопоставлять с результатами измерений, полученных в ходе экспериментов. Для решения подобных проблем необходимо моделирование на основе количественных закономерностей протекающих процессов. Именно такую возможность представляет математическое моделирование. Оно является незаменимой составляющей в развитии науки и техники.

Любой объект имеет множество свойств. Математическое моделирование какого-либо объекта или процесса связано с отражением **количественных характеристик** его свойств, как правило, в числовом виде. Последнее время получило развитие направление применения и изучения т.н. «жестких» и «мягких» математических моделей. «Мягкие» модели ориентированы на изучение качественных свойств объектов или изучение тенденций в развитии процессов.

Количественное выражение свойства объекта выполняется с помощью системы **параметров**, но только в том случае, если эти свойства можно **измерить**. Лишь в результате измерения свойств параметры могут получить свои значения. Измерения любого свойства всегда связаны с какой-либо **шкалой**, в рамках которой они проводятся. Именно шкала позволяет выполнить сравнение нескольких объектов по одному свойству.

В результате измерения наблюдаемому состоянию объекта ставится в соответствие определенное обозначение: число, номер, знак, символ. Такое соответствие обеспечивает информативность результатов измерений.

Будем рассматривать такие состояния объекта, про которые определенно можно сказать, различимы они или нет. Допустим, что число различимых со-

стояний конечно. Каждому состоянию поставим в соответствие свое обозначение. Суть измерений состоит в определении принадлежности результата к конкретному классу. Результат измерения описывают с помощью определенного обозначения. Такие измерения называются измерениями в **шкале наименований** (классификационной шкале).

Раз обозначения классов – символы (даже, если это цифры), то при обработке данных в шкале наименований можно выполнить только операцию проверки различия или совпадения этих результатов

Когда наблюдения позволяют **сравнивать** разные классы, для измерения можно выбрать более сильную **порядковую шкалу**. В этом случае между классами можно установить соотношения типа $A > B$ или $B = A$. К таким шкалам можно отнести систему воинских званий, обозначение мест в конкурсе, оценки успеваемости в процессе обучения. В этом случае для обозначения классов могут использоваться слова (отлично, хорошо и т.д.) или цифры (5, 4, 3, 2). Отношения порядка не определяют **«расстояния»** между классами. Измерения в таких шкалах нельзя обрабатывать, как числа (например, вычислять среднеарифметическое значение оценок успеваемости).

Если упорядочение можно выполнить настолько точно, что становятся известны **расстояния** между любыми двумя классами, то измерения можно проводить в **шкале интервалов**. Расстояние между классами должно выражаться в единицах, одинаковых по всей шкале. Подобная шкала может иметь произвольное начало отсчета. Такие шкалы используются при измерении времени, высоты, температуры. В подобных шкалах только длины интервалов имеют смысл настоящих чисел, а с числами можно выполнять арифметические операции. Частным случаем интервальных шкал является **циклическая шкала** (время суток, фаза колебаний и т.п.).

Наиболее «сильной» шкалой является **абсолютная шкала**, она имеет абсолютный нуль и абсолютную единицу. Абсолютная шкала безразмерна (абсолютная шкала температур). С результатами измерений в такой шкале можно выполнять любые операции.

Рассмотренные выше шкалы имеют общее свойство: они основаны на том, что два состояния или результаты двух измерений либо тождественны, либо различимы. В действительности очень часто встречаются случаи, когда о результатах измерения нельзя говорить с полной уверенностью. Наиболее ярко это видно на примере шкал, где классы обозначаются словами естественного языка, например: высокий, низкий; тяжелый, легкий и т.д. В подобных случаях мы имеем дело с нечеткой оценкой свойств объекта. Размытость встречается не только в естественном языке. В математике успешно применяются понятия «значительно больше» (\gg) или «приблизительно равно» (\cong), которые являются расплывчатыми.

Опираясь на общее определение модели, можно сказать, что математическая модель – это математический объект, который по своим свойствам подобен объекту-оригиналу. Математическая модель – это абстракция, **в которой отношения и связи между реальными элементами объекта моделирования заменены подходящими математическими соотношениями между параметрами объекта.** Математическая модель отражает количественные характеристики процессов, которые протекают в объекте, т.е. она отражает связи между параметрами объекта. Параметры можно разделить по степени принадлежности собственно объекту или по характеру взаимодействия с окружающей средой:

1. **Входные параметры.** Характеризуют внешние воздействия на объект (например, внешние управляющие воздействия).
2. **Выходные параметры.** Характеризуют реакцию объекта, его воздействие на окружающую среду, т. е. характеризуют внешнее проявление объекта.
3. **Внутренние параметры.** Характеризуют свойства процессов, протекающих в самом объекте. Внутренние параметры могут быть зависимыми или независимыми. Естественно, что независимые параметры можно изменять произвольно, а зависимые параметры изменяются только косвенно, в силу их зависимости от других факторов.

Таким образом, объект моделирования характеризуется:

- совокупностью внешних управляющих воздействий: $\overset{1}{X}(t)$;

- совокупностью воздействий окружающей среды: $\dot{V}(t)$;
- совокупностью внутренних независимых параметров: $\dot{H}(t)$;
- совокупностью внутренних зависимых параметров: $\dot{P}(t)$;
- совокупностью выходных параметров системы: $\dot{Y}(t)$.

С учетом вышесказанного, можно утверждать, что математическая модель устанавливает количественные связи между входными и независимыми внутренними параметрами с одной стороны и выходными и внутренними зависимыми параметрами с другой стороны. Как правило, математическая модель – это система уравнений (алгебраических, дифференциальных, интегральных).

В общем случае векторные величины $\overset{P}{X}$, $\overset{P}{V}$, $\overset{P}{H}$ могут содержать как детерминированные, так и стохастические (случайные) составляющие. Входные воздействия $\overset{P}{X}$, внутренние параметры $\overset{P}{V}$ и частично воздействия окружающей среды являются независимыми переменными. Совокупность векторных величин $\overset{P}{X}$, $\overset{P}{V}$, $\overset{P}{H}$, $\overset{P}{Y}$ полностью характеризует состояние объекта. Процесс изменения состояния объекта можно выразить следующей обобщенной математической моделью:

$$\dot{Y}(t) = F_1(t, \overset{P}{X}, \overset{P}{V}, \overset{P}{P}, \overset{P}{H}), \quad \dot{P}(t) = F_2(t, \overset{P}{X}, \overset{P}{V}, \overset{P}{H}), \quad \dot{V}(t) = F_3(t, \overset{P}{Y}).$$

Представленные соотношения дополняются начальными и краевыми условиями, которые определяют состояние объекта моделирования в момент времени $t=0$ и взаимодействие с окружающей средой. Данные соотношения называются законом функционирования объекта. Подобные модели принято называть динамическими. В этом случае параметры, которые изменяют свои значения во времени и требуют определения, называются переменными.

Статические модели отражают состояние объекта, которое не меняется во времени (состояние равновесия). Совокупность переменных, определяющих состояние динамической системы, называют фазовым вектором, а область изменения этого вектора – **фазовым пространством**.

В большинстве случаев математическая модель представляет собой задачу некоторого раздела математики, для которой методы исследования уже разработаны.

Замечательным свойством является формальное сходство (**аналогия**) математических моделей разнородных по своей природе объектов и процессов. Таким образом, имеется возможность сгруппировать математические модели в однородные с точки зрения математики классы и исследовать их как самостоятельные абстрактные математические объекты безотносительно оригиналов.

Основой построения математической модели могут быть **фундаментальные законы природы**. Наиболее распространенный способ построения математических моделей как раз и состоит в применении фундаментальных законов к конкретной ситуации. Однако чисто теоретическим путем математическую модель какого-либо объекта построить проблематично, на определенном этапе всегда приходится использовать данные экспериментов и наблюдений, феноменологические законы или полуэмпирические зависимости.

В научной литературе в качестве необходимых условий содержательного математического моделирования предполагается наличие **априорной информации** о природе и характере исследуемых объектов и процессов, например, в форме научных теорий, законов и т.п. Кроме того, необходимо наличие некоторых опытных данных о процессах в исследуемом объекте. Из исходной априорной информации выводятся общие математические соотношения, описывающие законы функционирования объекта моделирования. А на основе статистической обработки опытных данных определяются численные значения параметров модели. Результаты такой обработки опытных данных могут использоваться в виде фундаментальных физических констант или полуэмпирических зависимостей, которые во множестве можно найти в специальной справочной литературе.

Таким образом, при построении математических моделей существует несколько возможностей решения задачи:

1. Построение модели на основе законов, описывающих протекающие в объекте процессы, т.п. на основе знания о механизмах процессов и явлений с привлечением фундаментальных законов природы.

Такой метод можно назвать **аналитическим или теоретическим**. При построении модели составляется описание закономерностей протекающих в объекте процессов в виде набора математических соотношений. Далее, на основе анализа модели, делаются определенные выводы, которые проверяются на практике. Достоинством этого метода является то, что он обеспечивает получение новой информации о свойствах объекта моделирования. Например, гелиоцентрическая модель солнечной системы построена на основе закона всемирного тяготения и законов механики Ньютона. Такая модель позволила установить наличие в солнечной системе неизвестной ранее планеты. Построенные таким образом математические модели используются для детального анализа процессов в элементах систем.

2. Построение модели объекта путем ее идентификации, т.е. чисто формальным путем с помощью статистической обработки результатов измерений без опоры на какие-либо знания о закономерностях процессов.

Суть метода состоит в том, чтобы по данным наблюдений за входными и выходными параметрами объекта построить такую математическую модель (математическую зависимость), которая описывала бы связь между этими параметрами. Как правило, заранее выбирается определенный вид математической **зависимости**. В этом случае при идентификации определению подлежат только параметры принятого математического описания. Такие модели используются при моделировании систем. Элементы систем моделируются предельно простыми формальными математическими зависимостями, которые называются «**черный ящик**» и описывают связь только между входными и выходными параметрами.

3. Построение модели системы на основе моделей элементов.

Обычно этот метод используется тогда, когда необходимо построить модель сложной системы на основе моделей ее элементов или когда из заданного набора элементов необходимо составить сложный объект и определить его свойства. Подобный подход используется в программном комплексе Simulink.

Первый путь реализуется при достаточной изученности общих закономерностей процессов, протекающих в моделируемом объекте. Параметры та-

ких моделей определяются либо на основе полуэмпирических зависимостей, либо на основе теории подобия, либо путем обработки данных экспериментов. Например, для применения закона всемирного тяготения в моделировании движения космических тел требуется экспериментальное определение гравитационной константы.

Недостатком аналитических моделей является сложность и нелинейность получающихся при этом уравнений. Достоинством является общность результатов моделирования и большая информативность моделей, способных предсказать новые неизвестные свойства изучаемых процессов и явлений.

Второй путь, который называется **экспериментальным** методом, применяется при отсутствии информации о механизмах процессов, слабой изученности либо сложности объекта моделирования. Этот путь используется при исследовании объекта в достаточно узком, «рабочем», диапазоне параметров. Подобные методы чаще всего основаны на предположении о линейности зависимостей и сосредоточенности параметров объекта. При таком подходе требуется проведение опытов непосредственно на самом изучаемом объекте. Достоинством экспериментального метода является простота получаемых моделей при достаточно точном описании свойств объекта в узком диапазоне изменения параметров. Однако экспериментальный метод не всегда позволяет распространить полученные результаты на другие однотипные объекты.

Сочетание обоих методов, т.е. аналитическое описание и экспериментальное определение неизвестных параметров модели, позволяет соединить сильные стороны каждого метода.

Третий путь характерен для имитационного моделирования сложных систем, когда исследователя интересуют свойства системы в целом.

В **главе 1** мы уже рассматривали общую схему построения моделей. Теперь проанализируем особенности построения математических моделей.

Первоначально любая задача моделирования строится как задача какой-либо предметной области либо конкретной области науки. Описание (модель) объекта выполняется на основе естественного языка, а затем на основе более строгого научного языка, в котором используется система базовых понятий не-

которой области науки. Естественно, постановка задачи должна содержать определение конечных целей моделирования, того набора свойств системы, взаимосвязи между которыми актуальны для решения данной задачи. Подобная модель носит название **концептуальной модели**.

Построение математического описания объекта всегда требует формализации задачи моделирования, т.е. ее описания в рамках какой-либо формальной системы на основе строгих и однозначных правил. Необходимо также формулирование исходных допущений относительно исследуемого явления и формулирование упрощающих предположений.

При построении математической модели необходимо **выбрать класс математических объектов**, которые в принципе могут отображать количественные характеристики моделируемого объекта. Естественно, что выбранный математический объект должен учитывать структуру и связи объекта моделирования, иметь такое количество параметров, которого будет достаточно для отражения главных свойств объекта моделирования. Значения параметров модели должны быть определены путем идентификации или с помощью привлечения полуэмпирических законов. Иначе математическая модель остается неопределенной.

После построения модели необходимо сопоставление модельных результатов и результатов наблюдений, т.е. оценка адекватности модели.

Одной из серьезнейших проблем математического моделирования является большое количество параметров. В этом случае систематизировать результаты моделирования и установить скрытые связи практически невозможно. Причиной этого является то, что к простым по форме фундаментальным законам при построении модели добавляются начальные и граничные условия, которые выделяют единственное решение. Физические свойства системы определяются физическими константами, которые должны быть определены дополнительно. Ход процессов зависит от взаимодействия объекта моделирования с окружающей средой, что порождает ряд дополнительных параметров, характеризующих краевые условия. Аналогичным образом требуется определить геометрические свойства объекта моделирования и т.д.

Большая размерность моделей вовсе не является их собственным свойством. Оказывается, что влияние факторов (параметров) на свойства системы проявляется не порознь, а в некоторой совокупности. То есть при анализе моделей следует рассматривать не отдельные параметры, а их комплексы. **Теория подобия** дает способы построения таких комплексов, которые основаны на анализе исходной задачи. Подобная процедура позволяет существенно (иногда полностью) сократить количество параметров модели (т.е. построить своеобразную обобщенную математическую модель) и уже на этом этапе выявить определенные закономерности и свойства процессов. Методы построения таких моделей представлены в следующих разделах.

Итак, мы познакомились с понятием математическая модель, со способами и особенностями построения математических моделей.

2.2. ПРИМЕРЫ ПОСТРОЕНИЯ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ

В данном разделе в качестве примеров построения математических моделей используются модели физических процессов. Это классическая область применения математического моделирования. Материал будет рассмотрен в рамках подхода, основанного на построении обобщенных безразмерных моделей и применении инструментальной системы моделирования для их реализации.

Математические модели многих объектов или процессов, как правило, имеют достаточно большое количество параметров. Это существенно затрудняет выявление каких-либо закономерностей. Построение безразмерной модели еще на этапе, предшествующем проведению вычислительных экспериментов с математической моделью, позволяет существенно повысить ее информативность и информативность результатов моделирования. Это следствие сокращения количества параметров и выявления безразмерных комплексов, которые и определяют свойства моделируемого объекта.

Построение безразмерной модели позволяет установить законы подобия. Именно при построении моделей технических объектов были впервые разработаны основные положения теории подобия, которые впоследствии нашли применение при решении многих задач. Рассмотрим общую методику построения безразмерных моделей, суть которой состоит в следующем:

1. **Построение модели.** Математическая модель строится в виде системы уравнений в размерной форме. Только в этом виде уравнения отражают «физическую» суть моделируемых процессов.

2. **Определение безразмерных переменных.** Все переменные (пространственные координаты, время, искомые переменные) представляются в безразмерной форме путем введения неопределенных масштабов.

3. **Преобразование модели к безразмерному виду.** Уравнения, краевые и начальные условия чисто алгебраическими методами преобразуются к безразмерному виду. В этом случае в уравнениях образуются безразмерные комплексы размерных параметров. В эти комплексы включены неопределенные масштабы.

4. **Определение масштабов.** Безразмерные комплексы, содержащие неопределенные масштабы, приравниваются к единице. Тем самым образуется система алгебраических уравнений, которая позволит получить выражения для неопределенных масштабов. Естественно, что число алгебраических уравнений, которые используются для определения масштабов, должно быть равно числу определяемых масштабов.

В оставшихся безразмерных комплексах масштабы теперь получают конкретные значения. Эти комплексы и образуют безразмерные параметры задачи. Таким, чисто формальным путем, могут быть получены естественные безразмерные параметры модели, которые в каждом конкретном случае имеют свой «физический» смысл.

Результаты исследования безразмерной модели распространяются на множество реальных объектов, для которых безразмерные параметры имеют одинаковое значение.

Построим модель движения тела под действием силы тяжести в среде с сопротивлением. Рассмотрим весь путь построения модели и принимаемые допущения. Обычно подобные задачи решаются без анализа всех необходимых допущений, которые определяют адекватность модели.

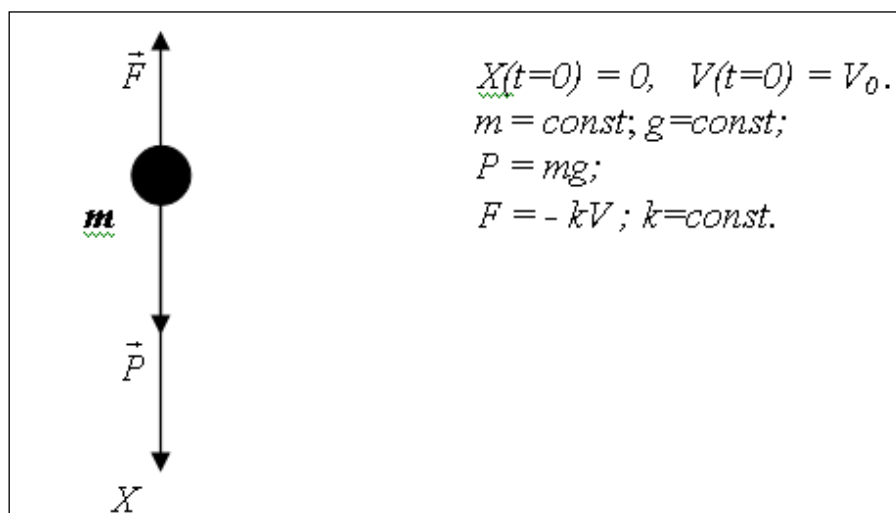


Рис. 2.1. Расчетная схема движения тела

Объект представляет собой тело, которое совершает прямолинейное движение под действием силы тяжести и при этом испытывает сопротивление движению со стороны окружающей среды. Расчетная схема процесса представлена на рис. 2.1.

Допущения, принятые при построении модели:

- Тело имеет правильную геометрическую форму, например шар, движение происходит прямолинейно под действием силы тяжести и силы сопротивления. Действительно, законы движения шарообразного тела и тела, имеющего форму пластины, имеют существенное различие.
- Масса тела – постоянная величина. Таким образом, модель не описывает, например, движение тел, вещество которых испаряется, сгорает или растворяется.
- В процессе движения форма тела не изменяется. Капля жидкости изменяет свою форму, для описания ее движения требуются более сложные модели.
- Плотность тела существенно выше плотности окружающей среды. Таким образом, силой Архимеда можно пренебречь.

- Вращение тела отсутствует. Таким образом, полет футбольного мяча данная модель в полной мере описать не в состоянии.

- Сила сопротивления линейно зависит от скорости движения тела: $F = -kV$. Данное допущение справедливо в определенном диапазоне скоростей движения тела. На практике часто реализуется квадратичный закон сопротивления: $F = -k \cdot V \cdot |V|$.

- Гравитационная сила – постоянная величина. Таким образом, модель не распространяется на описание процессов космического масштаба, например, движения космического летательного аппарата по околоземной орбите, полет метеорита.

Если перечисленные допущения выполняются, то тело можно считать **материальной точкой**, движение которой описывается законами классической механики Ньютона. Все допущения направлены на то, чтобы определить область корректного применения законов движения материальной точки. Однако определение значения коэффициента k возможно только путем идентификации этого параметра или с помощью полуэмпирических зависимостей.

На примере данной модели мы хотим подчеркнуть, что любая модель имеет свою область применения. Действительно, перечень допущений достаточно велик и невыполнение любого из них приведет к неадекватным результатам моделирования. Например, вращение тела при движении в среде с сопротивлением создает дополнительную силу (эффект Магнуса). Проявление данного эффекта можно наблюдать в ходе футбольного матча, когда мяч после углового удара залетает в ворота.

Цель моделирования: построить модель, на основе которой можно определить закон изменения скорости движения тела $V(t)$ и закон изменения координаты $X(t)$ во времени. В соответствии с принятыми допущениями, модель движения тела строится на основе второго закона механики. Система дифференциальных уравнений, описывающая движение тела, имеет вид:

$$m \frac{dV}{dt} = mg - kV, \quad \frac{dx}{dt} = V.$$

Начальные условия: $V(t=0)=V_0$; $X(t=0)=0$.

Здесь m – масса, g – ускорение свободного падения, k – коэффициент сопротивления, V – скорость, t – время, x – координата.

Данная модель имеет четыре параметра: m, k, g, V_0 , что существенно затрудняет анализ. Для упрощения анализа результатов моделирования необходимо свести количество параметров к минимуму.

Преобразуем модель к безразмерному виду следующим образом (обе части уравнения разделим на mg):

$$\frac{1}{g} \frac{dV}{dt} = 1 - \frac{k}{mg} V.$$

Определим безразмерную скорость \bar{V} как отношение текущего значения скорости V к ее начальному значению V_0 : $\bar{V} = V / V_0$ и с учетом этого соотношения преобразуем уравнение движения:

$$\frac{V_0}{g} \frac{d\bar{V}}{dt} = 1 - \frac{kV_0}{mg} \bar{V}.$$

Обозначим: $t^* = V_0 / g$ – характерное время процесса. Введем безразмерное время $\bar{t} = t / t^*$. Тогда уравнение движения примет вид:

$$\frac{d\bar{V}}{d\bar{t}} = 1 - \left(\frac{kV_0}{mg}\right) \bar{V}.$$

Обозначим $\bar{k} = \frac{kV_0}{mg}$ – безразмерный коэффициент сопротивления. С учетом принятых обозначений уравнение движения запишем в следующей форме:

$$\frac{d\bar{V}}{d\bar{t}} = 1 - \bar{k} \bar{V}.$$

Кинематическое уравнение преобразуется аналогичным образом:

$$\frac{1}{V_0 t^*} \frac{dx}{d\bar{t}} = \bar{V}, \quad \bar{x} = x / (V_0 t^*).$$

В итоге получим систему безразмерных дифференциальных уравнений:

$$\frac{d\bar{V}}{d\bar{t}} = 1 - \bar{k}\bar{V}, \quad \frac{d\bar{x}}{d\bar{t}} = \bar{V}.$$

$$\bar{x}(\bar{t} = 0) = 0, \quad \bar{V}(\bar{t} = 0) = 1.$$

Начальные условия:

После преобразований задача приведена к безразмерному виду и имеет всего один **безразмерный** параметр \bar{k} . Естественно, анализ свойств подобной модели проводить значительно проще, чем исходной.

При анализе модели в исходном размерном виде, задав конкретные значения параметров, мы установим свойства лишь единственной конкретной системы. Анализ модели в безразмерной форме для заданного значения \bar{k} дает информацию о свойствах бесконечного числа реальных систем, для которых:

$$\bar{k} = \frac{kV_0}{mg} = const.$$

Различные реальные системы, имеющие одинаковые значения параметра \bar{k} называются **подобными**, а параметр \bar{k} для данной задачи называется критерием подобия.

В частном случае, когда $V_0=0$, можно получить безразмерную модель с нулевым количеством параметров. Такая система называется **автомодельной**. В этом случае все реальные системы подобны друг другу.

Таким образом, проделанные предварительные преобразования существенно повысили информативность модели и упростили дальнейший вычислительный эксперимент. Построенная в среде RMD компьютерная модель показана на рис. 2.2.

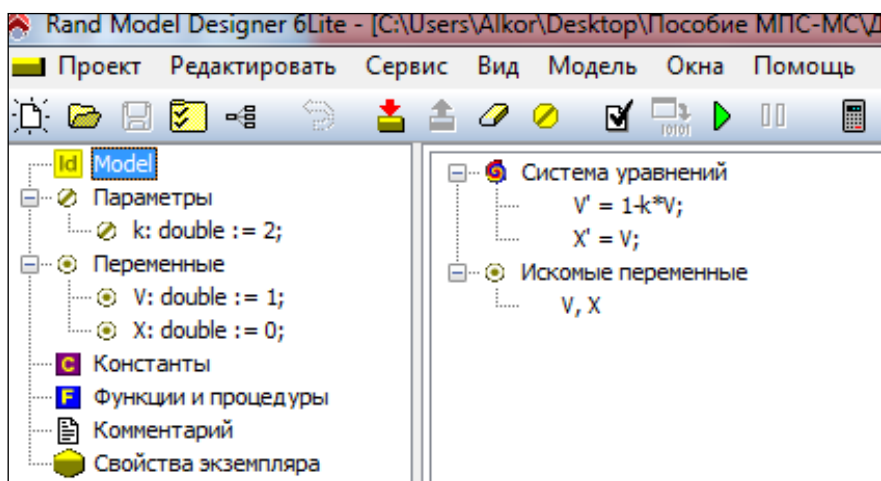


Рис. 2.2. Компьютерная RMD модель движения тела в среде с сопротивлением

На примере рассматриваемых в данном разделе задач показан весь цикл построения математической модели с момента принятия упрощающих допущений до получения уравнений, описывающих моделируемые процессы, и компьютерной реализации модели в среде инструментальной системы моделирования. Инструментальная система моделирования RMD существенно упрощает построение компьютерной модели, т.к. выбор численного метода, генерация программного и исполняемого кодов производится автоматически средствами системы. Результаты моделирования даны на рис. 2.3.

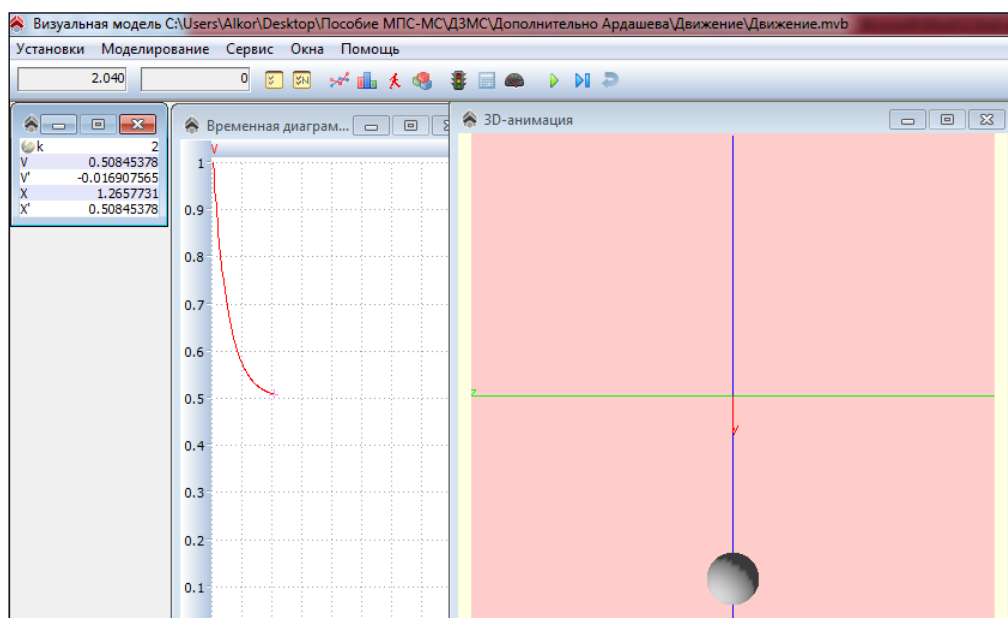


Рис. 2.3. Результаты моделирования движения тела (RMD) в среде с сопротивлением

В качестве примера построения модели рассмотрим преобразование к безразмерному виду модели динамической системы, которая представляет со-

бой тело массой m , совершающее колебания под действием упругой пружины и сил трения (рис. 2.4).

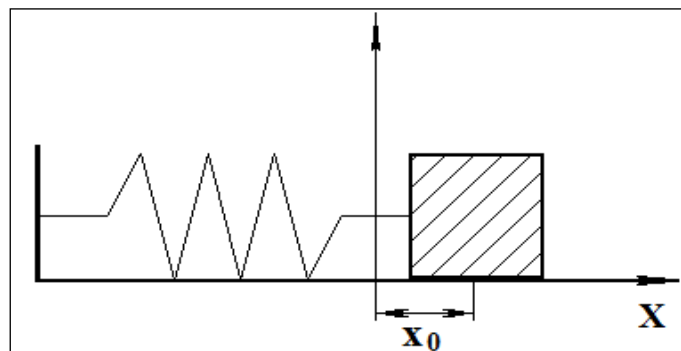


Рис. 2.4. Схема объекта исследования

Исходная размерная математическая модель, записанная в виде системы дифференциальных уравнений, имеет следующий вид:

$$\begin{cases} m \frac{dV}{dt} = -cx - kV, \\ \frac{dx}{dt} = V; \end{cases}$$

$$x(t=0) = x_0, \quad V(t=0) = 0,$$

где m – масса тела, x – координата тела, V – его скорость, c – жесткость пружины, k – коэффициент трения. Значение $x=0$ соответствует положению равновесия.

Определим безразмерные переменные следующими соотношениями:

$$\bar{x} = \frac{x}{x^*}, \quad \bar{V} = \frac{V}{V^*}, \quad \bar{t} = \frac{t}{t^*}.$$

Здесь x^* , V^* , t^* – неопределенные масштабы (положительные константы), x , V , t – размерные переменные. Преобразуем модель к безразмерному виду путем замены размерных переменных:

$$x = \bar{x} \cdot x^*, \quad V = \bar{V} \cdot V^*, \quad t = \bar{t} \cdot t^*.$$

Тогда уравнения и начальные условия примут вид:

$$\frac{V^* m}{t^*} \frac{d\bar{V}}{d\bar{t}} = -x^* c \cdot \bar{x} - kV^* \bar{V}; \quad \frac{x^*}{t^*} \frac{d\bar{x}}{d\bar{t}} = V^* \bar{V},$$

или

$$\frac{V^* m}{x^* t^* c} \frac{d\bar{V}}{d\bar{t}} = -\bar{x} - \frac{kV^*}{cx^*} \bar{V}; \quad \frac{x^*}{V^* t^*} \frac{d\bar{x}}{d\bar{t}} = \bar{V};$$

$$\bar{x}(\bar{t} = 0) = x_0 / x^*, \quad \bar{V}(\bar{t} = 0) = 0.$$

Масштабы x^*, y^*, t^* выберем таким образом, чтобы

$$\frac{V^* m}{x^* t^* c} = 1, \quad \frac{x^*}{V^* t^*} = 1, \quad x_0 / x^* = 1.$$

Тогда

$$t^* = \sqrt{\frac{m}{c}}, \quad x^* = x_0, \quad V^* = x_0 \sqrt{\frac{c}{m}}$$

и система уравнений окончательно будет иметь следующий безразмерный вид:

$$\frac{d\bar{V}}{d\bar{t}} = -\bar{x} - \bar{k}\bar{V}, \quad \frac{d\bar{x}}{d\bar{t}} = \bar{V}.$$

Здесь $\bar{k} = \frac{k}{\sqrt{mc}}$ – безразмерный параметр, безразмерный коэффициент трения. А начальные условия таковы: $\bar{x}(\bar{t} = 0) = 1, \quad \bar{V}(\bar{t} = 0) = 0.$

Проделанные преобразования показали, что поведение и свойства системы, которая моделируется (рис. 2.4) полностью определяются значением безразмерного параметра \bar{k} . Таким образом, результаты исследования модели при конкретном значении параметра \bar{k} будут представлять свойства целой группы **подобных** явлений, имеющих такие же значения безразмерного параметра \bar{k} .

Применение данной методики позволяет свести к минимуму количество параметров, что существенно упрощает задачу моделирования. Данная процедура может быть выполнена разными способами. В некоторых случаях в качестве значений масштабов могут быть выбраны характерные для конкретного объекта или процесса значения параметров. Однако такой подход в общем случае не позволяет минимизировать число безразмерных параметров. В данной задаче при отсутствии трения ($\bar{k} = 0$) все системы подобны друг другу (автономные системы).

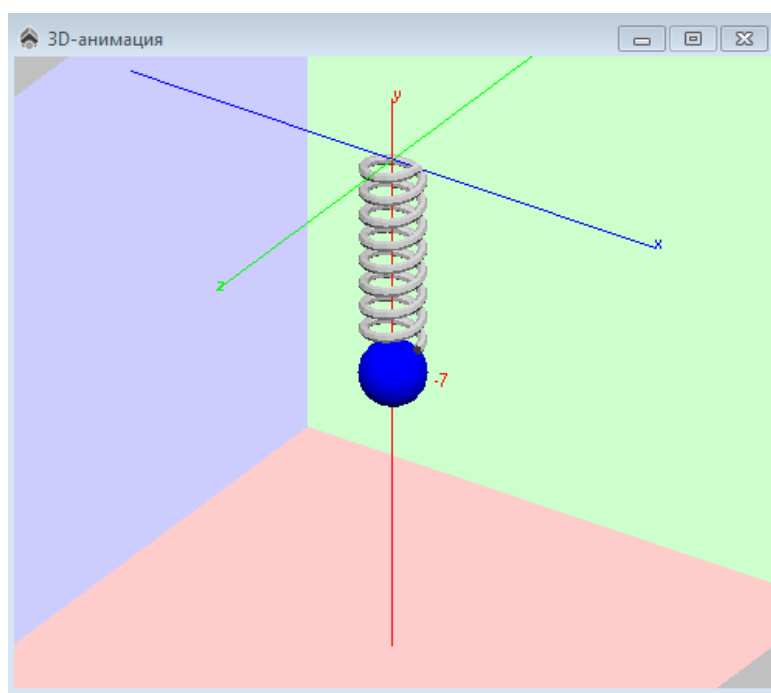


Рис. 2.5. 3D-анимация модели динамической системы по рис. 2.4

В итоге получается **обобщенная безразмерная модель**, которая сама по себе лишена всякого физического смысла. Физический смысл результатов моделирования истолковывается только через интерпретацию безразмерных параметров и переменных. При проведении вычислительного эксперимента на основе подобных моделей во многих случаях порядок величин и порядок вычислительных ошибок может быть оценен заранее.

Таким образом, представленный метод позволяет выявить важное для моделирования свойство объектов и процессов – явление **подобия**. В рамках

рассмотренной модели подобными будут считаться все процессы, имеющие одинаковые значения безразмерных параметров.

Утверждение о подобии явлений есть констатация определенной общности их свойств. Исторически понятие подобия сложилось на основе сравнительного анализа явлений одной и той же физической природы. Установлено, что результаты, полученные при анализе единичного явления (объекта или процесса), могут быть по определенным правилам перенесены на множество других явлений. Изучение и установление законов подобия явлений самой разнообразной природы есть предмет исследований **теории подобия**.

Для того чтобы модель замещала в каком-то отношении оригинал, между оригиналом и моделью должно быть установлено отношение подобия. Подобие может устанавливаться в процессе создания самой модели. Примерами таких отображений являются фотографии, масштабные модели самолетов, кораблей или гидротехнических сооружений, макеты зданий и т.п. Такое подобие называется прямым. Понятие подобия заимствовано из геометрии и означает существование определенных масштабных соотношений для параметров сопоставляемых геометрических объектов.

Понятие подобия физических процессов родственно понятию геометрического подобия. В этом случае под подобием понимается такое взаимно однозначное соответствие между сопоставляемыми объектами или процессами, при котором правила перехода от параметров одного объекта к параметрам другого объекта известны или заданы.

Замечательным свойством, окружающего нас мира, является то, что модели процессов и явлений самой разнообразной природы сводятся к тождественным по форме безразмерным математическим моделям. Таким образом, по аналогии результаты исследования одного явления могут быть перенесены на явление совершенно другой физической природы.

Наиболее известными примерами являются электротепловая аналогия и электромеханическая аналогия. Оказывается, что закономерности электрических и механических процессов, электрических и тепловых процессов описываются одинаковыми уравнениями. Различие состоит лишь в разной физиче-

ской интерпретации переменных и параметров, входящих в эти уравнения. Таким образом, по аналогии можно применить результаты исследования математических моделей на физически различные процессы, которые описываются аналогичными математическими моделями.

Таким образом, выявление законов подобия при моделировании систем различной природы показывает, как наиболее информативно представить результаты модельных экспериментов и выявить закономерности процессов.

2.3. МЕТОДЫ ИССЛЕДОВАНИЯ МОДЕЛЕЙ, ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ

Построение математической модели и преобразование ее к безразмерному виду позволяет уже на этой стадии получить определенную информацию о свойствах решения и, следовательно, объекта моделирования. Существует целый класс качественных математических методов, предназначенных для выявления свойств объекта или процесса без решения уравнений, составляющих математическую модель этого объекта. Они позволяют установить свойства решения только на основе анализа параметров модели. Однако во многих случаях получение решения уравнений математической модели необходимо.

Используемые в настоящее время методы исследования различного рода математических моделей в прикладных областях можно разделить на следующие виды: точные методы; асимптотические методы; приближенные методы; численные методы.

Под **точными методами** понимаются методы, которые позволяют получить решение исходной задачи в **аналитическом виде**. Такие методы применимы для решения достаточно простых линейных задач с постоянными коэффициентами. Достоинством точных аналитических решений является их наглядность, компактность и большая информативность.

Для решения современных задач моделирования такие методы практически не применяются. Однако точные решения могут использоваться в качестве тестовых задач при разработке других методов (приближенных и числен-

ных). Технические, экономические, экологические и другие системы, изучаемые современной наукой, не поддаются исследованию в необходимой для практики полноте аналитическими методами. Это обстоятельство обусловлено сложностью и нелинейностью их математических моделей. Для нелинейных задач большим достижением считается получение даже частных решений.

Изучение любого явления начинается с определения **основных закономерностей** его функционирования на качественном уровне. При этом уравнения модели часто настолько сложны, что необходимо построение упрощенных моделей, без которых невозможно выявить механизмы процессов и составить их ясное понимание.

В ряде случаев упрощение достигается за счет того, что рассматривается такой вариант задачи, в котором в уравнениях удастся выделить **«малый»** параметр. В этом случае решение представляется в виде разложения в ряд по «малому» параметру. Подобные методы носят название **асимптотических**, или методов возмущений, и широко применяются в гидродинамике, квантовой механике и т.д.

До сих пор сохраняют свое значение **приближенные методы**, которые опираются на неформальное понимание сути процессов. Приближенные методы удобны для получения грубых оценок на предварительном этапе исследования. Они играют большую роль в получении качественных представлений и часто используются, например, в инженерной практике, когда исходная задача имеет приближенную постановку и значения параметров определены с точностью до порядка. В качестве примера приближенных методов можно привести решение нелинейной задачи по линеаризованной модели, решение задачи с переменными параметрами по алгоритмам для задач с постоянными коэффициентами и т.д.

Как бы ни были разнообразны методы приближенного или качественного анализа математических моделей, область их применения ограничена. Это либо достаточно простые модели, либо упрощенные фрагменты сложных нелинейных моделей.

Как уже отмечалось, математические модели реальных объектов являются нелинейными. Для проведения экспериментов с ними, то есть для получения информации об их свойствах, требуется компьютерная реализация математических моделей на основе **численных методов**. Это единственный достаточно универсальный способ исследования математических моделей с помощью средств компьютерной техники. Поэтому современное математическое моделирование всегда предполагает применение численных методов анализа и проведение компьютерных вычислительных экспериментов.

Численные решения являются всегда приближенными и имеют дискретный характер. Доступный для компьютера вычислительный алгоритм должен удовлетворять достаточно жестким требованиям. К ним относятся, прежде всего, необходимость получить решение с заданной точностью за разумное время. Объемы обрабатываемой информации при этом не должны превышать возможностей компьютера.

Проблемы численного моделирования не решаются сами собой при развитии вычислительной техники, так как постоянно происходит усложнение задач, выдвигаемых теорией и практикой, существует необходимость проведения большого количества серий вычислительных экспериментов для более полного изучения объекта. Поэтому разработка и применение эффективных вычислительных методов остается одной из ключевых задач математического моделирования.

Для математических моделей, представленных дифференциальными уравнениями, процесс создания вычислительных алгоритмов состоит из построения **дискретного алгебраического аналога** исходных уравнений и его численного решения. Дискретный аналог позволяет приближенно определить решение уравнений в конечном числе фиксированных точек по времени или по пространственным координатам.

Численные методы позволяют получить решения в областях, где другие методы не применимы. Однако огромный потенциал численных методов не делает их всесильным средством. К недостаткам численных методов можно отнести слабую наглядность и информативность результатов по сравнению с

аналитическими и приближенными решениями, трудоемкость реализации, трудности применения в разрывных точках, отсутствие строгого обоснования корректности вычислительных процедур для сложных нелинейных задач, проявление эффектов конечной разрядной сетки кодирования чисел. Любой численный метод привносит в решение задачи количественную погрешность, а в ряде случаев и **качественные эффекты**, распознать природу которых весьма затруднительно.

Рассмотрим основные характеристики численных методов, актуальные для моделирования:

Устойчивость метода. Неустойчивый численный метод приводит к накоплению и неограниченному росту вычислительных ошибок. Характерный признак неустойчивости вычислений – решение имеет пилообразный характер с быстро растущей амплитудой (рис. 2.6).

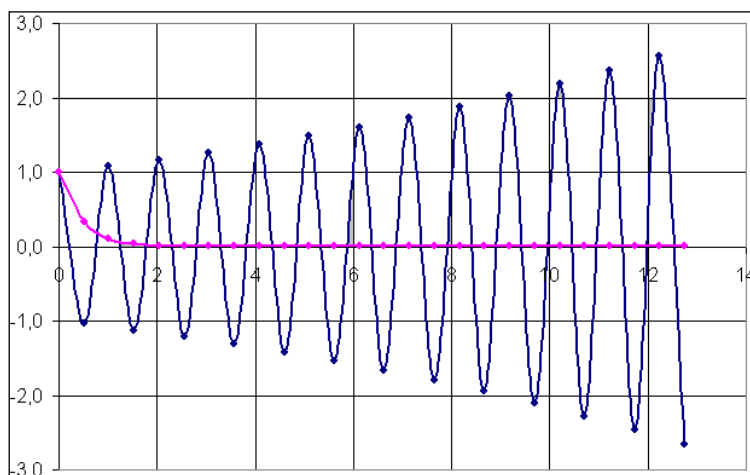


Рис. 2.6. Результаты решения задачи устойчивым и неустойчивым методами

Естественно, что неустойчивые методы применять нельзя. Существуют **условно и безусловно** устойчивые методы. Условно устойчивый метод это такой метод, когда устойчивое решение получается при выполнении определённого условия для параметров вычислений. Для безусловно устойчивого метода устойчивость вычислений обеспечивается при любых сочетаниях вычислительных параметров.

Степень точности метода (порядок метода). Любой численный метод решения, например, дифференциальных уравнений имеет основной характерный параметр. Обычно степень точности метода указывается в виде зависимости погрешность метода от величины этого параметра, например: $\delta \sim h^2$, где δ – погрешность метода, h – шаг вычисления. Такой метод называется методом второго порядка точности, если шаг вычислений уменьшить в 2 раза, то погрешность метода уменьшится в 4 раза.

Выбор численного метода при реализации математической модели имеет большое значение. Так как любой численный метод является приближенным, то результаты моделирования будут содержать определенную заранее неизвестную ошибку. Существует множество научных публикаций, в которых проявление свойств численных методов ошибочно принималось в качестве свойств объекта моделирования.

Следует иметь в виду, что выбор численного метода определяется задачей моделирования исходя из принципа **«оптимальной неточности»**. Это означает, что в задачах с неточными исходными данными или при качественном анализе процессов применение методов высокой степени точности ничем не оправдано. В большинстве случаев результат может быть получен на основе простейших методов.

Сходимость метода. Сходимость – теоретическая характеристика метода, которая констатирует тот факт, что в пределе метод может дать точное решение при стремлении его основного параметра к нулю (или бесконечности). В практике численного моделирования играет большую роль **скорость сходимости** метода, т.е. скорость приближения к «точному» значению. Например, при решении нелинейных алгебраических уравнений наибольшую скорость сходимости имеет метод Ньютона.

Во многих случаях математические модели представляют собой системы дифференциальных уравнений. Суть методов интегрирования дифференциальных уравнений рассмотрим на примере обыкновенного дифференциально-

го уравнения первого порядка $\frac{dy}{dt} = f(t, y)$ с начальными условиями $y(t=0) = y_0$ (задача Коши). Здесь t – время.

Численное интегрирование данного дифференциального уравнения сводится к вычислению дискретного ряда значений переменной y : $y_0, y_1, y_2 \dots y_k$ для заданных значений переменной t : $t_0, t_1, t_2 \dots t_k$. В большинстве случаев интегрирование проводится с постоянным шагом по времени: $\tau = t_{j+1} - t_j = const$.

Пусть в момент времени t_n значение $y_n = y(t_n)$ нам известно. Такая точка всегда имеется, т.к. для дифференциального уравнения задано начальное условие $y_0 = y(t=0)$. Необходимы расчетные соотношения для получения значения y в момент времени t_{n+1} : $y_{n+1} = y(t_{n+1})$. Проинтегрируем данное дифференциальное уравнение от t_n до t_{n+1} :

$$\int_{t_n}^{t_{n+1}} \frac{dy}{dt} dt = \int_{t_n}^{t_{n+1}} f(t, y) dt$$

Это соотношение дает основную вычислительную формулу:

$$y_{n+1} - y_n = \int_{t_n}^{t_{n+1}} f(t, y) dt$$

В последней формуле для конкретного численного метода интегрирования необходимо выбрать какой-либо способ вычисления определенного интеграла в правой части последнего соотношения.

В явном методе Эйлера значение интеграла в правой части основной формулы вычислим по формуле прямоугольников (рис. 2.7):

$$y_{n+1} - y_n \approx f(t_n, y_n) \cdot \tau$$

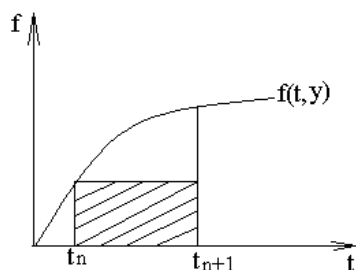


Рис. 2.7. Применение формулы прямоугольников в явном методе Эйлера

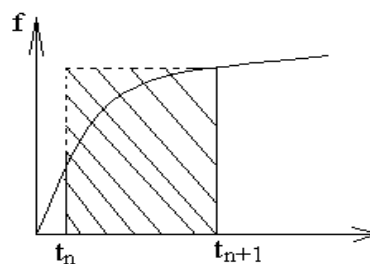


Рис. 2.8. Применение формулы прямоугольников в неявном методе Эйлера

При реализации данного метода значение функции $f(t, y)$ берется в момент времени t_n . При этом значение y_n известно. Таким образом, значение функции f может быть вычислено. Явный метод Эйлера – метод первого порядка точности, он имеет жесткие условия устойчивости. При решении многих практически значимых задач обеспечение устойчивости требует малых значений шага интегрирования τ . Поэтому явный метод Эйлера применяется редко.

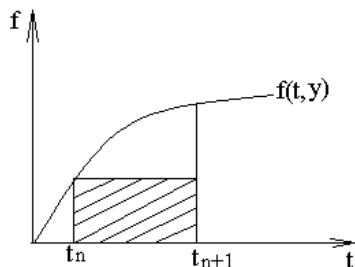


Рис. 2.7. Применение формулы прямоугольников в явном методе Эйлера

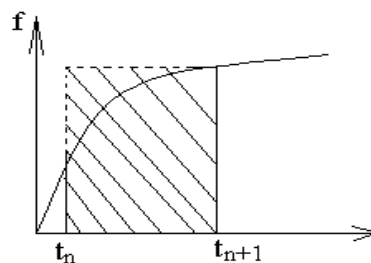


Рис. 2.8. Применение формулы прямоугольников в неявном методе Эйлера

В неявном методе Эйлера интеграл в правой части основной формулы вычисляется по методу прямоугольников (рис. 2.8):

$$y_{n+1} - y_n \approx f(t_{n+1}, y_{n+1}) \cdot \tau.$$

Вычислительная формула неявного метода реализуется элементарно, если функция $f(t, y)$ линейная. Тогда последняя формула легко преобразуется в явное вычислительное соотношение для y_{n+1} . В случае нелинейной функции f значение y_{n+1} вычисляется с применением интеграций, например, по методу

Ньютона. Неявный метод Эйлера абсолютно устойчив, однако имеет первый порядок точности. Тем не менее данный метод достаточно широко применяется в силу своей безусловной устойчивости. Таким образом, при использовании данного метода выбор шага интегрирования τ производится только из соображений обеспечения необходимой точности вычислений.

В методе Эйлера-Коши для вычисления значения интеграла в правой части основной формулы используется метод трапеций (рис. 2.9):

$$y_{n+1} - y_n \approx (f(t_{n+1}, y_{n+1}) + f(t_n, y_n)) \cdot \tau / 2.$$

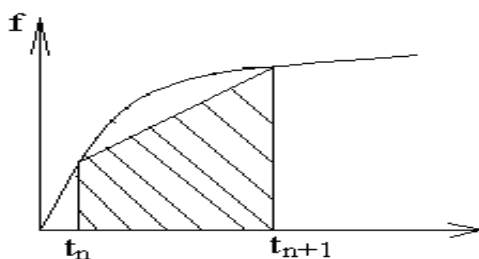


Рис. 2.9. Применение формулы трапеций в методе Эйлера-Коши

Данный метод имеет второй порядок точности и практически устойчив. Так как в правой части содержится неизвестная величина y_{n+1} , то метод реализуется в два этапа:

1. На первом этапе применяется явный метод Эйлера первого порядка точности и вычисляется вспомогательное значение \tilde{y}_{n+1} :

$$\tilde{y}_{n+1} = y_n + f(t_n, y_n) \cdot \tau.$$

2. На втором этапе решение уточняется по формуле трапеций:

$$y_{n+1} = y_n + \frac{\tau}{2} (f(t_{n+1}, \tilde{y}_{n+1}) + f(t_n, y_n)).$$

Практическое применение численных методов при решении задач моделирования рассмотрим на примере численной реализации модели прямолинейного движения тела под действием силы тяжести в среде с сопротивлением. Данная модель представляется системой дифференциальных уравнений в безразмерном виде:

$$\frac{d\bar{V}}{d\bar{t}} = 1 - k\bar{V}, \quad \frac{d\bar{x}}{d\bar{t}} = \bar{V}.$$

Начальные условия: $\bar{x}(\bar{t} = 0) = 0$, $\bar{V}(\bar{t} = 0) = 1$.

Эту задачу решим неявным методом Эйлера первого порядка. Применив соответствующую формулу неявного метода Эйлера, получим:

$$V_{n+1} = V_n + (1 - kV_{n+1}) \cdot \tau, \quad V_{n+1} + kV_{n+1} \cdot \tau = V_n + \tau,$$

$$V_{n+1}(1 + k\tau) = V_n + \tau, \quad V_{n+1} = \frac{V_n + \tau}{1 + k\tau}.$$

Напоминаем, что в момент времени t_n значения переменных V_n , x_n , считаются известными, значения переменных V_{n+1} и x_{n+1} в момент времени t_{n+1} требуется определить. Для расчета значения координаты x_{n+1} используем метод трапеций, т. к. значения скорости уже известны и x_{n+1} рассчитывается

независимо от V_{n+1} : $x_{n+1} = x_n + \frac{\tau}{2}(V_n + V_{n+1})$.

На рис. 2.10 представлены решения задачи для различных значений параметра k .

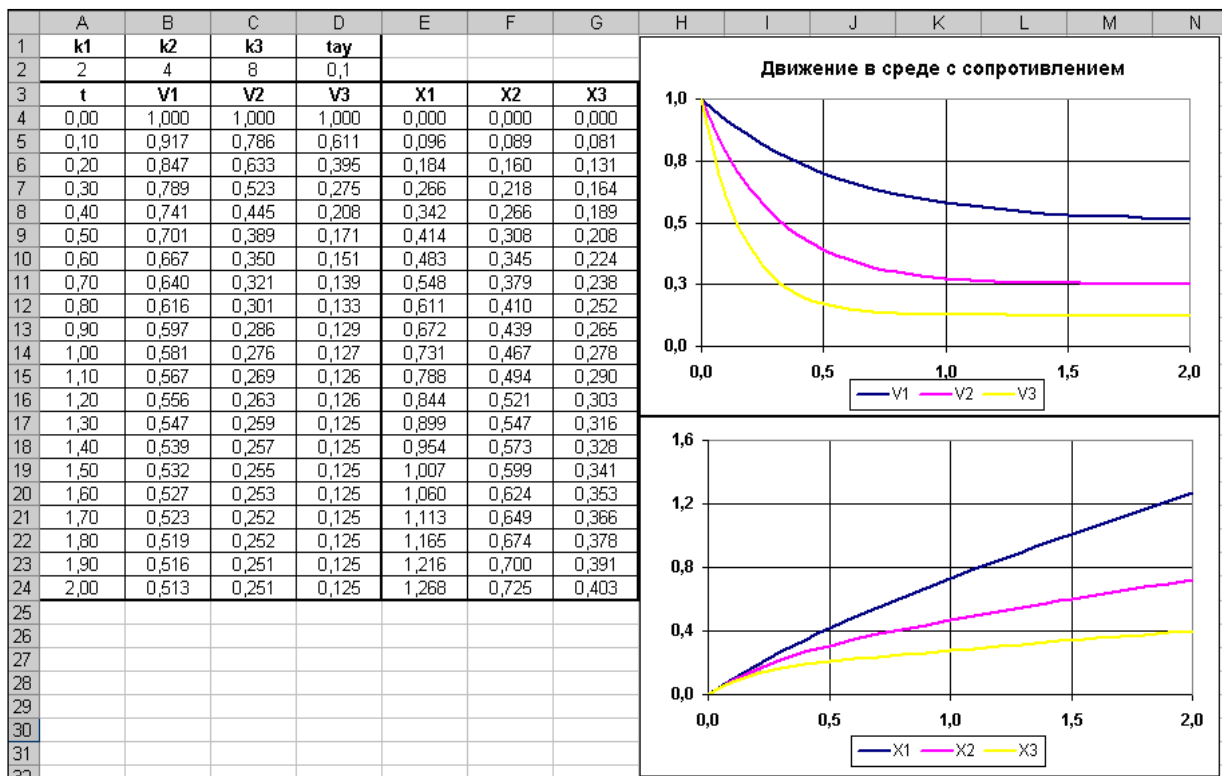


Рис. 2.10. Решение задачи моделирования в среде электронных таблиц

Точность полученного решения можно проверить простым практическим путем двойного решения задачи – с шагом τ и с шагом $\tau/2$ – или сравнением численных решений разного порядка точности. Сравнение решений позволяет оценить правильность выбора шага интегрирования τ . Если решения заметно отличаются, то шаг необходимо уменьшить, если решения совпадают с достаточной точностью, то шаг интегрирования выбран верно.

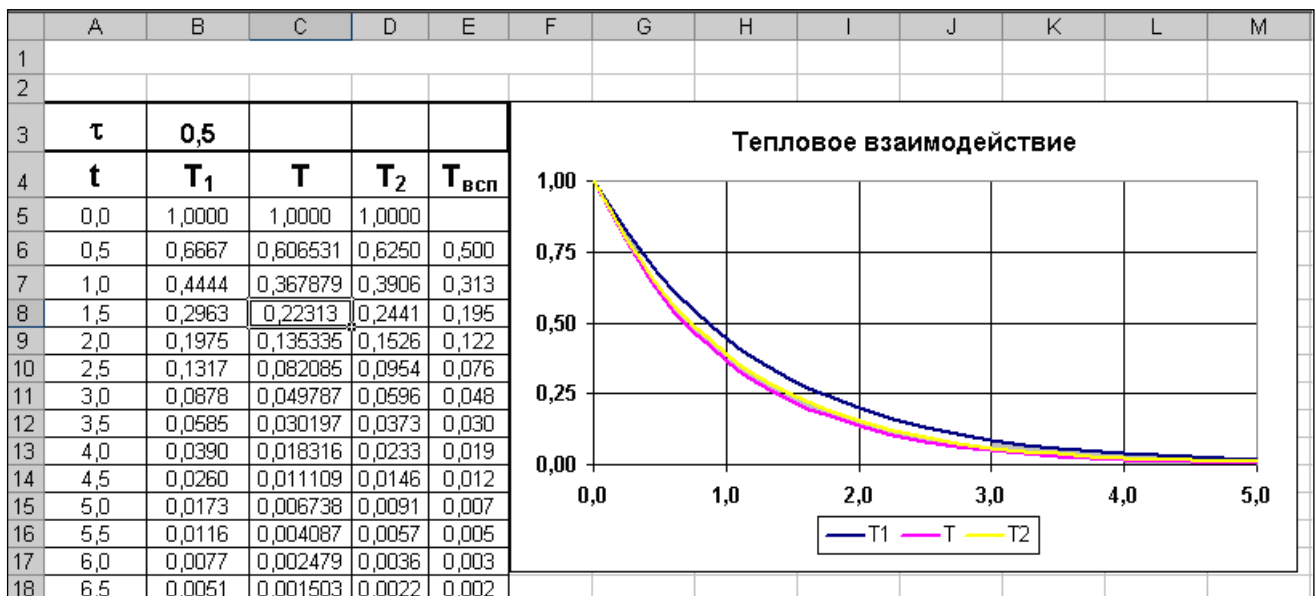


Рис. 2.11. Решение задачи различными методами

Здесь T – аналитическое решение; T_1 – решение методом Эйлера; T_2 – решение методом Эйлера-Коши. Вспомогательное решение T^{*6+1} на рис. 2.11 обозначено как $T_{всп}$.

На рис. 2.11 представлено решение задачи теплового взаимодействия нагретого тела различными численными методами. Задача моделирования представлена уравнением:

$$\frac{d\bar{T}}{dt} = -\bar{T}, \quad \bar{T}(t=0) = 1.$$

Она имеет точное аналитическое решение: $T(t) = -\exp(t)$. Решение методом Эйлера проводилось на основе соотношений:

$$\frac{T^{n+1} - T^n}{\tau} = -T^{n+1}, \quad T^{n+1} = \frac{T^n}{1 + \tau}.$$

Метод Эйлера-Коши для данной задачи реализуется в два этапа:

1. $T^{*6+1} = T^n - \tau \cdot T^n$;
2. $T^{n+1} = T^n - \frac{\tau}{2}(T^n + T^{*6+1})$.

Известны и другие, более точные методы, например, метод Рунге-Кутты. Однако в большинстве случаев простые методы способны дать вполне приемлемое решение.

2.4. МОДЕЛИ ПРОЦЕССОВ С РАСПРЕДЕЛЕННЫМИ ПАРАМЕТРАМИ

Многие процессы описываются моделями с **распределенными параметрами**. Такие параметры представляют собой некоторое множество (поле) значений, и усреднить их (свести к единственному значению) зачастую не представляется возможным. Использование при моделировании более простых моделей с сосредоточенными параметрами весьма грубое приближение. Примерами таких процессов является распространение волн, течение газов и жид-

костей, передача тепла, диффузия и другие процессы, протекающие в **сплошных средах**.

Подобные модели строятся на основе дифференциальных уравнений в **частных производных**. Исследования процессов в сплошных средах компьютерное численное моделирование позволило достичь существенного прогресса и перейти от решения простых классических задач математической физики к анализу сложнейших процессов различной физической природы, представляющих интерес для науки и техники. Родилось целое научное направление – **вычислительная физика**.

Далее мы рассмотрим подобные процессы, модели и методы их численного исследования. Анализ всех рассматриваемых моделей завершается описанием вычислительных алгоритмов, по которым можно провести вычислительный эксперимент.

Модель переноса тепла отражает процесс переноса тепла жидкостью (теплоносителем). Аналогичные модели строятся и для процессов переноса вещества движущейся средой.

Пусть имеется теплообменник, представляющий собой канал, обогрев которого производится путем пропускания электрического тока через его стенки. Примем следующие предположения: в теплообменник поступает холодная жидкость с известной температурой T_0 ; скорость движения жидкости внутри канала считается неизменной; свойства теплоносителя (плотность, теплоемкость) слабо зависят от его температуры и могут считаться неизменными. Температура теплоносителя по сечению канала считается одинаковой.

В начальный момент времени $t = 0$ включается обогрев канала, т. е. $q=0$, при $t < 0$ и $q > 0$, при $t > 0$. Здесь q – количество тепла, которое передается через стенки канала к теплоносителю в единицу времени через единицу площади поверхности канала.

Расчетная схема объекта моделирования представлена на рис. 2.12.

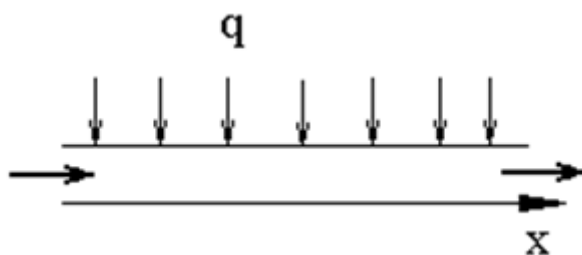


Рис. 2.12. Расчетная схема объекта исследования

До начала обогрева теплообменник находится в **стационарном** тепловом режиме. После включения обогрева теплообменник через некоторое время выйдет на новый стационарный режим. Задачей моделирования является установить **закон** изменения температуры теплоносителя в канале теплообменника как функцию координаты и времени – $T(t,x)$.

С учетом принятых допущений динамика изменения температуры теплоносителя в теплообменнике может быть описана следующим уравнением, которое является следствием закона **сохранения энергии**:

$$S(\rho c \frac{\partial T}{\partial t} + c \rho u \frac{\partial T}{\partial x}) = q\Pi$$

$$S = \pi R^2 \quad \Pi = 2\pi R.$$

Здесь S – площадь поперечного сечения канала теплообменника, Π – его периметр, ρ – плотность теплоносителя, c – его теплоемкость, u – скорость движения теплоносителя по каналу теплообменника. Таким образом, через выделенный элемент канала идет поток теплоносителя, который нагревается за счет подвода тепла через стенки канала (рис. 2.13).

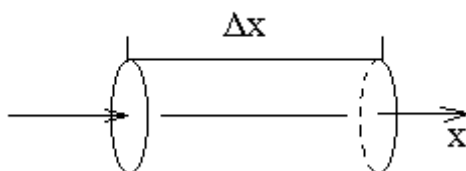


Рис. 2.13. Элемент канала

Физический смысл уравнения состоит в том, что его левая часть отражает изменение запаса тепловой энергии в элементе канала длиной Δx (рис. 2.13), а

правая часть уравнения учитывает количество тепла, которое подведено к элементу канала через боковую поверхность за счет обогрева. Левая часть уравнения содержит частную производную температуры по времени, которая собственно и отражает изменение во времени температуры в выделенном элементе канала, а также частную производную температуры по координате, которая отражает изменение запаса тепла в выделенном элементе вследствие движения теплоносителя через выделенный элемент (рис. 2.13).

Вид уравнения, соответствующий его физическому смыслу, представлен ниже:

$$\Delta x \cdot S \left(\rho c \frac{\partial T}{\partial t} + \rho c u \frac{\partial T}{\partial x} \right) = q \Pi \cdot \Delta x.$$

Простейшее преобразование исходного уравнения дает следующее:

$$\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} = \frac{q \Pi}{S \cdot \rho c} = Q$$

Пусть L – длина канала, выберем этот параметр в качестве масштаба для продольной координаты. В качестве масштаба времени выберем время прохождения теплоносителя по каналу L/u . Конечно, мы могли бы выполнить преобразование уравнения к безразмерному виду чисто формальным путем, однако осмысленный выбор масштабов в данном случае более прозрачен с физической точки зрения. С учетом этого безразмерные соотношения для продольной координаты и времени представим в следующем виде:

$$\bar{x} = \frac{x}{L} \quad \bar{t} = t / \left(\frac{L}{u} \right)$$

Преобразования уравнения дают соотношения:

$$\frac{\partial T}{\partial t} + \frac{u}{L} \frac{\partial T}{\partial \bar{x}} = Q; \quad \frac{L}{u} \frac{\partial T}{\partial t} + \frac{\partial T}{\partial \bar{x}} = Q \frac{L}{u}; \quad \frac{\partial T}{\partial \bar{t}} + \frac{\partial T}{\partial \bar{x}} = \frac{L}{u} Q$$

Здесь $Q = q \Pi / (S \rho c)$. Теперь каждый член уравнения имеет размерность температуры. В качестве ее безразмерного аналога выберем переменную y , которую определим следующей зависимостью:

$$y = (T - T_0) / \left(\frac{L}{u} Q \right).$$

Представление для безразмерной температуры обусловлено упрощением уравнения и нормированием переменной y . Теперь диапазон изменения y находится в пределах от 0 до 1. После простейших преобразований безразмерное уравнение процесса переноса тепла примет вид:

$$\frac{\partial y}{\partial t} + \frac{\partial y}{\partial x} = 1.$$

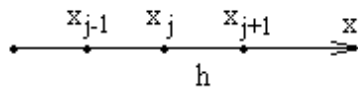
Краевые и начальные условия, с учетом преобразований, задаются следующими соотношениями:

$$y(t, x=0) = 0 \quad y(t=0, x) = 0.$$

Для численного решения данной начально-краевой задачи применим метод конечных разностей. Идея данного метода состоит в том, что дифференциальные операторы $\frac{\partial}{\partial t}$ и $\frac{\partial}{\partial x}$ заменяются их приближенными конечно-разностными аналогами. Методом конечных разностей будет построено приближенное дискретное и по времени, и по пространственной координате решение. В результате будут получены приближенные значения y в узловых точках x_j для дискретных моментов времени t_n :

ренциальные операторы $\frac{\partial}{\partial t}$ и $\frac{\partial}{\partial x}$ заменяются их приближенными конечно-разностными аналогами. Методом конечных разностей будет построено приближенное дискретное и по времени, и по пространственной координате решение. В результате будут получены приближенные значения y в узловых точках x_j для дискретных моментов времени t_n :

$$x_{j+1} = x_j + h, \quad t_{n+1} = t_n + \tau, \text{ где } h - \text{ шаг по координате } x, \tau - \text{ шаг по времени } t.$$



Введем общепринятую индексную форму обозначения: $y_j^n = y(t_n, x_j)$, $y_j^{n+1} = y(t_{n+1}, x_j)$, $y_{j-1}^{n+1} = y(t_{n+1}, x_{j-1})$. Пусть в момент времени $t=t_n$ нам известны значения y_j^n в любой точке канала по оси x .

Используем следующие конечно-разностные соотношения для дифференциальных операторов $\frac{\partial}{\partial t}$ и $\frac{\partial}{\partial x}$:

ренциальных операторов $\frac{\partial}{\partial t}$ и $\frac{\partial}{\partial x}$:

$$\frac{\partial y^{n+1}}{\partial x} \approx \frac{y^{n+1}(x_j) - y^{n+1}(x_{j-1})}{h}; \quad \frac{dy_j}{dt} \approx \frac{y_j^{n+1} - y_j^n}{\tau}.$$

Тогда исходное уравнение примет вид:

$$\frac{y_j^{n+1} - y_j^n}{\tau} + \frac{y_j^{n+1} - y_{j-1}^{n+1}}{h} = 1.$$

Последнее соотношение есть конечно-разностный аналог исходного дифференциального уравнения в частных производных, который имеет первый порядок точности и по координате, и по времени. Первое слагаемое есть конечно-разностный аналог частной производной по времени, а второе слагаемое есть конечно-разностный аналог частной производной по координате.

Преобразуем конечно-разностный аналог уравнения следующим образом:

$$y_j^{n+1} \left(\frac{1}{\tau} + \frac{1}{h} \right) = 1 + \frac{y_j^n}{\tau} + \frac{y_{j-1}^{n+1}}{h}, \quad y_j^{n+1} = \frac{1}{\left(\frac{1}{\tau} + \frac{1}{h} \right)} + y_j^n \frac{\frac{1}{\tau}}{\left(\frac{1}{\tau} + \frac{1}{h} \right)} + y_{j-1}^{n+1} \frac{\frac{1}{h}}{\left(\frac{1}{\tau} + \frac{1}{h} \right)}.$$

Последнее соотношение есть окончательная расчетная формула, последовательно применяя которую для точек по x с индексами $j = 2 - M$, (M – номер последней точки по координате x) по известным значениям y_j^n получим искомые величины y_j^{n+1} . Для $j=1$, т. е. при $x=0$, значение y_j^{n+1} всегда известно, так как задано краевыми условиями. Решение задачи переноса в координатах $y(x)$ для различных моментов времени представлено на рис. 2.14–2.15.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	h	Xi	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10
2	0,02	0	0	0	0	0	0	0	0	0	0	0	0
3	tay	0,02	0	0,014	0,018	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02
4	0,05	0,04	0	0,024	0,034	0,038	0,039	0,04	0,04	0,04	0,04	0,04	0,04
5	A	0,06	0	0,032	0,048	0,055	0,058	0,059	0,06	0,06	0,06	0,06	0,06
6	0,285714	0,08	0	0,037	0,059	0,071	0,076	0,078	0,079	0,08	0,08	0,08	0,08
7	B	0,1	0	0,041	0,068	0,084	0,093	0,097	0,099	0,099	0,1	0,1	0,1
8	0,714286	0,12	0	0,043	0,075	0,096	0,108	0,114	0,117	0,119	0,12	0,12	0,12
9	C	0,14	0	0,045	0,081	0,106	0,122	0,131	0,135	0,138	0,139	0,14	0,14
10	0,014286	0,16	0	0,047	0,085	0,114	0,134	0,146	0,153	0,156	0,158	0,159	0,16
11		0,18	0	0,048	0,089	0,121	0,145	0,16	0,169	0,174	0,177	0,179	0,179
12		0,2	0	0,048	0,092	0,127	0,154	0,172	0,184	0,191	0,195	0,198	0,199
47		0,9	0	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,399	0,448	0,497
48		0,92	0	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,399	0,449	0,497
49		0,94	0	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,449	0,498
50		0,96	0	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,449	0,498
51		0,98	0	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,449	0,499
52		1	0	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,449	0,499
53													

Рис. 2.14. Электронная таблица для решения задачи переноса

$$A = \frac{1/\tau}{(1/\tau + 1/h)}; \quad B = \frac{1/h}{(1/\tau + 1/h)}; \quad C = \frac{1}{(1/\tau + 1/h)}.$$

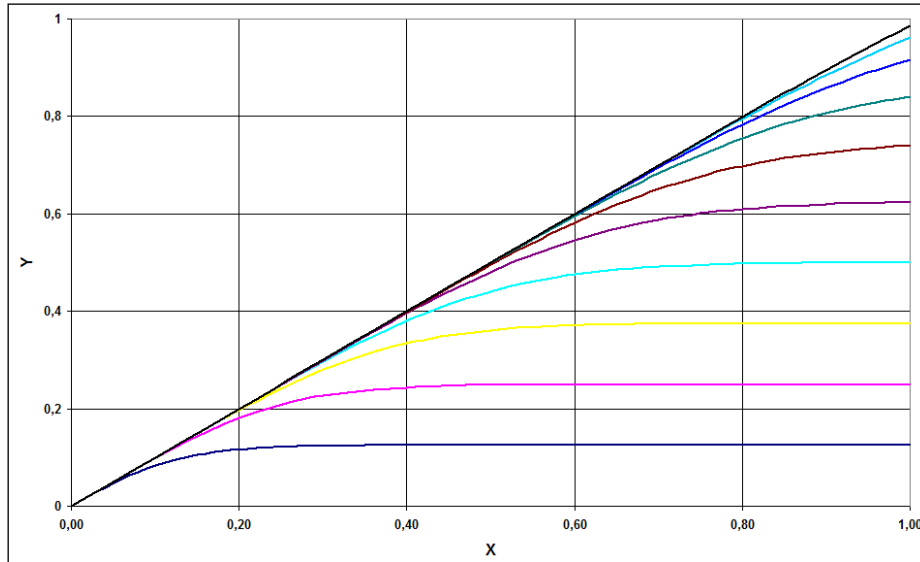


Рис. 2.15. Результат решения задачи переноса в среде электронных таблиц

Рассмотренный метод решения уравнения в частных производных называется методом бегущего счета. Метод, безусловно, устойчив, так как значения τ и h выбираются только исходя из требуемой точности решения.

Процессы теплопроводности и диффузии сводятся к абсолютно идентичным математическим моделям. Построение модели процесса теплопроводности рассмотрим на основе следующей задачи.

Имеется длинный металлический стержень, который в начальный момент времени имеет одинаковую температуру T_0 по всей длине. В тот же самый начальный момент времени температура на правом торце стержня изменяется скачкообразно до значения T_1 и сохраняет это значение в течение всего времени. Левый торец стержня сохраняет постоянную температуру T_0 во все моменты времени. По боковой поверхности стержень имеет возможность взаимодействовать с окружающей средой (нагреваться или охлаждаться). Целью моделирования является исследование **переходного процесса** изменения температуры стержня после скачкообразного изменения температуры правого торца стержня.

Считаем, что длина стержня значительно больше его радиуса $L \gg R$. Материал стержня имеет достаточно высокую теплопроводность, следовательно, изменением температуры в поперечном сечении пренебрегаем (она считается однородной в любом поперечном сечении). На рис. 2.16 представлена расчетная схема объекта исследования:

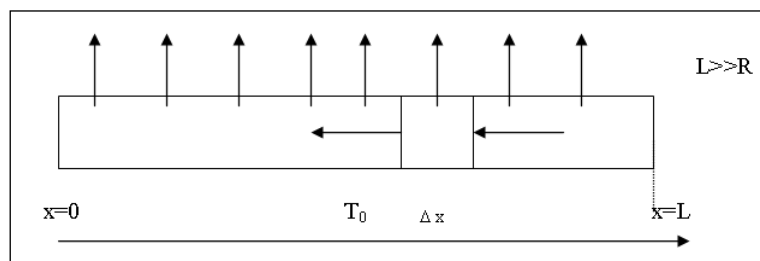


Рис. 2.16. Расчетная схема объекта исследования

Процесс передачи тепла в стержне описывается уравнением:

$$\rho \cdot c \frac{\partial T}{\partial t} = \lambda \frac{\partial^2 T}{\partial x^2} + \frac{\Pi}{F} \alpha (T_0 - T),$$

здесь x – продольная координата, t – время, Π – периметр поперечного сечения стержня, F – площадь поперечного сечения, α – коэффициент теплоотдачи в окружающую среду через боковую поверхность стержня, λ – коэффициент теплопроводности материала стержня, ρ – плотность материала стержня, T_0 – тем-

пература окружающей среды, $T(t,x)$ – температура стержня, функция времени и координаты.

В начальный момент времени $T(t=0, x)=T_0$, а краевые условия имеют вид:

$$T(t,x=0)= T_0, \quad T(t,x=L)= T_1.$$

Уравнение теплопроводности построено на основе закона сохранения энергии и закона Фурье. Закон Фурье связывает величину осевого потока тепла по стержню за счет теплопроводности с градиентом температуры:

$$q = -\lambda \frac{\partial T}{\partial x}.$$

Тепловой поток в окружающую среду через боковую поверхность стержня определяется законом Ньютона:

$$q = \alpha(T_0 - T).$$

Данные законы имеют чисто феноменологический характер. Тепловой поток – это количество тепла, проходящее за единицу времени через сечение площадью в 1м^2 .

Физический смысл членов уравнения теплопроводности состоит в следующем:

1. Скорость изменения запаса тепла в выделенном элементарном объёме.
2. Изменение запаса тепла в элементарном объёме, которое вызвано теплопроводностью.
3. Изменение запаса тепла в элементарном объёме вследствие теплоотдачи в окружающую среду (рис. 2.16).

$$\Delta x \cdot F \cdot \rho \cdot c \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left(\Delta x \cdot F \cdot \lambda \frac{\partial T}{\partial x} \right) + \alpha (T_0 - T) \cdot \Pi \cdot \Delta x,$$

Таким образом, в выделенном элементе протекают следующие процессы: накопление тепла, продольная передача тепла по стержню вследствие теплопроводности и теплоотдача через боковую поверхность в окружающую среду.

Уточним цели моделирования. Представляет интерес выявление нового стационарного температурного режима, который установится по окончании переходного процесса, а также определение закономерности и времени про-

текания переходного процесса. Различают задачи стационарной и нестационарной теплопроводности.

Предварительно приведем уравнения к безразмерному виду, преобразование исходного уравнения теплопроводности дает следующее:

$$\frac{\partial T}{\partial t} = a \frac{\partial^2 T}{\partial x^2} + k(T_0 - T), \quad a = \frac{\lambda}{\rho \cdot c}, \quad k = \frac{\Pi \alpha}{F \cdot \rho \cdot c}.$$

В качестве масштаба по продольной координате выберем длину стержня L . Тогда безразмерная координата определяется соотношением: $\bar{x} = x / L$.

После преобразования уравнение примет вид:

$$\frac{L^2}{a} \frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial \bar{x}^2} + \frac{kL^2}{a} (T_0 - T).$$

В качестве масштаба по времени примем следующую величину $t^* = L^2 / a$. Безразмерное время определим следующим образом: $\bar{t} = t / t^*$. Безразмерную температуру зададим формулой $y = (T - T_0) / (T_1 - T_0)$. С учетом данных соотношений уравнение теплопроводности, начальные и краевые условия преобразуется в следующее:

$$\frac{\partial y}{\partial \bar{t}} = \frac{\partial^2 y}{\partial \bar{x}^2} - \bar{k} \cdot y, \quad y(\bar{t} = 0, \bar{x}) = 0, \quad y(\bar{t}, \bar{x} = 0) = 0, \quad y(\bar{t}, \bar{x} = 1) = 1.$$

Безразмерный параметр \bar{k} является критерием подобия и характеризует соотношение интенсивности процесса теплоотдачи в окружающую среду через боковую поверхность стержня и процесса теплопроводности вдоль стержня.

Для решения данной начально-краевой задачи применим метод конечных разностей. Значение производной по времени приближенно зададим соотношением:

$$\frac{y_j^{n+1} - y_j^n}{\tau} \approx \left. \frac{\partial y}{\partial t} \right|_{x=x_j}.$$

Значение второй производной по координате выразим через известную формулу:

$$\frac{y_{j+1}^{n+1} - 2y_j^{n+1} + y_{j-1}^{n+1}}{h^2} \approx \frac{\partial^2 y}{\partial x^2}.$$

Тогда конечно-разностный аналог уравнения примет вид:

$$\frac{y_j^{n+1} - y_j^n}{\tau} = \frac{y_{j+1}^{n+1} - 2y_j^{n+1} + y_{j-1}^{n+1}}{h^2} - ky_j^{n+1},$$

$$\left(\frac{1}{h^2} y_{j-1}^{n+1} - \left(\frac{1}{\tau} + \frac{2}{h^2} + k \right) y_j^{n+1} + \frac{1}{h^2} y_{j+1}^{n+1} \right) = -\frac{y_j^n}{\tau}.$$

Последнее соотношение справедливо для $j = 2 \div (M - 1)$, т.е. для внутренних узловых точек по x . Для краевых точек значения y_j^{n+1} заданы краевыми условиями: $j=1$, $y_j^{n+1} = 0$, а для $j=M$, $y_j^{n+1} = 1$.

Таким образом, применение метода конечных разностей позволяет преобразовать дифференциальное уравнение в систему линейных алгебраических уравнений относительно y_j^{n+1} , которая даст искомое решение. Подобная специфическая система линейных уравнений с трехдиагональной ленточной матрицей решается с помощью специальной разновидности метода исключения Гаусса, который называется методом прогонки. Полученная выше разностная схема безусловно устойчива и имеет первый порядок точности по времени и второй порядок точности по координате.

В общем случае полученная система линейных алгебраических уравнений может быть представлена в следующем каноническом виде:

$$\begin{cases} A_j y_{j-1} - C_j y_j + B_j y_{j+1} = -F_j \\ y_1 = \chi_1 y_2 + \upsilon_1 \\ y_M = \chi_2 y_{M-1} + \upsilon_2 \end{cases}$$

В рассматриваемом случае коэффициенты A , B , C и F имеют конкретные значения:

$$A_j = \frac{1}{h^2}, \quad B_j = \frac{1}{h^2}, \quad C_j = \frac{1}{\tau} + \frac{2}{h^2} + k, \quad F_j = \frac{y_j^n}{\tau};$$

$$\chi_1 = 0, \quad \nu_1 = 1, \quad \chi_2 = 0, \quad \nu_2 = 0.$$

Главная идея метода прогонки состоит в том, что решение системы уравнений ищется в виде рекуррентного соотношения: $y_j = \alpha_j y_{j+1} + \beta_j$. Коэффициенты α_j и β_j называются прогоночными. Положим $j=1$, тогда $y_1 = \alpha_1 y_2 + \beta_1$, в то же время $y_1 = \chi_1 y_2 + \nu_1$, из чего следует, что $\alpha_1 = \chi_1$, $\beta_1 = \nu_1$. Расчет остальных прогоночных коэффициентов выполняется по следующим рекуррентным формулам:

$$\alpha_j = \frac{B_j}{C_j - A_j \alpha_{j-1}}; \quad \beta_j = \frac{F_j + A_j \beta_{j-1}}{C_j - A_j \alpha_{j-1}}.$$

После определения прогоночных коэффициентов, решение определяется по соотношению:

$$y_j = \alpha_j y_{j+1} + \beta_j, \quad \text{где } j = (M-1) \div 2.$$

Таким образом, алгоритм решения задачи содержит следующие этапы:

1. Определить значения коэффициентов A_j, B_j, C_j, F_j .
2. Исходя из краевых условий, определить значения $\chi_1, \nu_1, \chi_2, \nu_2$.
3. Задать значения прогоночных коэффициентов α_1 и β_1 .
4. Рассчитать α_j и β_j , причем j изменяется от $j=2$ до $j=(M-1)$.
5. Рассчитать y_j^{n+1} , в этом случае j изменяется от $j=(M-1)$ до $j=2$.

Результат численного решения задачи теплопроводности в координатах $y(x)$ для различных моментов времени и значения параметра $\bar{k}^{\text{в}} = 4$ представлен на рис. 2.17–2.18.

	A	B	C	D	E	F	G	H	I
1	tau	Xi	Y0	alfa	beta	Y1	alfa	beta	Y2
2	0,01	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
3	h	0,0200	0,0000	0,4898	0,0000	0,0000	0,4898	0,0000	0,0001
4	0,02	0,0400	0,0000	0,6444	0,0000	0,0000	0,6444	0,0000	0,0002
5	k	0,0600	0,0000	0,7157	0,0000	0,0000	0,7157	0,0000	0,0003
6	4,00	0,0800	0,0000	0,7542	0,0000	0,0001	0,7542	0,0000	0,0004
7	A	0,1000	0,0000	0,7768	0,0000	0,0001	0,7768	0,0000	0,0005
8	2500,00	0,1200	0,0000	0,7906	0,0000	0,0001	0,7906	0,0000	0,0006
9	B	0,1400	0,0000	0,7994	0,0000	0,0001	0,7994	0,0000	0,0008
10	2500,00	0,1600	0,0000	0,8050	0,0000	0,0002	0,8050	0,0000	0,0009
11	C	0,1800	0,0000	0,8087	0,0000	0,0002	0,8087	0,0000	0,0011
12	5104,00	0,2000	0,0000	0,8111	0,0000	0,0003	0,8111	0,0000	0,0014
42		0,8000	0,0000	0,8158	0,0000	0,1305	0,8158	0,0127	0,2579
43		0,8200	0,0000	0,8158	0,0000	0,1600	0,8158	0,0156	0,3005
44		0,8400	0,0000	0,8158	0,0000	0,1962	0,8158	0,0191	0,3492
45		0,8600	0,0000	0,8158	0,0000	0,2404	0,8158	0,0235	0,4046
46		0,8800	0,0000	0,8158	0,0000	0,2947	0,8158	0,0288	0,4673
47		0,9000	0,0000	0,8158	0,0000	0,3613	0,8158	0,0352	0,5375
48		0,9200	0,0000	0,8158	0,0000	0,4429	0,8158	0,0432	0,6157
49		0,9400	0,0000	0,8158	0,0000	0,5429	0,8158	0,0530	0,7018
50		0,9600	0,0000	0,8158	0,0000	0,6655	0,8158	0,0649	0,7953
51		0,9800	0,0000	0,8158	0,0000	0,8158	0,8158	0,0796	0,8954
52		1,0000	1,0000			1,0000			1,0000

Рис. 2.17. Электронная таблица решения задачи теплопроводности

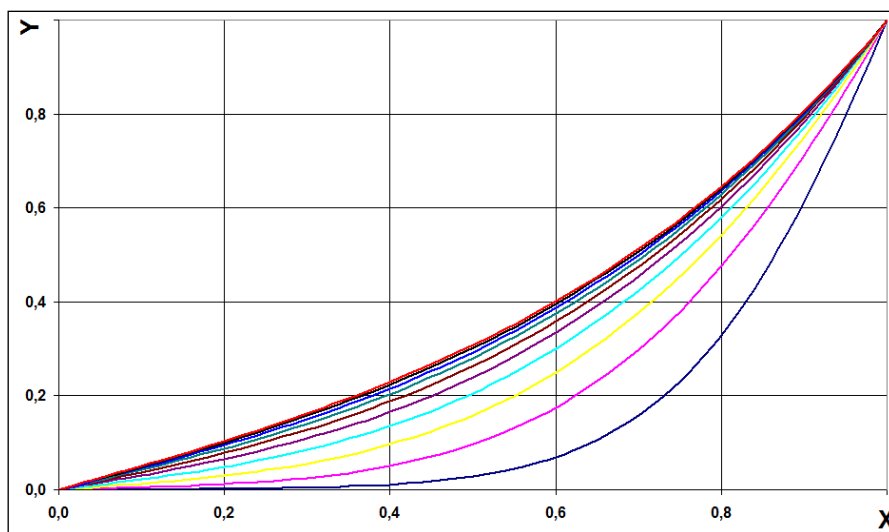


Рис. 2.18. Результат решения задачи теплопроводности в координатах для различных моментов времени

Выше мы рассмотрели решение задачи теплопроводности, когда краевые условия заданы в виде значений искомой функции. Такие краевые условия называются краевыми условиями первого рода. Но рассмотренные методы позволяют решать задачу для более сложных краевых условий.

2.5. КОМПЬЮТЕРНЫЙ ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ

Компьютерный вычислительный эксперимент проводится с компьютерной моделью с целью получения информации о свойствах объекта моделирования. В принципе компьютерный эксперимент и эксперимент с натурным объектом имеют много общего в плане методологии проведения. Однако компьютерный вычислительный эксперимент, несомненно, имеет ряд особенностей, которые состоят в следующем:

1. Простота проведения и повторения эксперимента с моделью.
2. Возможность полного воспроизведения условий эксперимента.
3. Возможность управления экспериментом.
4. Варьирование условий проведения эксперимента.
5. Наглядное представление результатов экспериментирования.

Во многих случаях компьютерное математическое моделирование применяется для изучения уже произошедших событий. Однако существует значительное количество областей практической деятельности, в которых невозможно полагаться на непосредственное наблюдение и экспериментирование с реальными системами, поскольку последствия таких действий могут оказаться необратимыми. В технике актуальной проблемой является изучение свойств еще несуществующих объектов в процессе проектирования. Вычислительный эксперимент позволяет исследовать события или явления, которые могут носить исключительно гипотетический характер (например, ядерная зима). Изучать такие объекты и процессы, при этом получая информацию о параметрах процессов, можно только с помощью компьютерного эксперимента с математическими моделями. Кроме того, компьютерный эксперимент позволяет выяснить механизмы развития процессов путем численного анализа влияния различных факторов на свойства системы. Это практически невозможно в натурном эксперименте.

Таким образом, в современной науке и технике появляется все больше областей, задачи в которых необходимо решать исключительно на основе компьютерного вычислительного эксперимента.

Энергетика. Прогнозирование и диагностика работы атомных реакторов, энергетических установок и энергетических систем на основе детального математического моделирования происходящих в них процессов. Моделирование аварийных ситуаций в энергосистемах, выбор оптимальных режимов их работы.

Аэрокосмическая техника. Расчет траекторий летательных аппаратов, решение задач аэродинамики, автоматизированное проектирование.

Технологические процессы. Моделирование технологических процессов, которые опираются на тонкие физико-химические эффекты.

Физика. Это классическая область применения математического моделирования. В 1982 г. Нобелевская премия по физике присуждена К. Вильсону, предложившему ряд фундаментальных моделей в теории элементарных частиц, которые исследовались численно.

В целом технологический цикл вычислительного эксперимента содержит следующие этапы:

1. Для исследуемого объекта или процесса строится математическая модель, формулируются допущения и границы применимости модели.

2. Математическая модель на основе численных методов преобразуется в такую форму, которая позволит получить информацию о свойствах объекта в ходе вычислительного эксперимента. В итоге разрабатывается вычислительный алгоритм.

3. Для реализации вычислительного алгоритма создается компьютерная программа. Компьютерная программа позволяет проводить компьютерный вычислительный эксперимент с целью получения необходимой информации о свойствах объекта моделирования.

На определенном этапе проведения вычислительных экспериментов наступает фаза прогноза: средствами математического моделирования предсказывается поведение объекта в условиях, где опыты не проводились или они

вообще невозможны. Для этого уровня справедлива точка зрения Р. Хемминга: **«Цель расчетов – понимание, а не числа».**

В итоге схема компьютерного эксперимента может быть представлена в виде следующей цепочки: Объект – математическая модель – вычислительный алгоритм – программа – компьютерный эксперимент. Таким образом, основой компьютерного вычислительного эксперимента является математическая модель, теоретической базой – прикладная математика, а технической базой – электронная вычислительная техника (компьютер).

Применение инструментальных систем моделирования по содержанию не меняет указанных действий. Инструментальные системы лишь упрощают процесс создания и численной реализации математической модели. План вычислительного эксперимента и анализ его результатов всегда остается прерогативой исследователя.

2.6. ФОРМАЛИЗАЦИЯ И МОДЕЛИРОВАНИЕ

Во многих случаях при построении модели необходимо описание объекта, которое не допускает неоднозначного толкования. Такое описание требует определенной формы представления и выполняется по строгим правилам. Подобное описание называется **формализованным**, оно необходимо, если, например, требуется исследование свойств модели **математическими методами**.

Термин «формализация» означает сведение содержания к определенной форме, т.е. некоторые знания приводятся к достаточно строгой и точной форме представления. Потребность в формализации возникла вследствие необходимости описания профессиональных знаний и введения четких профессиональных понятий для массового обучения. **«Определяйте значения слов, и вы избавите мир от половины его заблуждений» – Рене Декарт.**

По мнению академика Н.Н. Моисеева, «на первой ступени познания всегда лежит опыт человеческой деятельности: наблюдение, эксперимент, изуче-

ние явления, накопление фактов. Вторая ступень – это абстрактное описание изучаемого объекта, обобщение, построение теории. При этом описание объекта выполняется на одном из языков, которые отличаются разным уровнем формализации». Наиболее ярко это видно на примере конкретных наук. Разделение наук объективно потребовало специализированных языков, более точных, чем естественный язык. Модели специальных наук более точны и конкретны, они содержат больше информации.

Преобразование смысла в информацию есть, по сути, первый этап формализации, который осуществляется мышлением человека. Результат этой формализации можно видеть опосредованно, например, в форме устной речи. Древняя мудрость гласит: «Кто ясно мыслит, тот ясно излагает». Формализация – отображение результатов мышления в точных понятиях и сущностях, которое противопоставляется интуитивному мышлению.

Таким образом, возможность отчуждения профессиональных знаний от их носителей определяется возможностью формализации этих знаний. Области знаний, которые оказались более формализуемыми, получили название **точных наук** (science), например: физика, химия и др. В этих науках широко применяются математические методы анализа. Остальные области знания образовали **гуманитарные науки** (art).

Всякое представление об окружающем мире связано с построением определенной модели. Описание объектов и отношений между ними средствами языка приводит к тому, что сами объекты и их свойства замещаются определенными **именами**.

С помощью имени можно однозначно определить или опознать объект. Действительно, если человек встречает новый для него объект, то задается вопросом: «Что это такое?», т.е. «Каково **имя** этого объекта?».

Принципиально новый объект именуется и в дальнейшем имя этого объекта является его заменителем в любом описании. Имена объектов, их свойств и отношений затем используются при построении моделей. Это основа формализации (рис. 2.19).

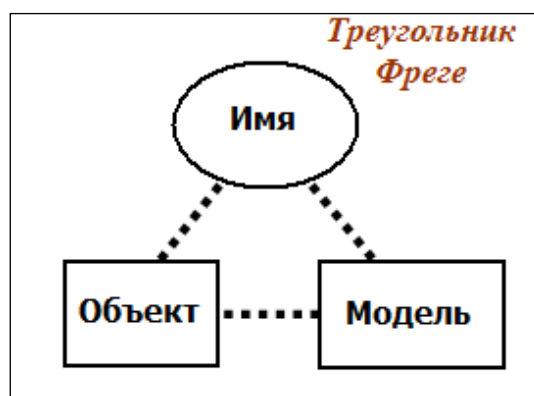


Рис. 2.19. Треугольник Фреге

Формализация выполняется на основе некоторой формальной системы. Формальная система – это система, определяющая множество объектов путем описания исходных объектов и правил построения новых объектов. Объекты формальной системы – это символическое или графическое представления реальных объектов, состояний, структур и т.п. Формальная система различает объекты по их ясно различимым описательным признакам.

Для формальных систем существует **проблема выводимости**. Суть проблемы состоит в следующем, можно ли с помощью конкретной формальной системы построить описание данного объекта.

Итак, формализованное описание объекта моделирования позволяет построить его строгое и однозначное описание (модель). Напомним, что любая теоретическая проблема поставлена точно, если она формализована, т.п. представлена в рамках определенной формальной системы. В этом случае для решения проблемы можно использовать **математические методы**.

Результаты анализа формального описания (формальной модели) с целью их практического использования необходимо перенести на реальные объекты или **интерпретировать**. Оказалось, что формальные модели, разнообразные по своей первоначальной сущности, могут быть тождественными по форме. По этой причине математические модели разделены математиками на соответствующие классы, свойства которых исследуются, как формальных математических объектов.

Интерпретация результатов анализа формальной модели объекта (обратное преобразование по сравнению с формализацией) не является однозначной процедурой. Она может быть выполнена по-разному в зависимости от

конкретной предметной области. Результаты математического анализа формальных моделей интерпретируются уже специалистами конкретных прикладных наук. Таким образом, одна и та же формальная модель может иметь разные интерпретации. Такое явление называется аналогией.

Дополнительные теоретические материалы по математическому моделированию объектов, процессов и систем можно найти в монографиях, указанных в библиографическом списке [1; 2; 7–8; 16; 28; 32–33; 37–38; 41].

2.7. КОНТРОЛЬНЫЕ ВОПРОСЫ К ГЛАВЕ 2

1. Что такое математическая модель?
2. Каковы особенности математических моделей?
3. В чем состоит актуальность математического моделирования?
4. Каковы особенности построения математических моделей?
5. Что такое параметр?
6. Какова роль измерений в математическом моделировании?
7. Какие характеристики численных методов актуальны для моделирования?
8. Какие существуют способы построения математических моделей?
9. В чём преимущество безразмерных моделей?
10. В чём суть понятия «подобие»?
11. В чём суть понятия «аналогия»?
12. В чем суть формализации?
13. Что такое вычислительный эксперимент?
14. На какие классы можно разделить методы исследования математических моделей?
15. Какие этапы включает в себя технологический цикл вычислительного эксперимента?
16. Приведите примеры задач, которые решаются исключительно на основе вычислительного эксперимента.
17. Чем принципиально отличаются модели с сосредоточенными параметрами от математических моделей с распределенными параметрами?

ГЛАВА 3. МОДЕЛИРОВАНИЕ СИСТЕМ

3.1. МОДЕЛИРОВАНИЕ СЛОЖНЫХ СИСТЕМ

Одной из проблем современных технических, социальных и естественных наук является разработка и внедрение в практику методов моделирования **сложных систем**. Как отмечал еще академик Н.Н. Моисеев «сложность изучаемых и проектируемых систем приводит к необходимости создания специальной, качественно новой техники исследования».

Сложными системами обычно считают крупные технологические, производственные, энергетические системы; коммуникационные комплексы; системы управления; социальные, экономические, экологические системы и т.п. Сложные системы могут быть искусственного или естественного происхождения.

Сложные системы характеризуются **большим количеством элементов и связей**, разнообразием взаимодействий между системой и окружающей средой, наличием **случайных факторов** внутренней и внешней природы, возможностью случайного изменения структуры. Обычно выделяют **структурную и поведенческую** сложность моделируемых объектов.

Для систем сложного поведения характерно несколько качественно различных, последовательно сменяющих друг друга во времени, режимов функционирования. Подобная сложная динамическая система в каждый конкретный момент времени ведет себя как некоторая относительно простая динамическая система. Однако при определенных условиях режим функционирования системы скачкообразно меняется. Например, бассейн с двумя трубами обладает сложным поведением, поскольку его поведение при переполнении качественно отличается от его поведения при нормальном уровне воды или при опорожнении.

Дать полный перечень всех функций моделей сложных систем затруднительно. Рассмотрим наиболее распространенные.

1. Модели сложных систем помогают упорядочить представления о свойствах этих систем. Такие модели используются, например, в социальных науках. А в технике такие модели служат в качестве средства для создания новых более совершенных систем.

2. Модели сложных систем часто применяются как средство обучения лиц, которые должны уметь справляться со всевозможными случайностями до возникновения критической ситуации (модели космических кораблей, тренажеры для обучения водителей, операторов АЭС и др.).

3. Одним из важных применений моделей сложных систем является прогнозирование поведения моделируемых объектов и диагностика их состояния.

4. Модели сложных систем позволяют производить контролируемые эксперименты в ситуациях, где экспериментирование на реальных объектах экономически нецелесообразно, опасно или практически невозможно.

Естественно, что при исследовании сложной системы, содержащей большое количество элементов, невозможно и не имеет смысла детально моделировать процессы в каждом элементе. Если строить детальное описание элементов, то модель сложной системы вряд ли удастся исследовать с необходимой полнотой даже средствами современной вычислительной техники. С общесистемной точки зрения представляют интерес только те свойства элементов, которые непосредственно влияют на **свойства системы в целом**.

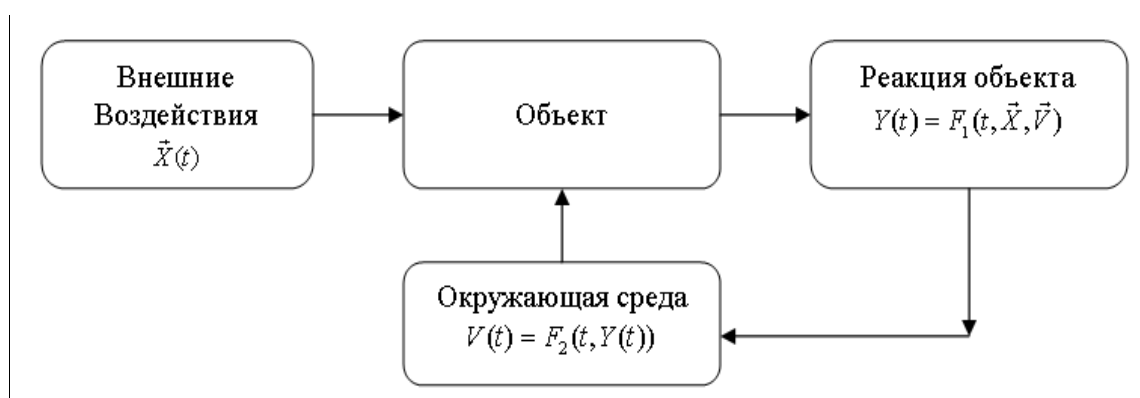


Рис. 3.1. Структура модели элемента типа «черный ящик»

Таким образом, в силу сложности системы модели элементов должны быть просты для реализации средствами вычислительной техники. Поэтому

модели сложных систем строятся на основе моделей элементов типа «**черный ящик**». Такие модели отражают только зависимость между реакцией элемента, внешними воздействиями и влиянием окружающей среды, но не отражают процессы внутри элемента системы (рис. 3.1). Название модели подчеркивает полное отсутствие сведений о внутреннем содержании «черного ящика». Вектор внешних воздействий $\overset{1}{X}(t)$ в данном случае включает в себя и действие на объект других элементов системы. Такая модель (зависимости F_1 и F_2 , рис. 3.1) может быть построена на основании данных наблюдений и экспериментов исключительно формальными методами, например, на основе регрессионного анализа.

Преимущества таких моделей состоит не только в простоте их реализации, но и в возможности построения моделей элементов, даже в том случае, если законы протекающих в них процессов не исследованы теоретически, но поведение элементов может быть изучено в ходе натурных испытаний. Таким образом, модели элементов типа «черный ящик» позволяют создавать модели сложных систем, с которыми можно производить компьютерный эксперимент.

Для сложных систем определяющую роль играет моделирование **структуры системы**, взаимодействия между ее частями и взаимодействия с окружающей средой. Таким образом, при моделировании сложных систем возникает задача анализа функционирования системы в целом. При этом «физическая» сущность процессов, протекающих в системе, отодвигается на второй план. Малосущественно, идет ли речь о машиностроительном предприятии, химическом реакторе, телефонной сети, информационной системе или рынке. Поэтому изучение комплексных проблем сложных систем выделилось в самостоятельное научно-техническое направление, названное **системотехникой**.

Любая модель является абстрактным, формально построенным объектом. Ясно, что модель функционирования сложной системы в состоянии охватить только основные, характерные закономерности, естественно, оставляя в стороне несущественные второстепенные факторы. Построение модели систе-

мы начинается с изучения ее структуры и протекающих в ней процессов. В результате этого появляется **содержательное описание** системы.

Содержательное описание – это первая попытка четко описать закономерности, характерные для исследуемой системы и поставить задачу моделирования. На этапе содержательного описания выявляются **подсистемы**, которые входят в систему; определяются **структурные взаимосвязи**; определяется то, что будет считаться **окружающей средой**.

Построение модели сложной системы начинается с разработки **структурно-функциональной модели** (схемы). В этой схеме отображаются подсистемы и элементы системы, связи между ними и т.д.

Модель сложной системы строится по **иерархическому принципу** на основе моделей подсистем, входящих в состав системы, моделей элементов системы; на основе моделирования связей между подсистемами, между элементами, моделирования взаимодействия с окружающей средой (рис. 3.2).

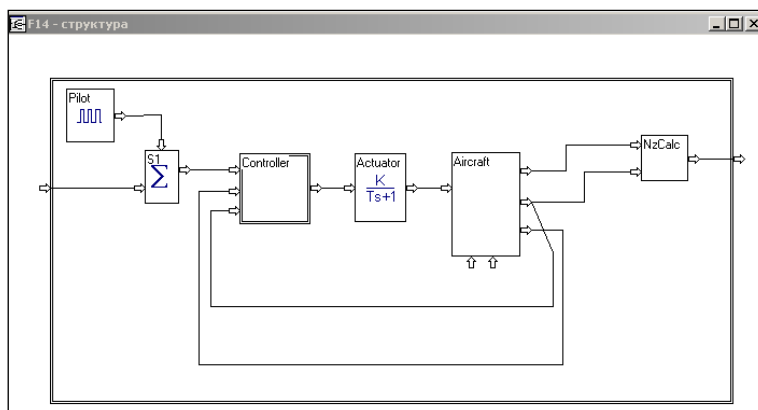


Рис. 3.2. Модель системы управления самолетом, каждый блок – подсистема

Первоначально в схеме отображаются только сами элементы, подсистемы и связи между ними. Далее производится «наполнение» элементов и подсистем законами их функционирования, задаются формальные соотношения, характеризующие взаимодействия элементов в системе и т.п.

Среди задач исследования сложных систем можно выделить задачи анализа, связанные с изучением свойств и поведения системы в зависимости от ее структуры и значений параметров; задачи синтеза, сводящиеся к выбору структуры системы и значений параметров исходя из желаемых свойств

системы в целом. Одной из задач является исследование способов управления сложной системой. Эта задача может быть решена исключительно средствами моделирования.

Итак, характерными признаками сложной системы является наличие большого количества взаимодействующих элементов; сложность функций, выполняемых системой; возможность разбиения системы на подсистемы; наличие управления системой, наличие взаимодействия с окружающей средой; функционирование в условиях воздействия случайных факторов (рис. 3.3).

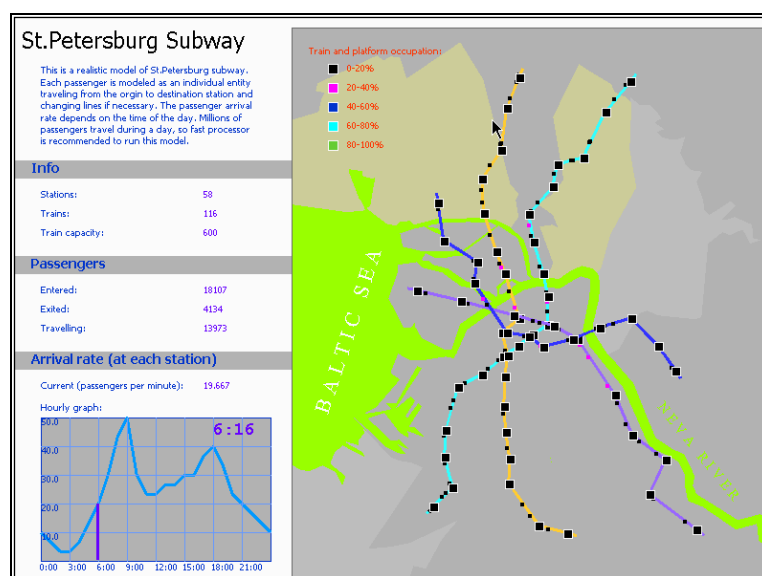


Рис. 3.3. Пример сложной системы – метрополитен Санкт-Петербурга

Исследование сложных динамических систем может быть выполнено на основе современных пакетов **визуального моделирования** (рис. 3.4). Такие пакеты на сегодняшний день являются одним из основных инструментов быстрого построения и исследования моделей в инженерной и научной практике.

Пакеты визуального моделирования позволяют вводить описание моделируемой системы в естественной для прикладных наук форме. Имеется возможность в буквальном смысле рисовать функциональную схему, размещать на ней блоки и соединять их связями. Примеры инструментальных систем визуального моделирования были нами рассмотрены ранее.

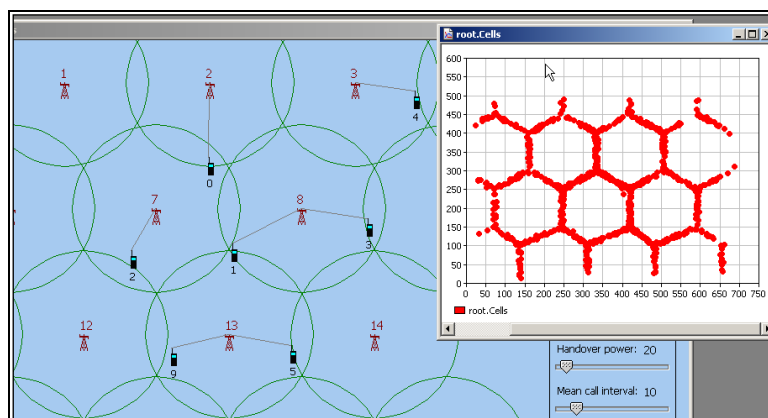


Рис. 3.4. AnyLogic – модель сотовой телекоммуникационной системы

Современные универсальные пакеты визуального моделирования позволяют строить из блоков иерархические функциональные схемы, предоставляют пользователю библиотеки численных методов, средства визуализации поведения, поддерживают технологию объектно-ориентированного моделирования.

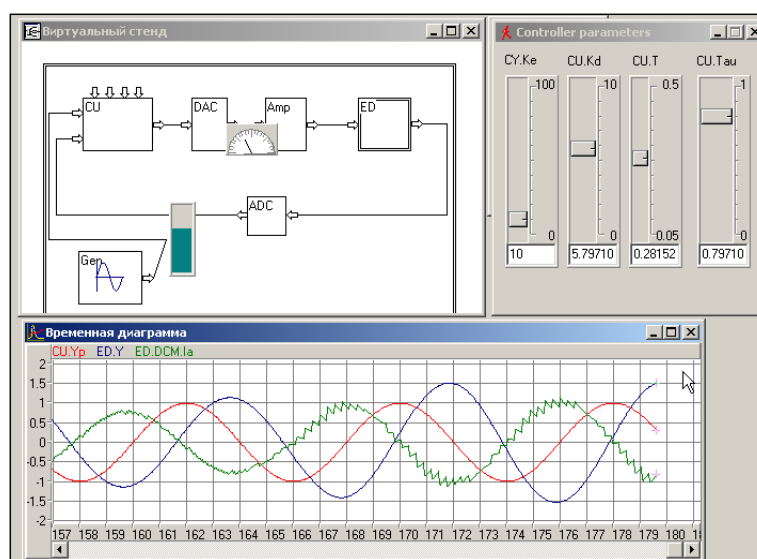


Рис. 3.5. Виртуальный стенд моделирования в среде пакета MVS

Пакеты визуального моделирования позволяют, не заботиться о программной реализации модели, они предоставляют компьютерную среду, в которой можно создавать виртуальные модели систем и проводить эксперименты с этими моделями. Графическая среда позволяет создать «испытательный стенд», где вместо реальных измерительных приборов пользователь будет иметь дело с их образами на экране монитора (рис. 3.5).

3.2. СТРУКТУРНЫЕ МОДЕЛИ

Ранее мы определили систему как совокупность взаимодействующих между собой элементов. Любая система имеет **структуру** – это одно из ее основных свойств. В структуре отображается состав элементов системы и связи между ними. Структура – это внутренняя форма организации системы, определяющая способ взаимодействия составляющих ее элементов. Она придает целостность системе и обуславливает возникновение новых качеств. Так как структура – одно из важнейших свойств системы, то ее моделирование, представляет особый интерес.

Полезность моделирования структуры системы очевидна. Такие модели могут помочь упорядочить нечеткие представления о системе. Например, представление совокупности работ в виде структурной модели позволяет продумать последовательность проведения работ для обеспечения выполнения проекта в заданные сроки. Такая модель помогает выявить временные ограничения, требуемые ресурсы, имеющиеся резервы и т.п. Представление структуры учебного курса позволяет определить его наиболее сложные темы и разработать последовательность изучения его разделов.

При моделировании структуры главным свойством системы является наличие или отсутствие **связей** между элементами, пространственное же размещение элементов для структурных моделей не имеет значения.

Примеры структурных моделей следующие: схема метрополитена, схема энергоснабжения предприятия, электрическая схема, блок-схема алгоритма и т.п. Положение элементов на подобных схемах выбирается из соображений наглядности и удобства анализа, оно никак не связано с их реальным пространственным размещением в отличие, например, от плана местности или географических карт, т.е. геометрических моделей.

Например, схема метрополитена (рис. 3.6) содержит информацию о существующих маршрутах и станциях пересадки пассажиров.



Рис. 3.6. Схема московского метрополитена

Естественно, что в реальности кольцевой маршрут это не окружность, радиальные маршруты по конфигурации далеки от прямых линий. По такой схеме невозможно определить расстояние между станциями и время движения по маршруту.

Все структурные модели имеют нечто общее, что позволило рассматривать их как особый объект. Для этого необходимо отвлечься от содержательной стороны, оставив в схеме только общее: **наличие элементов и связей между ними**. Такая схема называется **графом**. Теория графов родилась непосредственно в ходе решения прикладных задач.

Это работы Эйлера (1737 г.; задача о кёнигсбергских мостах), работы Кирхгофа в области анализа электрических цепей, работы Кэли в области органической химии. Теория графов является разделом прикладной математики.

Граф состоит из элементов произвольной природы, называемых **вершинами**, и связей между ними, называемых **ребрами** или **дугами**. Для отображе-

ния направленности связей, ребра в ряде случаев изображают со стрелками. Если направление обозначено, то граф называется ориентированным, в противном случае называется неориентированным. Пара вершин может быть соединена любым количеством ребер. Вершина может быть соединена сама с собой, в этом случае ребро называется **петлей**.

Если указаны количественные характеристики вершин или ребер, то граф называется **взвешенным**. Если нет ограничений на пересечение ребер, то можно утверждать, что графы могут отображать структуры любой сложности. Некоторые виды структур имеют особое значение для практики, они получили специальные названия: линейные, древовидные (иерархические), сетевые, матричные. Особое место занимают структуры с обратными связями, которые соответствуют кольцевым путям в ориентированных графах.

Построение структурной модели в виде графа позволяет формализовать описание структуры системы. Если представить граф в виде матрицы инцидентий, то структура системы может быть подвергнута анализу средствами вычислительной техники.

Рассмотрим несколько примеров применения структурных моделей в практической деятельности. **Структурные модели** достаточно широко используются в сфере управления в виде моделей **сетевого планирования**. Подобные модели отражают взаимосвязи различных работ в их технологической последовательности при реализации некоторого проекта. Анализ сетевой модели позволяет установить ту последовательность работ, которая определяет сроки выполнения всего проекта (**критический путь**).

Основными понятиями сетевой модели планирования и управления являются событие, работа, путь. Работа характеризует действие, требующее использования ресурсов. Графически работа отображается стрелкой, которая соединяет два события. Каждая работа имеет определенную продолжительность. В последнее время структурные модели получили новое значение при решении логистических задач.

Событиями называются результаты выполнения одной или нескольких работ. События в сетевой модели – это вершины графа (рис. 3.7). Естественно,

что всегда имеется начальное событие (начало работы по реализации проекта) и конечное событие (полное завершение проекта). Работы, выходящие из некоторого события, не могут начинаться, пока не будут завершены все операции, входящие в это событие. Путь – это цепочка работ, соединяющих начальную и конечную вершины. Числа у стрелок означают длительность работы.

Главное значение при планировании и управлении реализацией проекта играет **критический путь**, который имеет максимальную длину.

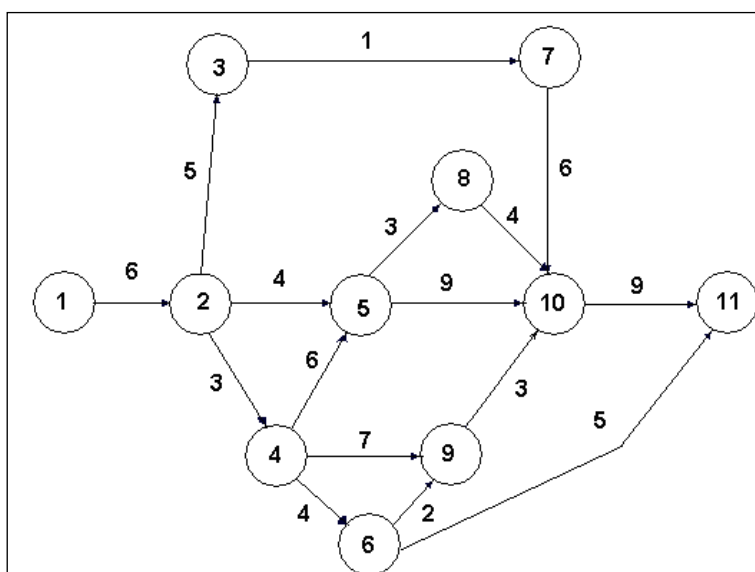


Рис. 3.7. Структурная модель сетевого планирования

Несвоевременное выполнение работ, составляющих критический путь, ведет к срыву выполнения всего проекта. Сетевая модель планирования и управления позволяет выявить резервы времени для работ, которые не принадлежат критическому пути при неизменных сроках реализации проекта в целом.

Таким образом, сетевая модель планирования и управления позволяет выявить главные работы и оптимизировать сроки реализации всего проекта в целом. Пример сетевой модели планирования представлен на рис. 3.7. Критический путь 1–2–4–5–10–11, длительностью в 33 единицы времени.

Особую роль структурные модели играют при проектировании новых технических объектов.

Построение **структурно-функциональной** модели технической системы позволяет создать новые технические решения на уровне изобретений. Дей-

ствительно, в каждой технической системе любой элемент выполняет определенные функции и функционально связан с другими элементами. Любой элемент технической системы реализует свои функции на основе каких-либо физических, химических либо других эффектов. Выяснение функциональных связей между элементами системы на основе структурной модели позволяет получить четкое представление об ее устройстве.

Структурно-функциональную модель можно представить в виде графа, отражающего **взаимодействие элементов** при выполнении своих функций. При этом вершины такого графа – **элементы системы**, а ребра – **функции**, выполняемые элементами системы. Ребра выходят из вершин-элементов, чьи функции они описывают, и заканчиваются в вершинах-элементах, чью работу они обеспечивают или в вершинах-объектах окружающей среды, взаимодействующих с рассматриваемым элементом. Из каждой вершины выходит столько ребер, сколько функций элемент выполняет.

Имея функционально-структурную модель системы, можно выбрать различные **физические эффекты** для реализации функций элементов.

Подобный подход позволяет, на базе вычислительной техники, автоматизировать процесс поиска новых технических решений. Пример функционально-структурной модели одной простой технической системы представлен на рис. 3.8.

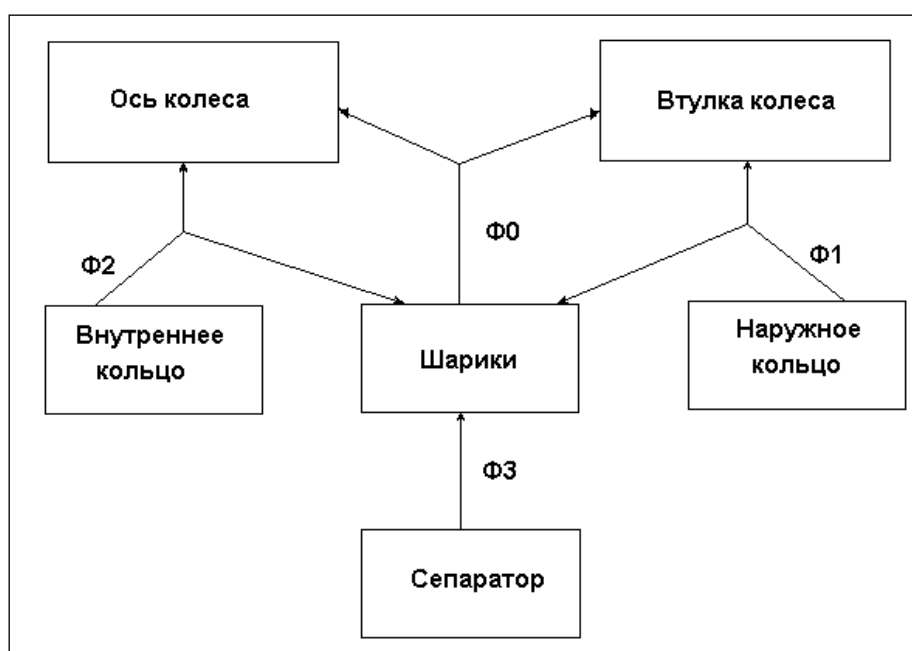


Рис. 3.8. Структурно-функциональная модель подшипника

Функции элементов:

- Ф0 – снижение трения вращения втулки вокруг оси;
- Ф1 – обеспечение качения втулки по шарикам;
- Ф2 – обеспечение качения шариков по оси;
- Ф3 – обеспечение равного удаления между шариками.

Структурно-функциональная модель, представленная на рис. 3.8, позволяет путем вариации физических эффектов получить любой тип подшипника, который известен на сегодня.

Одно лишь построение структурной модели позволяет выявить определенные **качественные свойства** системы. Примером могут служить **когнитивные модели**, которые являются отображениями качественных представлений или рассуждений человека. Такие модели отражают причинно-следственные связи между различными факторами сложной системы или процесса. Они могут быть представлены в виде структурных схем.

Построение когнитивной модели – это необходимый этап любого мыслительного процесса, предшествующий постановке проблемы или формулированию задачи. Психологи считают, что подобного рода модели создаются человеком на уровне подсознания и всегда предшествуют сознательному анализу любой проблемы. С их помощью отражается путь размышлений о возможных причинах данного следствия. Это позволяет человеку делать **причинно-следственные выводы** на основе ограниченной информации (по крайней мере, строить определенные гипотезы).

Например, при принятии решений в различных ситуациях у эксперта возникает модель проблемной области, на основе которой он пытается объяснить происходящие в реальности процессы. При этом объективные закономерности реального мира представляются субъективными экспертными оценками. В результате образ наблюдаемой ситуации отражает не только законы и закономерности ситуации, но и мировоззрение субъекта.

Построение когнитивной модели – объективно необходимый этап моделирования. Действительно, наблюдения за объектом позволяют сформировать

некоторый качественный образ объекта – когнитивную модель. Дальнейшие этапы моделирования связаны с построением более точных моделей с привлечением теорий, законов, эмпирических зависимостей, а главное, с использованием строгого понятийного аппарата конкретной науки.

Цель построения когнитивной модели состоит в формировании и уточнении гипотез о функционировании исследуемого объекта, который рассматривается как сложная система, состоящая из отдельных элементов и подсистем. Для анализа поведения сложной системы строится **схема** (граф) причинно-следственных связей. Элементы схемы отображаются в виде вершин, соединенными ориентированными дугами, направления отражают причинно-следственные связи.

Если фактор А влияет на фактор В, то на схеме это отображается в виде связи: **A→B**, где А – причина, В – следствие. Причем количественная мера взаимодействия факторов может быть неизвестна. Тем не менее, во многих случаях удается получить содержательные выводы, пользуясь только качественными оценками.

В рамках когнитивной схемы причинно-следственные связи делятся на положительные и отрицательные. Связь **A+ → B** называется положительной, если увеличение фактора А приводит к росту фактора В, а уменьшение А ведет к снижению В. Отрицательная связь обозначается **A- → B**, т.е. увеличение А ведет к уменьшению В, а уменьшение А ведет к росту В. Подобные схемы широко используются для анализа сложных взаимодействий в социологии, экономике, психологии, а также при анализе чрезвычайных ситуаций.

Когнитивная модель, в первую очередь, позволяет выявить **обратные связи**. Различают положительные и отрицательные обратные связи. Положительная обратная связь интенсифицирует исходный процесс, в то время как отрицательная обратная связь направлена на торможение процесса, поддерживая его в определенных пределах, т.е. является стабилизирующим фактором. Если схема содержит контуры с положительной обратной связью, то это указывает на возможность неустойчивости системы.

В качестве примера рассмотрим когнитивную модель потребления электроэнергии некоторого региона (рис. 3.9).

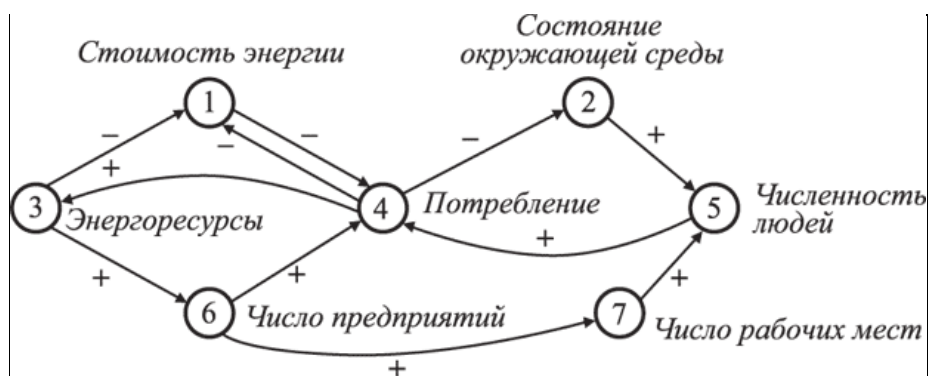


Рис. 3.9. Когнитивная модель потребления электроэнергии

Данная схема отражает взаимодействие различных социально-экономических факторов с развитием энергетического потенциала данного региона. Улучшение окружающей среды ведет к росту населения, а ухудшение окружающей среды ведет к оттоку населения. Увеличение потребления электроэнергии ухудшает состояние окружающей среды. Рост числа жителей вызывает увеличение потребления электроэнергии. Предположим, что численность населения возросла. Это приведет к увеличению потребления электроэнергии и, следовательно, к ухудшению состояния окружающей среды. Такое изменение окружающей среды приведет к уменьшению числа жителей.

Следовательно, контур компенсирует влияние импульса в вершине 5 – (**Численность людей**). Стабилизирется ли поведение системы? На этот вопрос может дать ответ количественный анализ (анализ устойчивости). В контуре 3–6–4–3 все дуги положительные, и, следовательно, любое отклонение в этом контуре будет усилено.

Данная схема соответствует интуитивным приближенным представлениям о причинности. Расставленные знаки указывают лишь на определенную тенденцию, которую нельзя считать абсолютной и линейной. Взаимодействие факторов может быть более сложным и нелинейным. Например, положительная связь между факторами может иметь место только до определенного уровня, после которого может наступить так называемое насыщение.

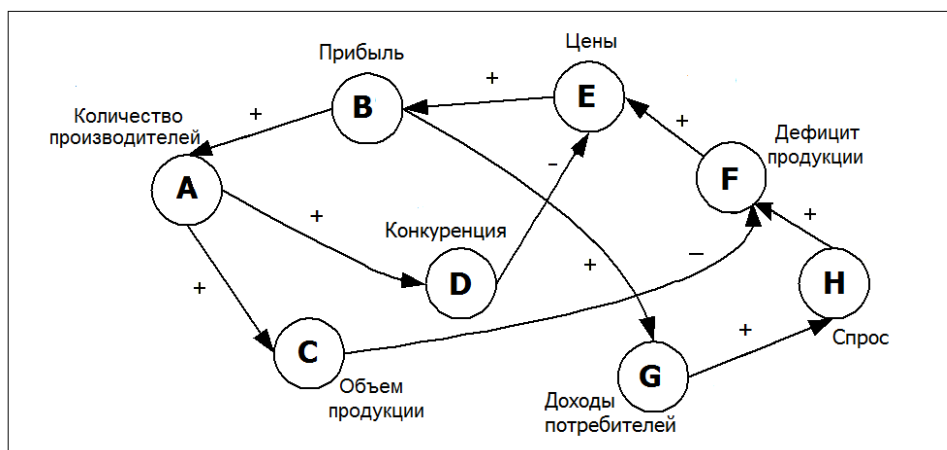


Рис. 3.10. Когнитивная схема экономической ситуации

Когнитивная модель особенно полезна для анализа взаимодействия **трудно формализуемых факторов**, измерение которых является сложной проблемой. Петля В–G–H–F–E–В (рис. 3.10) называется инфляционной. Развитие инфляционного процесса сдерживается контуром А–D–E–B–А (рис. 3.10). Таким образом, схема по рис. 3.10 позволяет на качественном уровне выявить свойства системы.

Существуют системы с отрицательной или положительной обратными связями. Системы с положительной обратной связью являются принципиально неустойчивыми. Действительно, усиление фактора А приводит к росту фактора В, увеличение фактора В ведет к усилению фактора С, увеличение С вызывает рост D, который усиливает А и т.д. (рис.3.11).

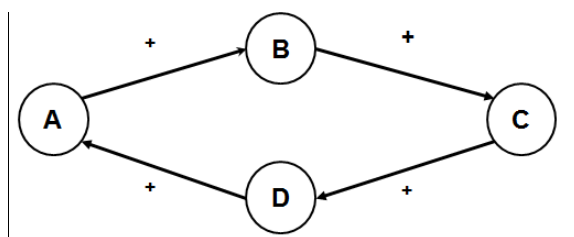


Рис. 3.11. Когнитивная схема системы с положительной обратной связью

В итоге развитие неустойчивости может разрушить систему. Примером может служить так называемый «микрофонный эффект».

В заключение отметим, что рассмотренный в данном разделе специфический вид моделей играет весьма важную роль в моделировании, так как ши-

роко используется как одно из средств анализа при построении многих других моделей, например, рис. 3.12.

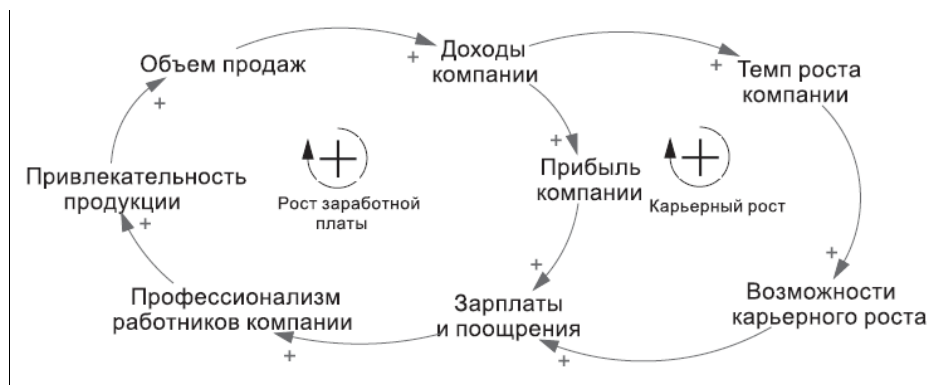


Рис. 3.12. Акселератор квалифицированных кадров

Следует заметить, что на рис. 3.12 отражены не все факторы влияния, например, конкуренция с другими производителями. Это становится ясно при сравнении с рис. 3.10.

3.3. ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ

Имитационное моделирование представляет собой исключительно компьютерное моделирование—исследование свойств объекта путем экспериментирования с его компьютерной моделью, которая только имитирует (воспроизводит) свойства реального объекта.

Имитационное моделирование особо актуально:

- при моделировании сложных систем;
- при отсутствии математической модели системы или объекта;
- при необходимости получения результатов моделирования не позже определенного срока;
- при моделировании действия случайных факторов.

При имитационном моделировании систем:

- модель системы отражает взаимодействие между элементами системы и взаимодействие всей системы с окружающей средой;

- модели систем строятся по блочно-иерархическому принципу: используются модели элементов, модели подсистем и т.д.;
- моделируются только общесистемные процессы;
- если моделируются сложные системы, то процессы в элементах системы не рассматриваются;
- модели элементов строятся формально, отображают связь между выходом и входом (модели типа «**черный ящик**»).

Имитация (simulation), симуляция и моделирование являются в некоторой степени словами-синонимами. Имитация тесно связана с моделированием функционирования сложных систем. Как известно, система есть совокупность объектов, объединенных какой-либо формой регулярного взаимодействия с целью выполнения определенных функций. Примерами систем, которые исследуются методами имитационного моделирования, могут быть промышленное предприятие, транспортная сеть, больница, человек и машина, которой он управляет. Например, при аварии на космическом корабле «Аполлон-13» имитационное моделирование было применено для анализа чрезвычайных мер до того, как были даны команды на их осуществление. Эти меры дали возможность космонавтам благополучно вернуться на Землю после частичного выхода из строя важных энергетических систем.

Имитационное моделирование является одним из способов решения практических задач. Зачастую решение проблемы нельзя найти другим путем. Мощным инструментом моделирования являются компьютеры, ведь они предоставляют легко управляемый виртуальный мир, в котором мы можем создать практически все, что способны представить. Конечно, существует множество различных типов компьютерных моделей: от электронных таблиц, позволяющих моделировать расходы, до сложных инструментов имитационного моделирования, которые помогают исследовать динамические системы (например, потребительский рынок или зону боевых действий).

Сам термин «имитационное моделирование» означает, что мы имеем дело с моделями систем такой степени сложности, что предсказание их поведения принципиально возможно только в ходе **компьютерного модельного**

эксперимента, т.е. в ходе компьютерной имитации поведения системы на определенном промежутке времени.

Исторически первым сложился аналитический подход к исследованию систем. Однако аналитическое исследование процессов функционирования сложных систем, содержащих сотни или тысячи элементов, сталкивается со значительными трудностями, преодоление которых в принципе невозможно.

Имитационные модели являются более универсальными и могут быть построены и при **отсутствии математической модели оригинала**. В свое время С. Уолфрем высказал гипотезу, согласно которой для многих сложных систем не существует простого (математического) описания, их анализ возможен только путем вычислительного компьютерного эксперимента.

Идея имитационного моделирования очень проста. В имитационной модели поведение элементов сложной системы может быть задано в форме алгоритма функционирования этого элемента, т.е. в виде **алгоритмической модели**, которая реализуется компьютерной программой. Такое описание поведения элементов системы в ряде случаев является наиболее простым, естественным и единственно возможным.

Например, В.Н. Глушков предложил использовать методы имитационного моделирования для систем, характеризующихся так называемыми **качественными параметрами**. Подобные параметры измеряются в шкале **наименований** и способны принимать дискретное множество значений. При этом не исключается возможность, что ряд параметров может принимать обычные числовые значения, непрерывные или дискретные. Такой подход применим при моделировании плохо формализуемых систем (например, в медицине), он позволяет получить информацию о поведении системы в шкале качественных параметров и прогнозировать развитие событий.

Помимо сложности, характерной особенностью многих систем является воздействие различного рода внешних или внутренних **случайных факторов**. Исследование влияния случайных факторов на поведение системы связано с проведением множества компьютерных экспериментов (**стохастическое моделирование**) с целью получения статистических данных, на основе которых де-

лается вывод о законах распределения параметров системы как случайных величин (см. п. 3.4).

Естественно, что подобная информация другими способами, кроме компьютерного имитационного эксперимента, не может быть получена. Действительно, экстремальную оценку влияния случайных факторов можно получить по принципу «наихудшего или наилучшего случая». При этом предполагается, что все случайные величины одновременно получили максимально возможные отклонениями в какую-либо сторону. Однако по законам теории вероятности такие события имеют ничтожно малую вероятность, т.е. практически невозможны. Если делать выводы на основе подобного предположения, то результаты дадут весьма затратное, неэффективное и практически неосуществимое решение.

Преимущества имитационного моделирования:

1. Имитационные модели позволяют анализировать системы и находить решения в тех случаях, когда такие методы, как аналитические вычисления и линейное программирование, не справляются с задачей.

2. Разрабатывать имитационную модель гораздо проще, чем математическую, поскольку процесс создания модели будет пошаговым и модульным.

3. Структура имитационной модели естественным образом отображает структуру моделируемой системы.

4. Имитационная модель позволяет отслеживать все объекты системы, учтенные в модели, и проводить статистический анализ.

5. Одним из главных преимуществ имитационного моделирования является возможность проигрывать модель во времени и анимировать ее поведение. Анимация – неоспоримое преимущество при демонстрации модели.

6. Имитационные модели намного убедительнее электронных таблиц при презентации проекта.

Имитационное моделирование позволяет достаточно просто учитывать такие факторы как наличие в системе дискретных и непрерывных элементов, элементов с нелинейными характеристиками, наличие запаздывания и т.д. Взаимодействие элементов сложной системы характерно тем, что оно может

происходить дискретно в определенные моменты времени, которые либо заданы, либо определяются выполнением каких-то условий, либо являются следствием определенных случайных событий. Таким образом, имитируемые процессы в сложных системах могут протекать параллельно, взаимодействуя только в определенные моменты времени. Подобные системы образуют особый класс систем, называемый гибридными.

Другой особенностью имитационного моделирования является возможность его использования даже **при слабой изученности процессов**, протекающих в элементах системы. При попытке построения математической модели подобных процессов обнаруживается, что не все параметры уравнений известны, не все зависимости могут быть представлены в виде конкретных соотношений, механизмы протекания ряда процессов слабо изучены или модель процесса получается настолько сложной, что только для ее реализации необходимы существенные затраты времени и других ресурсов. Однако поведение элементов системы хорошо изучено по результатам их натурных испытаний или специальных экспериментов. В этом случае возможно построение моделей элементов чисто формальными методами регрессионного анализа по принципу «черный ящик».

Большинство имитационных моделей и представляют собой совокупность моделей подобного типа. Это означает, что описание элементов системы представлено в виде формальных функциональных зависимостей между входными и выходными параметрами. Подобные модели обеспечивают выдачу выходного сигнала, если на них воздействует входной сигнал. Поэтому для получения результатов необходимо осуществлять компьютерный «**прогон**» имитационных моделей.

Ранее мы уже отмечали проблемы моделирования сложных систем. Применение моделей типа «черный ящик» связано еще и с тем, что в силу ограниченности возможностей исследования моделей сложных систем, содержащих сотни и тысячи элементов (даже исследований с использованием современных компьютеров), ученые и инженеры вынуждены упрощать описания (модели) элементов системы. Только таким образом можно создать мо-

дель сложной системы, пригодную для проведения компьютерного эксперимента, позволяющую получить какую-либо информацию о поведении системы в ограниченные сроки.

Следует еще раз отметить, что имитационное моделирование зародилось и широко применяется в ходе разработок сложных систем современной техники и исследований по конкретным техническим проектам. В этом случае, с учетом жесткой производственной дисциплины, результаты моделирования должны быть получены **не позднее установленного срока**. Этот временной фактор является весьма серьезным ограничением для всего процесса разработки модели и ее исследования.

Таким образом, при имитационном моделировании сложных систем строится модель, в первую очередь, адекватно отражающая внутреннюю структуру системы, связи между элементами системы, их взаимодействие, динамику изменения состояния каждого элемента и всей системы в целом. Имитационное моделирование доказало свою успешность во многих областях применения. Появление новых методов моделирования и рост вычислительной мощности компьютеров позволяет утверждать, что количество этих областей будет только расти (рис. 3.13).



Рис. 3.13. Области применения имитационного моделирования

В современном имитационном моделировании используются три подхода (методологии): дискретно-событийное моделирование, агентное моделирование и системная динамика (рис. 3.14). Системная динамика предполагает очень высокий уровень абстракции и, как правило, используется для стратегического моделирования.

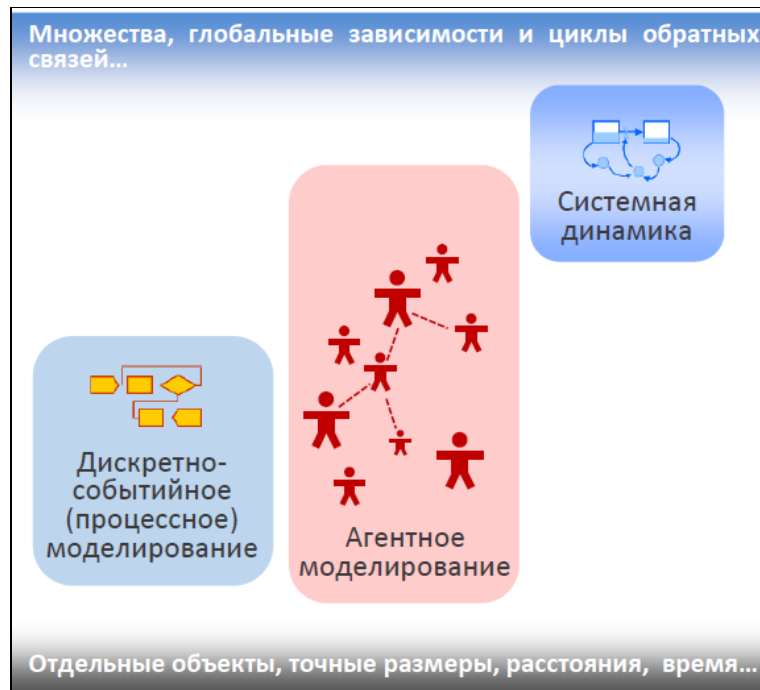


Рис. 3.14. Три метода имитационного моделирования

Дискретно-событийное моделирование поддерживает средний и низкий уровни абстракции. Между ними находятся агентные модели, которые могут быть как очень детализированными, когда агенты представляют физические объекты, так и предельно абстрактными, когда с помощью агентов моделируются конкурирующие компании или правительства государств.

Имитационное моделирование состоит из двух больших этапов: создания модели и анализа полученных с помощью модели результатов с целью принятия решения.

При детальном рассмотрении построение действительно полезной имитационной модели требует большой работы. Сначала разработчик модели должен определить, какие задачи будут решаться с ее помощью, т.е. собственно моделированию в любой его форме должна предшествовать формулировка

цели моделирования. От цели зависит то, какие процессы в реальной системе следует выделить и отразить в модели, а от каких процессов абстрагироваться, какие характеристики этих процессов учитывать, а какие – нет, какие соотношения между переменными и параметрами модели должны быть отражены в модели.

Данный этап можно охарактеризовать как создание концептуальной (содержательной) модели. На нем происходит структуризация модели, т.е. выделение подсистем, определение элементарных компонентов модели и их связей на каждом уровне иерархии.

В имитационном моделировании структура модели отражает структуру реального объекта моделирования на некотором уровне абстракции, а связи между компонентами модели являются отражением реальных связей. Элементы системы, их связи, параметры и переменные, а также их соотношения и законы их изменения должны быть выражены средствами среды моделирования, т.е. в этой среде должны быть определены переменные и параметры модели, построены процедуры вычисления изменения переменных и характеристик модели во времени.

При необходимости для большего понимания процессов, протекающих в модели, должно быть разработано анимационное представление этих процессов. Затем построенная модель должна быть проверена с точки зрения корректности ее реализации.

Следующий этап – это идентификация модели, т.е. сбор данных и проведение измерений тех характеристик в реальной системе, которые должны быть введены в модель в виде значений параметров и распределений случайных величин.

Далее необходимо выполнить проверку правильности модели, которая состоит в том, что выход модели проверяется на нескольких тестовых режимах, в которых характеристики поведения реальной системы известны либо очевидны. Последним этапом работы с моделью является компьютерный эксперимент, т.е. собственно то, ради чего и создавалась модель.

В простейшем случае эксперимент – это выполнение модели при различных значениях ее параметров (факторов) и наблюдение за ее поведением с регистрацией характеристик поведения. Этот вид использования модели называется прогнозом, или экспериментом типа «что будет, если...». Компьютерное моделирование позволяет не только получить прогноз, но и определить, какие управляющие воздействия на систему приведут к благоприятному развитию событий. Более сложные эксперименты позволяют выполнить анализ чувствительности модели, оценку рисков различных вариантов управляющих решений, а также оптимизацию для определения параметров и условий рационального функционирования модели.

Один из важных вопросов – представление и анализ результатов моделирования. Для этого в инструментальной среде могут быть использованы специальные средства для обработки статистической информации, для представления в структурированном или графическом виде полученных данных; интеграция с внешними базами данных и т.п.

Имитационные модели не способны формировать решение, а могут лишь служить в качестве средства для анализа поведения системы в условиях, которые определяются экспериментатором (рис. 3.15).



Рис. 3.15. Имитационная модель системы ПВО

Возникает естественный вопрос: «В каких случаях имитационное моделирование особенно актуально?» Мы определили имитационное моделирова-

ние как экспериментирование с компьютерной моделью в отличие от экспериментирования с реальной системой.

Натурные эксперименты иногда по ряду причин затруднены:

1. Натурные эксперименты могут нарушить установленный порядок или привести к необратимым последствиям.

2. Если составной частью системы являются люди, то на результаты экспериментов может повлиять изменение поведения людей, если им известно, что за ними наблюдают.

3. Поддержание одних и тех же условий при каждом повторении эксперимента сложно или невозможно.

4. Эксперименты с реальными системами не позволяют провести исследование множества альтернативных вариантов.

Дополнительным преимуществом имитационного моделирования можно считать широчайшие возможности его применения в сфере образования и профессиональной подготовки. Разработка и использование имитационной модели позволяет экспериментатору «разыгрывать» на модели разнообразные ситуации. Это, в свою очередь, должно в значительной мере помочь понять и прочувствовать проблему, что стимулирует процесс поиска новых идей и решений.

Когда специалист, экспериментируя с имитационной моделью, достигает подлинного понимания проблемы и начинает свободно управлять ей, он обретает способность видеть содержание своей работы с иных точек зрения. Он захочет проиграть на модели множество альтернативных вариантов, чтобы проверить правильность своих представлений и оценить открывшиеся новые возможности. По сути дела, он использует модель для повышения мастерства управления, позволяющего ему на новом уровне четко установить все существенные последствия вносимых в систему изменений. Возможно, он мог бы проделать это и на реальной системе, но вследствие ее сложности это было бы очень дорого и сопряжено с ошибками. Вот почему он обращается к модели как к средству оценки своих новых интуитивных предположений и умозаключений.

Существует несколько способов построения моделирующих имитационных алгоритмов. Одним из них является принцип повременного моделирования (**принцип Δt**). Суть его состоит в том, что состояние системы анализируется через постоянные промежутки времени. Состояние системы в текущий момент времени определяется на основе состояний системы в предыдущие моменты времени. По истечении шага времени, который равен Δt , объекты обмениваются информацией друг с другом о своих состояниях, которые являются входной информацией для следующего шага. Следовательно, все объекты влияют друг на друга только в отдельных временных точках – на границе шага. Данный принцип является наиболее универсальным принципом построения алгоритмов моделирования широкого класса непрерывных и дискретных систем.

Для увеличения точности моделирования необходимо уменьшение шага по времени. Чаще всего размер шага неизменен. Поэтому возможно, что на отдельных интервалах состояние системы не меняется, а время на вычисления затрачивается. Этот недостаток проявляется в динамических системах с различными значениями параметров инерционности у различных элементов.

В системе можно выделить два типа состояний: обычные состояния, в которых система находится почти все время; особые состояния, связанные с поступлением в систему входных сигналов и т.д. Особые состояния характерны тем, что свойства системы в эти моменты времени изменяются скачком. Как правило, свойства таких систем оцениваются по информации об особых состояниях. Например, в простейшей системе массового обслуживания (СМО) особые состояния появляются в моменты поступления новых заявок на обслуживание или в моменты освобождения канала обслуживания.

Очевидно, что для описания подобных систем должны быть построены алгоритмы по специальному принципу (**принцип особых состояний**). Таким образом, по принципу особых состояний расчет состояния системы осуществляется только в те моменты, когда в ней происходит переход из одного состояния в другое. Это существенно экономит время вычислений, но усложняет систему управления расчетом, да и сами модели. В общем случае события, порождающие особые состояния, могут быть зависимыми.

При моделировании процессов обработки заявок в системах массового обслуживания иногда удобно строить моделирующие алгоритмы с последовательным воспроизведением истории отдельных заявок в порядке поступления их в систему. Алгоритм обращается к сведениям о других заявках лишь в том случае, если это необходимо для решения вопроса о дальнейшем обслуживании данной заявки. Такой принцип называется принципом **последовательной проводки заявок**. В рамках данного принципа один элемент входного сигнала проводится через все блоки системы до ее выхода. Потом проводится следующий элемент и так далее. Реализация подобных моделей затруднена за счет необходимости непрерывного отслеживания памяти системы.

В настоящее время получил развитие принцип **объектно-ориентированного моделирования**, согласно которому элементы модели – объекты, существующие в некоторой среде, – могут обмениваться друг с другом сообщениями. Среда при этом должна выполнять функцию их интерфейса. Каждый элемент имеет имя и посылает сообщения другим объектам. Объекты могут «спать» до момента, пока кто-либо не «разбудит» их сообщением. На одном такте объекты могут обмениваться сообщениями сколько угодно раз, поэтому проблемы задержки также не существует. Подобный подход реализуется, например, при моделировании многоагентных систем. Это системы, динамика функционирования которых определяется не глобальными правилами, а наоборот, поведение таких систем есть результат индивидуальной активности членов группы (агентов). Цель моделирования подобных систем – получить представление о законах поведения многоагентной системы исходя из информации об индивидуальном поведении агента и его взаимодействиях.

Имитационные модели обычно имеют очень сложную логическую структуру, характеризующуюся множеством взаимосвязей между элементами системы, причем многие из этих взаимосвязей претерпевают в ходе выполнения программы динамические изменения. Эта ситуация, в силу ряда особенностей имитационного моделирования, побудила разработать специальные языки и инструментальные системы имитационного моделирования. Известны десятки подобных систем и языков моделирования.

В настоящее время наиболее широко используются три группы инструментальных систем: системы имитационного моделирования непрерывных динамических процессов, системы моделирования дискретных процессов и универсальные системы имитационного моделирования.

Системы имитационного моделирования непрерывных процессов предназначены для моделирования динамических объектов с непрерывным изменением состояния. Как правило, такие объекты описываются системами дифференциальных уравнений с детерминированными или стохастическими параметрами.

Широко известной системой имитационного моделирования дискретных процессов является система GPSS (General Purpose Simulation System), которая появилась еще в 1961 г. Основное назначение GPSS – моделирование систем массового обслуживания или им подобных.

Главными понятиями GPSS являются: транзакт, блок, оператор. Транзакт – это динамический объект, под которым может подразумеваться клиент (в обычном смысле этого слова), требование, вызов или заявка на обслуживание. Транзакты могут создаваться, уничтожаться, задерживаться, сливаться, накапливаться и т.д. Блок представляет собой некоторый самостоятельный элемент моделируемой системы. Каждый блок реализует одну или несколько операций над транзактом или параметрами транзактов. Совокупность блоков составляет моделирующую программу. В целом программа имеет блочную структуру и может применяться для структурно-функционального моделирования не очень сложных систем.

Одной из современных универсальных систем имитационного моделирования является система AnyLogic – профессиональный инструмент имитационного моделирования нового поколения. В AnyLogic поддерживаются все существующие подходы дискретно-событийное моделирования, системная динамика, агентное моделирование и т.д. AnyLogic позволяет строить как стохастические, так и детерминированные модели и проводить анализ результатов моделирования. Подводя итог данному разделу, отметим еще раз, что имитационное моделирование зародилось и стало актуальным вследствие необхо-

димости исследования сложных систем, имеющих важное практическое значение, которые проанализировать обычными способами, кроме компьютерного эксперимента, в заданные сроки, не представляется возможным. Имитационное моделирование позволяет экспериментально исследовать сложные взаимодействия в рассматриваемой системе, позволяет изучать воздействие на функционирование системы различного рода факторов, в том числе и случайных, и проследить их влияние (рис. 3.16).



Рис. 3.16. AnyLogic-модель имитация движения пассажиров в аэропорту

Детальные компьютерные эксперименты с имитационной моделью системы позволяют лучше понять законы ее функционирования и разработать предложения по ее улучшению, которые были бы невозможны без имитации. Имитация сложных систем дает представление о наиболее существенных факторах, определяющих ее свойства.

Имитация позволяет исследовать новые ситуации, которые ранее были неизвестны, и провести предварительную проверку различных стратегий в принятии решений перед проведением экспериментов на реальной системе. Имитация позволяет экспериментально исследовать сложные внутренние взаимодействия в системе, будь то отрасль, отдельная фирма или экономика в целом. Имитационное моделирование является полезным методом решения сложных задач, однако, это не решение всех проблем моделирования.

3.3. МОДЕЛИРОВАНИЕ СТОХАСТИЧЕСКИХ ПРОЦЕССОВ

Моделирование случайных (стохастических) процессов – это одно из важнейших направлений имитационного моделирования. Само понятие «случайный» является фундаментальным как в математике, так и повседневной жизни. Событие называется случайным, если оно достоверно **непредсказуемо**. Случайность играет в нашей жизни и положительную, и отрицательную роль. Например, в ходе эволюции естественный отбор закрепляет положительные свойства живых организмов, которые появились как следствие случайного изменения.

В практике моделирования весьма часто приходится иметь дело с системами, которые в процессе функционирования подвергаются случайным воздействиям окружающей среды или содержат внутренние механизмы, порождающие случайное изменение параметров. Примером могут служить экономические, социальные, экологические, производственные и любые другие системы, в которых значительную роль играет человеческий фактор.

В задаче оптимального размещения ресурсов предполагается, что для всех параметров задачи известны их точные значения. Иными словами все параметры задачи считаются **детерминированными**. Однако на практике точных значений этих параметров просто не существует. Действительно, при выполнении любой работы могут быть **непредсказуемые** перебои с поставками сырья, цены на рынке подвержены колебаниям и т.п. Все эти факторы имеют случайный характер, говорить об их конкретных значениях можно только с определенной **вероятностью**.

Существует несколько точек зрения на природу случайности. Согласно первой точке зрения, случайным нам представляется нечто такое, в чем мы не уловили еще закономерности. Данной точки зрения придерживался Лаплас, который считал, что случайность не присуща самим объектам, а связана только с незнанием, которое в принципе устранимо.

Противоположная точка зрения состоит в том, что случайность является объективным свойством всех явлений. Промежуточная позиция признает су-

ществование как вполне детерминированных явлений, так и в принципе случайных, описываемых статистическими закономерностями: законы Менделя, статистические законы физики, химии, термодинамики и т.п.

В последние годы представители школы И. Пригожина развивают подход, согласно которому случайные и детерминированные периоды сменяют друг друга. Детерминированные процессы постепенно сменяются хаотическими, пока случайность не становится причиной перехода системы в одно из возможных новых устойчивых равновесных состояний.

Говоря о случайных явлениях, прежде всего, обращают внимание на их непредсказуемость, противопоставляя случайность детерминированности, хаотичность упорядоченности. Однако следует подчеркнуть, что случайность – это такой вид неопределенности, которая подчиняется строгой закономерности в виде **распределения вероятности**.

При моделировании случайных процессов значительно изменяется сама методология моделирования: основным методом изучения подобных систем становится **стохастическое моделирование** и статистическая обработка его результатов. Стохастическое моделирование, один из видов имитационного моделирования, представляет собой метод получения статистических данных с помощью компьютера о процессах, происходящих в моделируемой системе, параметры которой изменяются случайным образом с заданным законом распределения.

Сущность метода стохастического моделирования сводится к построению некоего моделирующего алгоритма, имитирующего функционирование системы, случайные воздействия на систему, случайные изменения параметров системы и случайные изменения начальных условий.

Данный алгоритм **многократно реализуется** с помощью компьютерных программных средств. В результате получается серия частных значений искомым величин, статистическая обработка которых позволит получить информацию о свойствах системы (рис. 3.17).

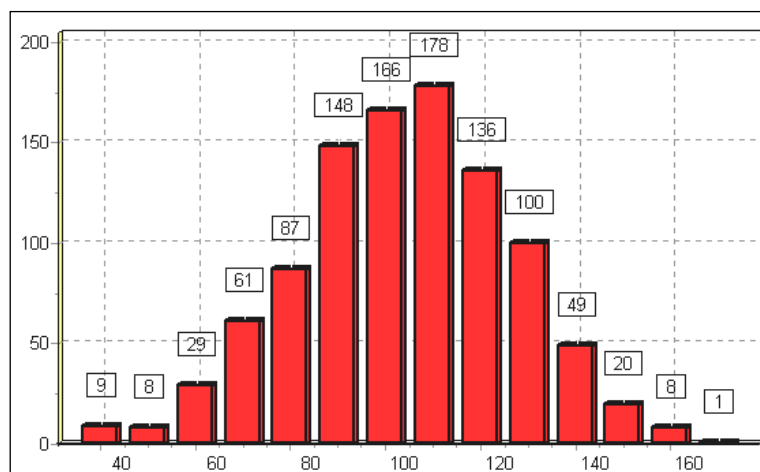


Рис. 3.17. Гистограмма распределения случайной величины

Если количество реализаций велико, то полученные результаты с достаточной точностью могут характеризовать процесс функционирования системы.

В ходе модельных экспериментов с помощью имитационной модели, прежде всего, воспроизводится влияние случайных факторов. При этом, естественно, для получения статистических данных о свойствах объекта или процесса требуется его многократное воспроизведение в ходе моделирования.

Например, предсказать результат бросания монеты трудно. Однако экспериментально можно легко установить, что с равной вероятностью может выпасть либо один результат, либо другой. Чем больше реализаций событий (бросаний монеты), тем больше частота реализации события приближается к вероятности. Это закон больших чисел (так называемая устойчивость частот). Например, Карл Пирсон, английский математик, статистик, биолог и философ, выполнил 24 000 бросаний монеты, герб выпал 12 012 раз, т.е. частота события равна 0,5005. Жорж Луи Бюффон, французский естествоиспытатель и популяризатор науки, провел 4 040 подбрасываний монеты. При этом получил частоту выпадения орла 0,508.

В качестве классической задачи стохастического моделирования обычно рассматривается задача вычисления площади круга методом Монте-Карло. Однако метод Монте-Карло является вычислительным приемом, когда приближенное решение детерминированной задачи получается с помощью генерации последовательности случайных чисел с равномерным законом распре-

деления. При этом собственно задача моделирования случайных событий не ставится. В рамках этого метода, детерминированная вычислительная задача заменяется эквивалентной схемой некоторой стохастической системы, выходные характеристики которой совпадают с результатом решения этой задачи. Естественно, что будет получено приближенное решение, точность которого увеличивается с ростом числа испытаний. Затем подобный метод стал применяться для компьютерной имитации функционирования систем, которые подвержены случайным воздействиям, т. е. появился метод стохастического моделирования.

Теоретической основой метода стохастического моделирования являются **предельные теоремы теории вероятности**. Теорема Бернулли устанавливает связь между относительной частотой появления события и его вероятностью.

Множество случайных явлений (событий, величин) подчиняется определенным закономерностям, что позволяет оценивать их усредненные характеристики, имеющие определенную устойчивость.

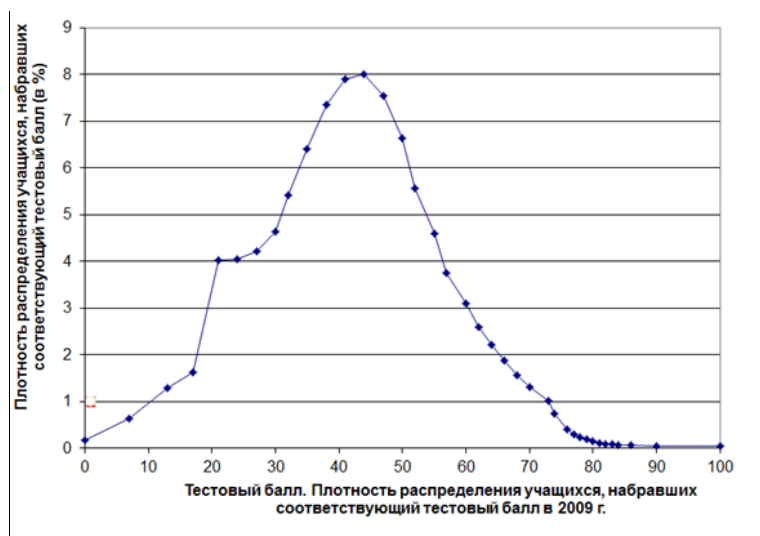


Рис. 3.18. Распределение тестовых баллов ЕГЭ

Характерные закономерности проявляются в **распределениях вероятности** случайных величин, образующихся при сложении множества случайных воздействий (рис. 3.18). Так называемые предельные теоремы и являются выражением этих закономерностей.

Совокупность теорем теории вероятностей, устанавливающих устойчивость средних показателей, еще называют законом больших чисел:

Теорема Бернулли

Если проводится N независимых испытаний, в каждом из которых некоторое событие осуществляется с вероятностью p , то относительная частота появления данного события m/N , где m – число положительных исходов появления события, при $N \rightarrow \infty$ стремится к p . На основе теоремы можно дать определение вероятности события как частоты его проявления.

Теорема Чебышева

Если в N независимых испытаниях наблюдается ряд значений некоторой случайной величины, то при $N \rightarrow \infty$ среднее арифметическое значений случайной величины стремится к ее математическому ожиданию.

Центральная предельная теорема

Если существует N независимых случайных величин, имеющих одинаковый закон распределения, математическое ожидание и дисперсию, то при $N \rightarrow \infty$ закон распределения их суммы неограниченно приближается к нормальному закону.

Теоремы гарантируют высокое качество статистических оценок при бесконечном числе испытаний. Практически значимые результаты получаются уже при сравнительно небольшом числе реализаций.

При статистическом моделировании формальное описание случайных факторов проводится с использованием понятий: случайная величина, случайное событие, случайный процесс и т.п.

Случайной величиной называется такая величина, которая может принимать ряд значений, причем заранее неизвестно какое именно. Таким образом, случайная величина – это величина определенного физического смысла, которая поддается измерению, но ее значения подвержены некоторому неконтролируемому разбросу при повторениях процесса.

Под **случайным событием** понимается всякий факт, который может наблюдаться в данных условиях. Наступление события всегда связано с опре-

деленным комплексом условий. Реализация данного комплекса условий называется **испытанием**.

Различают достоверное событие, которое наступает каждый раз при реализации данного комплекса условий. Невозможное (недостоверное) событие никогда не наступает при реализации комплекса условий. Случайное событие при реализации комплекса условий может наступить или не наступить. Достоверное и невозможное события можно рассматривать как два крайних варианта случайного события.

Частотой события называется отношение числа испытаний, в котором появлялось данное событие, к общему числу испытаний. Частота события обладает определенной устойчивостью, что отражает связь между комплексом условий и возможностью наступления события. Количественной мерой возможности проявления события является его **вероятность** (см. теорему Бернулли). При большом числе испытаний частота события близка к его вероятности.

Свойства вероятностей событий:

1. Вероятность события A заключена между нулем и единицей:
2. Вероятность невозможного события равна нулю, вероятность достоверного события равна единице.

3. Для любого события A вероятность противоположного события \bar{A} (НЕ A) равна $P(\bar{A}) = 1 - P(A)$.

В результате многократного повторения испытаний получается ряд значений случайной величины. В пределе такой бесконечный ряд называется **генеральной совокупностью**. На практике всегда имеют дело с конечным числом значений. Такая совокупность называется выборочной или выборкой. Чем большее число значений содержит выборка, тем ближе она соответствует той генеральной совокупности, частью которой она является.

Случайные величины подразделяются на дискретные, имеющие конечное или счетное множество значений, и непрерывные, значения которых принадлежат какому-либо интервалу числовой оси.

Закономерность распределения значений в генеральной или выборочной совокупности называется распределением. Распределение можно изобразить

в виде зависимости значения случайной величины от частоты появления этого значения. График такой зависимости называется кривой **плотности распределения вероятности**. График плотности распределения вероятности случайной величины показывает, какие ее значения наиболее часто реализуются.

При решении многих практических задач часто достаточно указать отдельные числовые характеристики случайной величины. Среднее значение или математическое ожидание случайной величины:

$$M[x] = \frac{1}{n} \sum_{i=1}^n x_i, \quad M[x] = \sum_{i=1}^n x_i p_i.$$

Здесь x_i – значение случайной величины в i -й реализации, p_i – вероятность появления i -го возможного значения, n – число реализаций.

Среднеквадратичное отклонение определяет разброс случайной величины относительно математического ожидания (среднего значения). Для некоторой выборки случайных величин из генеральной совокупности среднеквадратичное отклонение может быть определено зависимостью:

$$\sigma[x] = \sqrt{\frac{\sum_{i=1}^n (x_i - M[x])^2}{n-1}}.$$

Наиболее полной характеристикой случайной величины является закон ее распределения в виде функции плотности распределения вероятности $f(x)$ и в виде функции распределения вероятности $F(x)$.

$$F(x) = \int_{-\infty}^x f(x) dx.$$

Эти функции связаны соотношением:

Вероятность появления случайной величины, имеющей значение $x \leq a$, можно определить с помощью соотношения: $P(x \leq a) = F(a)$.

Формирование компьютерных реализаций случайных объектов любой природы сводится к генерации и преобразованию последовательностей случайных чисел с **равномерным распределением вероятности** на интервале (0;1). С помощью равномерно распределенных случайных величин можно

сконструировать как случайные события, возникающие с любой заданной вероятностью, так и случайные величины с любым законом распределения.

Простейшими случайными объектами при статистическом моделировании являются случайные события. Пусть в модели необходимо реализовать случайное событие A , которое наступает с заданной вероятностью p . Для компьютерной реализации данного события необходим датчик случайной величины, равномерно распределенной на интервале $(0;1)$. Пусть такой датчик генерирует последовательность случайных чисел: x_1, x_2, \dots, x_i . Определим, что событие A ($A=1$) наступает в том случае, если значение случайной величины x_i удовлетворяет неравенству $x_i \leq p$. В противном случае исходом испытания является противоположное событие HeA ($A=0$).

Рассмотрим построение компьютерной модели реализации случайного события с заданной вероятностью. Для решения задачи необходимо построить электронную таблицу по рис. 3.19.

	A	B	C	D	E
1	P	X_i	A	Кол-во событий A	Кол-во событий HE A
2	0,75	0,1738	1	77	23
3		0,1429	1	Частота A	Частота HE A
4		0,1960	1	0,77	0,23
5		0,2181	1		

Рис. 3.19. Таблица моделирования случайного события

Исходными данными для модели является значение вероятности реализации события p (ячейка A2). В колонку B (диапазон B2:B101) электронной таблицы вводится функция **СЛЧИС()**, которая генерирует последовательность случайных чисел x_i в диапазоне $(0;1)$ с равномерным законом распределения. В ячейку C2 вводится функция **ЕСЛИ()**, которая возвращает значение равное 1, если выполняется неравенство $x_i \leq p$, или значение 0 в противном случае: **=ЕСЛИ(B2<=A2;1;0)**. Далее формула копируется в ячейки столбца C (диапазон ячеек C2:C101).

В ячейках D2 и E2 с помощью функции **СЧЕТЕСЛИ()** подсчитывается количество реализаций событий A ($A=1$) и HEA ($A=0$); ($=СЧЕТЕСЛИ(С1:С101;1)$, $=СЧЕТЕСЛИ(С1:С101;0)$). В ячейках D4 и E4 необходимо рассчитать частоту реализации событий A и HEA с помощью формул: $=D2/(E2+D2)$; $=E2/(E2+D2)$. Естественно, в сумме должна получиться единица.

Можно провести ряд экспериментов при разном количестве реализаций события (50; 100; 500, 1000 и т.д.), и сопоставить значение заданной вероятности p события и частоты его реализации в эксперименте. Чем больше реализаций событий, тем больше частота приближается к вероятности. Это закон больших чисел (так называемая устойчивость частот).

Другая задача моделирования случайных событий связана с построением модели полной группы случайных несовместных событий A, B, C . События происходят с вероятностями P_a, P_b, P_c . Причем всегда реализуется только одно событие: $P_a + P_b + P_c = 1$ (рис. 3.20). Например, полная группа из шести несовместных случайных событий реализуется при бросании игрального кубика.

В этом случае событие A реализуется, если выполняется условие $x_i \leq P_a$, событие B реализуется, если справедливо соотношение $P_a < x_i \leq P_a + P_b$. В свою очередь, событие C реализуется, если справедливо соотношение $P_a + P_b < x_i$ ($x_i < 1$ по определению).

	A	B	C	D	E	F	G	H	I	J
1	P_a	X_i	A	B	C		Кол-во событий A	Кол-во событий B	Кол-во событий C	
2	0,6	0,3438	1	0	0		60	28	12	
3	P_b	0,4234	1	0	0		Частота A	Частота B	Частота C	Сумма частот
4	0,3	0,6923	0	1	0		0,6	0,28	0,12	1
5	P_c	0,0596	1	0	0					
6	0,1	0,2776	1	0	0					
7		0,9607	0	0	1					

Рис. 3.20. Моделирование полной группы случайных событий

Столбцы таблицы C, D, E (начиная с ячеек C2, D2, E2) заполняются по аналогии с предыдущей задачей с использованием функции **ЕСЛИ()**:

$=ЕСЛИ(B2 <= \$A\$2; 1; 0)$, то произошло событие A ($A=1$),

$=ЕСЛИ(И(B2 > \$A\$2; B2 <= \$A\$2 + \$A\$4); 1; 0)$, то произошло событие B ($B=1$),

$=ЕСЛИ(B2 > \$A\$2 + \$A\$4; 1; 0)$, то произошло событие C ($C=1$).

Количество событий A, B, C подсчитывается с помощью функции **СУММ()**. Частоту событий A, B, C можно посчитать с помощью функции **СРЗНАЧ()**. Сумма частот должна быть равна единице (рис. 3.20). Подобная процедура называется «определение исхода по жребию».

Подобная задача моделирования полной группы случайных событий возникает, например, при моделировании процессов «развития» систем. Представим, что система может находиться в одном из состояний A_1, A_2, \dots, A_s . В силу ряда причин состояние системы случайным образом изменяется через дискретные промежутки времени. Находясь в состоянии A_n , система с вероятностью p_n может перейти в состояние A_{n+1} и с вероятностью q_n может перейти в состояние A_{n-1} . Вероятность того, что система сохранит свое состояние, равна $1 - p_n - q_n$.

При компьютерном моделировании часто необходимо моделировать события, результат которых является сложным событием, зависящим от нескольких простых событий. Пусть независимые события A и B имеют вероятность наступления P_a и P_b . Возможными исходами совместных испытаний могут быть события: $(A \cup B)$, $(\text{НЕ } A \cup B)$, $(A \cup \text{НЕ } B)$, $(\text{НЕ } A \cup \text{НЕ } B)$, вероятности, которых соответственно равны:

$$P_1=(P_a \cdot P_b), \quad P_2=(1 - P_a) \cdot P_b, \quad P_3=(1 - P_b) \cdot P_a, \quad P_4=(1 - P_a) \cdot (1 - P_b).$$

В этом случае имеем четыре несовместимых события, вероятности, наступления которых заданы: P_1, P_2, P_3, P_4 .

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	P_a	X_i	A	Y_i	B		A и B	НЕ A и B	A и НЕ B	НЕ A и НЕ B		Вероятности событий				
2	0,6	0,921	0	0,968	0		0	0	0	1		$P_1=P_a \cdot P_b$	$P_2=(1 - P_a) \cdot P_b$	$P_3=P_a \cdot (1 - P_b)$	$P_4=(1 - P_a) \cdot (1 - P_b)$	Сумма
3	P_b	0,926	0	0,786	0		0	0	0	1		0,180	0,120	0,420	0,280	1,000
4	0,3	0,468	1	0,329	0		0	0	1	0						
5		0,592	1	0,486	0		0	0	1	0		Частоты событий				
6		0,418	1	0,608	0		0	0	1	0		A и B	НЕ A и B	A и НЕ B	НЕ A и НЕ B	Сумма
7		0,309	1	0,109	1		1	0	0	0		0,187	0,093	0,413	0,307	1,000
8		0,973	0	0,811	0		0	0	0	1						

Рис. 3.21. Модель для двух независимых событий

Определить результат моделирования можно «по жребию», т.е. методом, который описан выше. Данная задача может быть решена с помощью

двух датчиков случайных чисел отдельно для события A и для события B . Тогда в результате моделирования можно проверить справедливость последних формул для вероятностей (рис. 3.21). Если событие $(A \cup B)$ произошло, то в колонке G электронной таблицы появится значение 1, иначе 0. Это определено с помощью функции **ЕСЛИ**: =ЕСЛИ(И(C2=1;E2=1);1;0). Другие события определяются аналогично. Частоты событий вычисляются с помощью функции **СРЗНАЧ()** по соответствующему столбцу. Сумма частот и вероятностей (рис. 3.21), естественно, равна единице, так как имеем полную группу из четырех несовместных событий: $P_1 + P_2 + P_3 + P_4 = 1$.

Если события A и B являются **зависимыми**, причем событие B наступает с некоторой вероятностью $P(B/A)$ при условии, что событие A произошло. В этом случае из последовательности случайных чисел x_i извлекается очередное число x_m и проверяется справедливость неравенства $x_m \leq P_A$. Если это неравенство справедливо, то наступило событие A . Далее из совокупности чисел x_i извлекается следующее число x_{m+1} и проверяется условие $x_{m+1} \leq P(B/A)$. При выполнении этого условия считается, что наступило и событие B . В противном случае считается, что произошло событие $He B$. Случайное событие B происходит с вероятностью $P(B/A)$, но только в том случае, если произошло случайное событие A . В этом случае требуется второе обращение к функции **СЛЧИС()** (столбец D на рис. 3.22). Таким образом, для реализации события B (столбец E) требуется одновременное выполнение двух условий: $x_i \leq P_a$ и $y_i \leq P_b$, что реализуется с помощью функции: =ЕСЛИ(И(B2<=\$A\$2;D2<=\$A\$5);1;0).

	A	B	C	D	E	F	G	H
1	Pa	Xi	A	Yi	B		Частота A	Частота B
2	0,75	0,2777	1	0,3694	1		0,73	0,41
3		0,0950	1	0,1393	1			
4	Pb	0,6084	1	0,0161	1			
5	0,5	0,2614	1	0,2171	1			
6		0,9499	0	0,7204	0			

Рис. 3.22. Модель зависимых случайных событий

Частота событий подсчитывается с помощью функции **СРЗНАЧ()** по каждому столбцу С и Е на рис. 3.22.

Другой важной характеристикой случайной величины является распределение вероятностей. Можно сгенерировать случайную величину с помощью надстройки Excel «Анализ данных» (рис. 3.23).

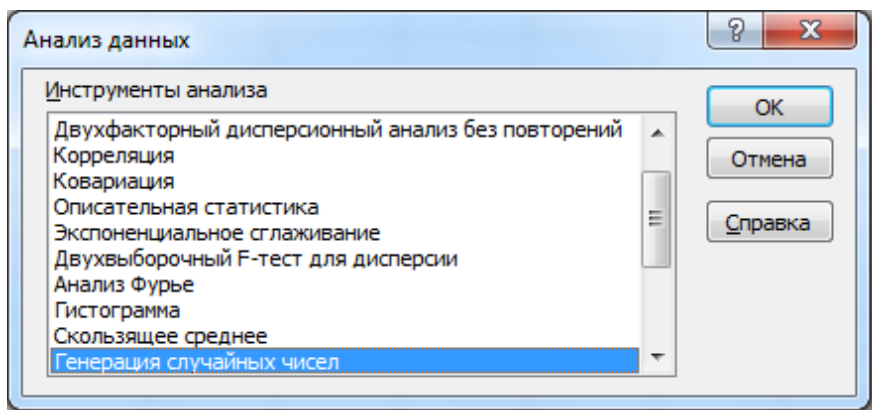


Рис. 3.23. Окно надстройки «Анализ данных»

Для получения доступа к надстройке Excel, ее необходимо подключить: «Файл» – «Параметры» – «Надстройки» – «Перейти» – «Доступные надстройки». Затем пометить надстройку «Анализ данных». Запуск надстройки осуществляется через вкладку Excel «Данные», раздел «Анализ».

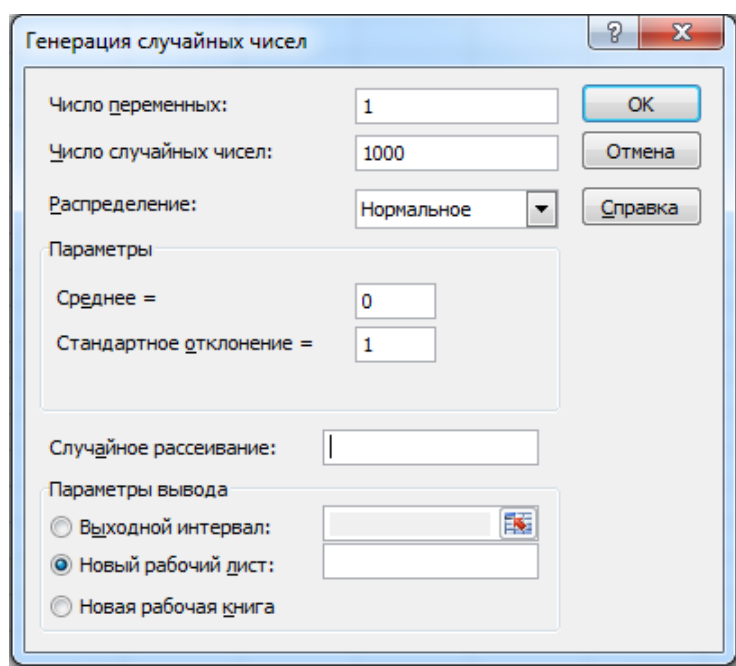


Рис. 3.24. «Пакет анализа». Генерация случайных чисел

Надстройка «**Анализ данных**» позволяет получить равномерное, нормальное, биномиальное и другие распределения случайных чисел (рис. 3.24). Нормальное распределение характерно для величины, которая является результатом воздействия множества случайных факторов, например, данные по количеству баллов, которые получены на ЕГЭ по конкретному предмету в масштабе всей страны (рис. 3.18). Получив выборку случайных чисел можно с помощью надстройки «Пакет анализа» (рис. 3.24) построить гистограмму. Желательно иметь выборку порядка 1 000 значений.

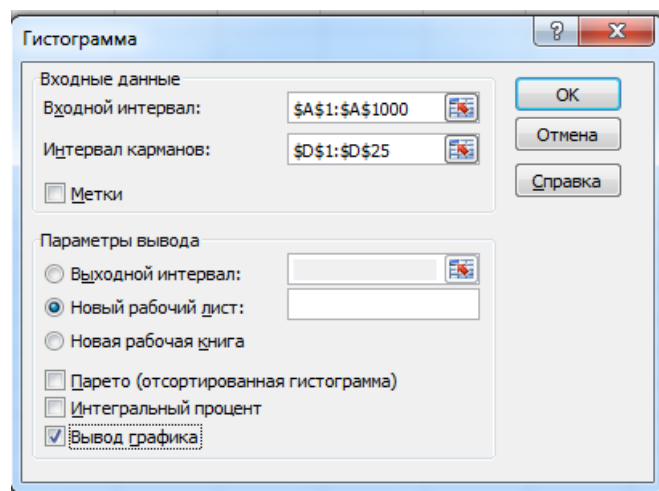


Рис. 3.25. «Пакет анализа». Построение гистограммы

Интервал карманов $\$D\$1:\$D\25 (рис. 3.25) это границы отрезков, для которых определяется частота сгенерированной случайной величины.

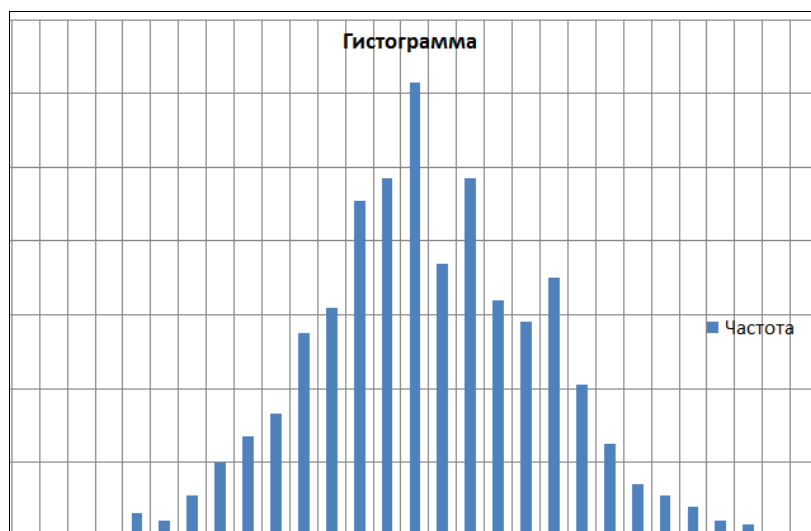


Рис. 3.26. Гистограмма распределения случайной величины

Впрочем, случайные величины с нормальным законом распределения можно получить как сумму случайных величин с равномерным законом распределения. В теории вероятностей доказано (центральная предельная теорема), что при сложении достаточно большого числа одинаково распределенных независимых случайных чисел получается случайная величина, имеющая нормальное распределение.

Как показали исследования, при сложении даже 12 случайных величин с равномерным законом распределения в интервале (0; 1) получается величина, которая может считаться распределенной нормально.

Примеры показывают, что для компьютерной реализации случайных событий, случайных величин с заданным законом распределения необходима программная реализация генератора случайных чисел с равномерным законом распределения на интервале (0; 1).

Проблема генерации цепочки случайных чисел с равномерным законом распределения связана с тем, что компьютер является полностью детерминированной машиной, которая получает случайные числа по вполне конкретному алгоритму. Такая последовательность в действительности является псевдослучайной и может повторяться с определенным периодом. Во многих генераторах случайных чисел используются алгоритмы, в которых каждое последующее число получается на основе предыдущего. Таким образом, последовательность определяется начальным значением. В основе наиболее распространенных генераторов лежит метод вычетов. Каждое последующее число определяется по соотношению: $x_{n+1} = (ax_n + c) \bmod m$, где a , c и m – натуральные числа. Так как требуется число в интервале (0,1), то генератор выдает значение отношения x_{n+1} / m , которое всегда меньше единицы. В общем случае период генерируемой последовательности зависит от этих параметров. Разработаны специальные правила выбора значений a , c и m для получения наибольшего периода последовательности.

В следующей задаче моделируется случайное изменение координат точки X и Y (случайное блуждание). Изменение координат происходит случайно и дискретно. Каждая координата может остаться неизменной, либо изменить

свое значение, добавив +1 или -1. Таким образом, возможна реализация 9 случайных событий:

$$(X, Y); (X, Y+1); (X, Y-1);$$

$$(X+1, Y); (X+1, Y+1); (X+1, Y-1);$$

$$(X-1, Y); (X-1, Y+1); (X-1, Y-1).$$

Вероятности изменения каждой координаты заданы: $PX_a, PX_b, PX_c; PY_a, PY_b, PY_c$. Причем $PX_a+PX_b+PX_c=1; PY_a+PY_b+PY_c=1$.

Таким образом возможны следующие события: событие $A=1$ – координата осталась неизменной; событие $B=1$ – координата изменилась на +1; событие $C=1$ – координата изменилась на -1. Изменение значения каждой координаты происходит независимо. Для решения данной задачи необходимо построить электронную таблицу по рис. 3.27. Исходными данными являются вероятности реализации событий A, B, C . Начальные значения координат: $X=0, Y=0$. На рис. 3.27 события изменения значения для каждой координаты обозначены A_x, B_x, C_x и A_y, B_y, C_y соответственно. В таблице X_i и Y_i сгенерированные функцией **СЛЧИС()** случайные числа. Таблица строится по аналогии с работой, в которой построена модель полной группы случайных событий (рис. 3.20). По результатам моделирования строится фазовая диаграмма $Y(X)$ (диаграмма точечная, точки с прямыми отрезками и маркерами, рис. 3.28).

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Pxa	Xi	Ax	Bx	Cx	Yi	Ay	By	Cy		t	X	Y
2	0,6	0,7030	0	1	0	0,8171	0	0	1		0	0	0
3	Pxb	0,3636	1	0	0	0,3516	1	0	0		1	1	-1
4	0,3	0,2998	1	0	0	0,9798	0	0	1		2	1	-1
5	Pxc	0,7300	0	1	0	0,2319	1	0	0		3	1	-2
6	0,1	0,1429	1	0	0	0,1341	1	0	0		4	2	-2
7	Pya	0,8642	0	1	0	0,5624	0	1	0		5	2	-2
8	0,4	0,4570	1	0	0	0,1203	1	0	0		6	3	-1
9	Pyb	0,2313	1	0	0	0,2667	1	0	0		7	3	-1
10	0,4	0,4876	1	0	0	0,4606	0	1	0		8	3	-1
11	Pyc	0,0692	1	0	0	0,4233	0	1	0		9	3	0
12	0,2	0,3794	1	0	0	0,8818	0	0	1		10	3	1

Рис. 3.27. Электронная таблица для моделирования случайного блуждания

Координаты подсчитываются следующим образом, например, в ячейку **L3** вводится формула **=L2+D2-E2**, а в ячейку **M3** формула **=M2+H2-I2**. В остальные ячейки столбцов L и M формулы копируются. С моделью можно провести серию экспериментов при разных значениях вероятностей.

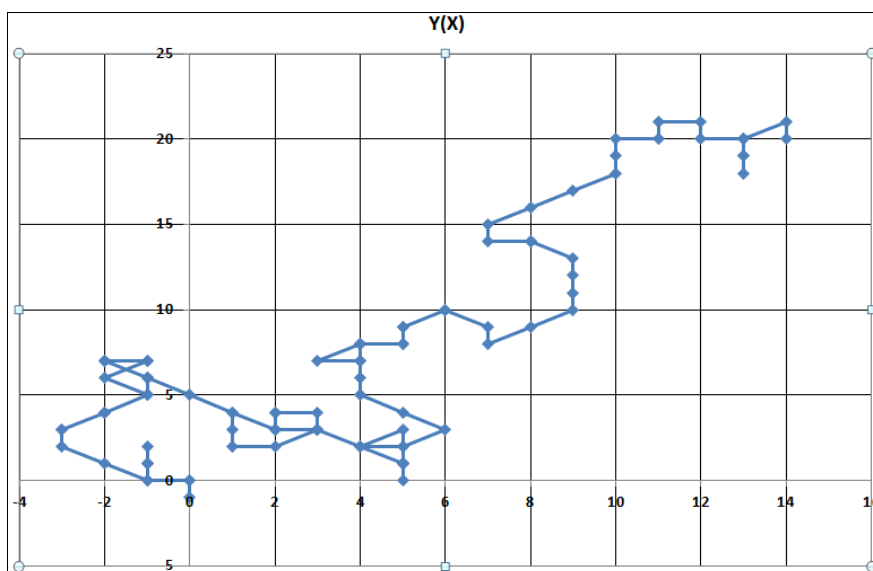


Рис. 3.28. Траектория случайного блуждания

Рассмотрим содержательную задачу моделирования случайных событий. Пусть необходимо выполнить прогноз работы автомобиля. В течение суток автомобиль может быть исправным или стать неисправным. Состояние автомобиля изменяется дискретно с шагом изменения «сутки». Смена состояния автомобиля является случайным событием, так как зависит от многих факторов, которые заранее учесть невозможно. Вероятности смены состояний автомобиля устанавливаются в реальной практике на основе статистических данных по эксплуатации и ремонту, а в данной задаче задаются как исходные величины.

В начальный момент времени автомобиль может быть исправным с вероятностью P_0 . Естественно, что новый автомобиль в начальный момент с достаточно большой вероятностью исправен. Дальнейшее развитие событий может быть следующим: автомобиль в течение суток с вероятностью P_1 может сохранить свое исправное состояние. Если он станет неисправным, то на следующие сутки приступят к его ремонту. Неисправный автомобиль после суток ремонта с вероятностью P_2 вернется к работе, т.е. станет исправным, или его ремонт продолжится. Изменение состояния автомобиля представлено графом (рис. 3.29). Будем считать, что вероятности переходов зависят только от конкретного состояния автомобиля (автомобиль исправен, автомобиль в ремонте), а не от того, каким путем данное состояние достигнуто.

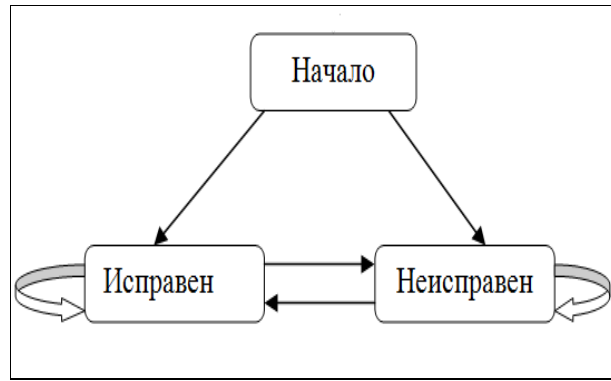


Рис. 3.29. Граф изменения состояния автомобиля

Далее автомобиль находится в каждом возможном состоянии ровно сутки и переходит в новое состояние или сохраняет текущее состояние с учетом вероятности перехода. Модель изменения состояния автомобиля представлена электронной таблицей на рис. 3.30.

	A	B	C	D	E	F
1	P0	P1	P2			
2	0,9	0,8	0,7			
3						
4	День	X_i	A-машина работает	Y_i	B-машина в ремонте	C-машина отремонтирована
5	1	0,035	1		0	0
6	2	0,854	0		1	0
7	3	0,776	1	0,119	0	1
8	4	0,532	1	0,575	0	0
9	5	0,597	1	0,398	0	0
10	6	0,590	1	0,825	0	0
11	7	0,968	0	0,014	1	0
12	8	0,460	1	0,631	0	1
13	9	0,343	1	0,711	0	0
14	10	0,880	0	0,188	1	0

Рис. 3.30. Модель изменения состояния автомобиля

Здесь X_i , Y_i – случайные числа, полученные с применением функции **СЛЧИС()**. Случайное число X_i используется при моделировании события «изменение состояния исправного автомобиля» (событие $A=1$ – автомобиль исправен; событие $A=0$ – автомобиль неисправен). Случайное число Y_i используется при моделировании результата ремонта автомобиля (событие $C=1$ – автомобиль отремонтирован; событие $C=0$ – автомобиль остается в ремонте или пере-

ходит в исправное состояние). Начальное состояние автомобиля (в первый день, ячейка C5, рис. 3.30) определяется формулой: **=ЕСЛИ(B5<=\$A\$2; 1;0)** (определение состояния автомобиля в первый день, как случайное событие). В ячейке E5 вводится число 0, так как по условиям задачи автомобиль поступит в ремонт на следующий день, если в первый день окажется неисправным. Соответственно, в ячейку F5 также вводится число 0, так как ремонт по условиям задачи длится минимум один день. В ячейку F6 также вводится 0, так как в первый день в ремонте не может быть автомобиля.

Случайные числа в ячейках D5, D6 не потребуются. Состояние исправного в первый день автомобиля (C5=1) на следующий день может измениться вследствие поломки – случайное событие. Поэтому в ячейку C6 вводится формула: **=ЕСЛИ(И(B6<=\$B\$2; C5=1); 1;0)**.

Если в текущий день автомобиль окажется неисправным, то на следующий день он поступит в ремонт: **=ЕСЛИ(C6+F6=0; 1; 0)** (ячейка E6, рис. 3.30). Действительно, в один и тот же день автомобиль может быть либо исправен, либо находиться в ремонте. Если автомобиль в этот день отремонтирован, то он на следующий день считается исправным и приступает к работе. Например, ячейка: E6=1 (день=2, автомобиль поступил в ремонт) → F7=1 (день=3, автомобиль отремонтирован) → C7=1 (и в этот же день автомобиль приступил к работе, рис. 3.30).

Результат ремонта определяется формулой **=ЕСЛИ(И(E6=1; D7<=\$C\$2); 1; 0)** (ячейка F7, рис. 3.30), в ремонте находится неисправный автомобиль (E6=1). Результат ремонта – случайное событие. Состояние автомобиля на третий день (ячейка C7, рис. 3.30) и последующие дни определяется тем, что автомобиль либо сохраняет свое исправное состояние, либо поступает из ремонта: **=ЕСЛИ(ИЛИ(И(C6=1; B7<=\$B\$2); F7=1); 1; 0)**. В остальные ячейки столбцов C, D, E, F формулы копируются. Результат моделирования изменения состояния одного автомобиля представлен на рис. 3.31.

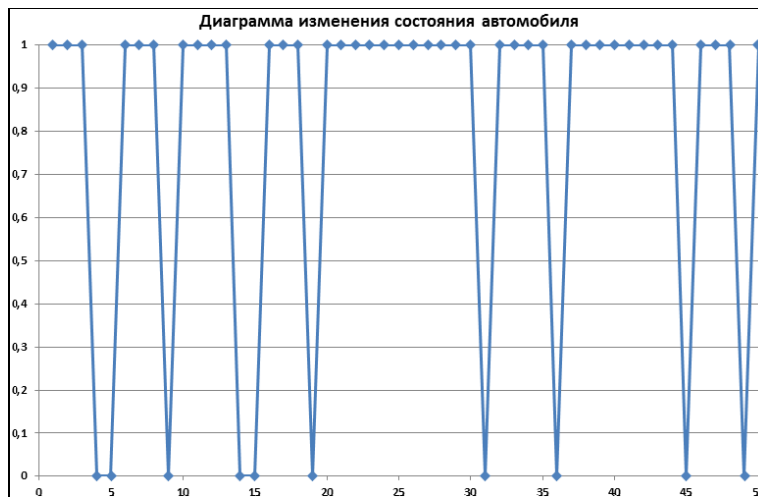


Рис. 3.31. Результат моделирования состояния одного автомобиля

Если автопарк состоит из нескольких автомобилей, то аналогичным образом строятся модели изменения состояния каждого автомобиля. Выполнить это можно путем копирования таблицы по рис. 3.30 и подсчета общего числа исправных автомобилей на каждый день (рис. 3.32).

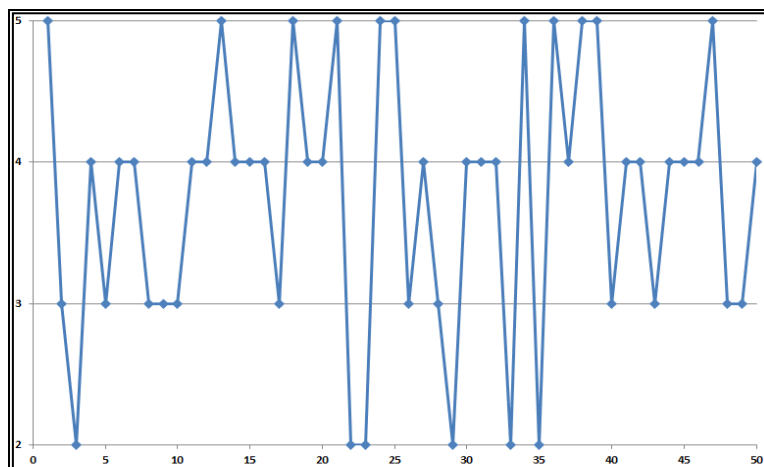


Рис. 3.32. Диаграмма изменения состояния автопарка из пяти автомобилей

Теперь можно решать типичную задачу **имитационного моделирования**.

Например:

Какое количество автомобилей необходимо иметь, с учетом их возможной поломки, чтобы выполнить за заданное время определенный объем работ, который определен в машино-днях? Какова вероятность выполнения этой работы имеющимся парком автомобилей за требуемое время?

В этом случае требуется многократное повторение имитационных экспериментов для получения статистических данных о состоянии парка автомобилей. Это является особенностью стохастического моделирования. Естественно, что в рассмотренных моделях приняты достаточно грубые предположения о вероятностях поломки автомобиля и окончания его ремонта. Однако уточнение модели требует привлечения теории надежности.

Допустим, что фирме, имеющей пять автомобилей, требуется выполнить за 100 дней определенную работу. Объем этой работы в машино-днях известен. Ясно, что 100 дней все автомобили работать не смогут, так как возможны их поломки. Для выяснения возможности выполнения работы был проведен имитационный эксперимент (модель для пяти автомобилей), который состоял из 20 испытаний. Его результаты представлены на рис. 3.33.

P0	P1	P2
0,9	0,8	0,7
Количество автомобилей	Срок выполнения работы (дней)	
5	100	
Объем работы (машино-дни)	Число положительных результатов	Частота реализации события: "Работа выполнена"
370	20	1,00
380	18	0,90
390	10	0,50
400	4	0,25
410	0	0,00

Рис. 3.33. Результат имитационного эксперимента

Таким образом, если работа требует 370 машино-дней или менее, то она будет выполнена. Объем работ в 380 машино-дней будет выполнен с вероятностью 0,9. В этом случае, скорее всего, дополнительной техники не потребуются. Если для выполнения работы требуется 390 машино-дней, то вероятность выполнения работы в срок оценивается как 0,5 и т.д. Можно сделать вывод, что для выполнения такого или большего объема работ необходимы дополнитель-

ные автомобили. В этом случае необходимо изменить модель и повторно провести имитационный эксперимент для нового количества автомобилей.

Ясно, что имитационный эксперимент позволит оценить вероятность выполнения работы имеющимся количеством автомобилей. Кроме того, результаты экспериментов позволят исключить привлечение излишнего количества автомобилей для выполнения работы. Можно, например, рассчитать, на сколько дней необходимо арендовать дополнительные автомобили, чтобы выполнить требуемую работу. Собственно в этом и состоит цель имитационного моделирования для данной задачи – определить оптимальные значения параметров системы.

3.5. МОДЕЛИРОВАНИЕ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ

Потребности обслуживания клиентов АТС, билетных кассах, магазинов и т.п. обусловили возникновение ряда интересных задач моделирования нового типа. Первоначально эти задачи касались преимущественно обслуживания абонентов телефонной сети, покупателей магазинов, установления рационального числа продавцов и кассиров в торговых предприятиях. Со временем оказалось, что такие задачи возникают в самых разнообразных областях техники, экономики и транспорта, в военном деле и в организации производства.

Подобные задачи решаются **теорией массового обслуживания**. В настоящее время теория массового обслуживания – это область прикладной математики, которая занимается анализом процессов в системах производства, обслуживания, управления, где **однородные события повторяются многократно**.

Развитие теории систем массового обслуживания связано с именами датского ученого А. Эрланга, российских математиков А.Н. Колмогорова, Е.С. Вентцель, Б.В. Гнеденко, А.Я. Хинчина. Рассмотрим несколько областей, порождающих задачи из области теории массового обслуживания. Предположим, что на телефонную станцию в **случайном** порядке поступают вызовы. Если в момент поступления вызова на станции имеются свободные линии, то происходит

подключение абонента и начинается разговор в течение того времени, которое необходимо для его завершения. Продолжительность разговора также является **случайной величиной**. Если все линии заняты, то возможно обслуживание с ожиданием или отказ в обслуживании. В первом случае вызов становится в очередь ожидания, при этом интерес представляет **среднее время ожидания** начала обслуживания или **средняя длина очереди**. Во втором случае, поступивший вызов получает отказ, и интерес представляет **вероятность отказа**.

Ситуация, которая создается у театральной кассы при обращении за билетами, напоминает описание предыдущей задачи. Стремление рационально обслуживать потребителей приводит к необходимости изучения закономерностей образования очереди. Знание этих закономерностей поможет решить вопрос о численности касс. Так как содержание касс требует расходов, а потеря клиентов приводит к потере прибыли, то возникает задача поиска **оптимального решения**. Такая же задача актуальна для универсама.

Аналогичная задача возникает при обслуживании одним наладчиком группы станков, которые в случайные моменты времени выходят из строя и требуют ремонта. Длительность ремонта станка является **случайной величиной**. Как велика **вероятность** того, что в определенный момент времени будет ожидать обслуживания то или иное количество станков? Как велико среднее время простоя? Сколько станков можно поручить рабочему для обслуживания?

Мы хорошо знаем из личного опыта, что зачастую отказываемся от обслуживания только из-за длительной задержки начала обслуживания. Если размер очереди велик, то мы откладываем предполагаемую покупку. Точно также мы поступаем при заказе телефонного разговора.

Несколько иная постановка задачи возникает, когда ограничено не время ожидания, а **время пребывания в системе обслуживания**. Такая ситуация возникает, например, при оказании помощи лицам, попавшим в аварию, которые требуют немедленной медицинской помощи.

На основе рассмотренных примеров видно, что можно сформулировать несколько групп типовых задач для описанных выше проблемных ситуаций, которые исследуются теорией массового обслуживания.

Из приведенных примеров ясно, что задачи теории массового обслуживания носят оптимизационный характер и в конечном счете должны определить **вариант** системы, при котором будет обеспечен **оптимальный конечный эффект** от обслуживания.



Рис. 3.34. Структура простейшей системы массового обслуживания

Таким образом, в качестве систем массового обслуживания могут быть представлены различные по своей природе системы. Как видно из рассмотренных примеров, для любой системы массового обслуживания характерно наличие трех отличительных элементов: 1) объекты, у которых может возникнуть потребность в удовлетворении некоторых заявок на обслуживание (источник заявок); 2) агрегаты или узлы для обслуживания заявок; 3) система приема заявок на обслуживание (накопитель заявок – очередь). Структура простейшей системы массового обслуживания представлена на рис. 3.34.

Природа потока требований или заявок на обслуживание обычно не играет роли. Имеют значение лишь **моменты времени поступления требований** на обслуживание, которые могут быть случайными или детерминированными. Для описания входного потока заявок требуется задать закон, определяющий моменты времени поступления заявок. Время обслуживания заявки также считается случайной величиной.

Различают системы с отказами на обслуживание (все каналы обслуживания заняты), системы с ожиданием обслуживания, в которых заявка поступает в

очередь, когда все каналы обслуживания заняты. Ожидание может быть ограниченным и неограниченным. Ограничение ожидания может относиться к длине очереди или ко времени нахождения в ней. Системы массового обслуживания могут также различаться числом каналов обслуживания: одноканальные и многоканальные.

Система массового обслуживания характеризуется также правилами расположения заявок в очереди. Дисциплина очереди определяет принципы обслуживания, которые могут быть следующими: первым пришел – первым обслуживаешься; пришел последним – обслуживаешься первым; случайный отбор заявок; отбор по критерию приоритета; отбор с учетом ограничения времени ожидания.

Поступив в обслуживающую систему, заявка присоединяется к очереди других, ранее поступивших требований. Узел обслуживания выбирает заявки из очереди с целью ее обработки. После завершения обслуживания одной заявки система приступает к обслуживанию следующей и т.д. Цикл функционирования системы массового обслуживания повторяется многократно.

Как правило, цель исследования системы массового обслуживания состоит в **выборе оптимальных параметров** системы по некоторому заданному критерию. При моделировании систем массового обслуживания определяются и **вероятности событий**, например вероятность отказа в обслуживании, и некоторые характеристики случайных величин, например: среднее время ожидания в очереди, средняя длина очереди и т.п. Рассмотрение процесса обслуживания отдельно взятой заявки представляет лишь ограниченный интерес.

Как уже отмечалось, обычно предполагается, что заявки образуют поток, который представляет собой последовательность заявок, а момент времени их появления является случайной величиной. Если с точки зрения обслуживания все заявки данного потока являются равноправными, то играет роль лишь сам факт наступления события поступления заявки. Такие потоки называются потоками **однородных событий**.

Большой практический интерес представляют стационарные потоки, для которых вероятностные характеристики неизменны во времени. В этом случае

математическое ожидание m случайной величины τ_j имеет смысл средней длины интервала между последовательными заявками. Величина $\lambda = 1/m$ имеет смысл среднего количества заявок, поступающих в единицу времени, она часто называется интенсивностью потока заявок.

Для простейшего потока вероятность наступления k событий за время t выражается распределением Пуассона. Функция плотности вероятности для случайной величины τ_j для простейшего потока имеет вид показательного закона: $f(\tau) = \lambda e^{-\lambda\tau}$, где λ – интенсивность потока заявок.

Исследование свойств и закономерностей функционирования систем массового обслуживания – это широкое поле применения методов имитационного моделирования. Реальные системы в силу своей сложности допускают только модельное исследование методами имитационного моделирования. Это яркий пример приложения методов имитационного моделирования для решения целого класса специфических задач, весьма актуальных в проектировании, бизнесе и другой практической деятельности. Для моделирования СМО эффективно использовать систему визуального компьютерного моделирования AnyLogic, которая имеет развитые библиотеки для решения подобных задач.

Например, поиск оптимального количества сотрудников отделения банка для предоставления клиентам требуемого уровня сервиса – типичная задача, решаемая при моделировании систем массового обслуживания.

На первом этапе устанавливается главный критерий уровня сервиса в банке – **средний размер очереди**. Далее выбираются соответствующие параметры системы для задания параметров модели: количество клиентов, интенсивность их прибытия, время на прием одного клиента и естественные отклонения от средних величин, которые периодически возникают, например, часы пик и сложные запросы клиентов.

Затем создается блок-схема, соответствующая структуре отделения банка и его бизнес-процессам. Модель учитывает только факторы, оказывающие влияние на анализируемую проблему. Например, наличие отделения обслуживания юридических лиц отдела не влияет на обслуживание физических лиц,

поскольку эти отделы функционально не связаны. AnyLogic-модель СМО «**отделение банка**» представлена на рис. 3.35.

➔ Объект «**Source**» генерирует заявки на обслуживание. Обычно он используется в качестве начальной точки диаграммы процесса, формализующей поток клиентов. В примере заявками будут посетители банка, а объект «**Source**» будет моделировать их приход в банковское отделение.

▢ Объект «**Queue**» моделирует очередь клиентов, ждущих освобождение банкомата.

🕒 Объект «**Delay**» моделирует задержку на некоторый период времени, в течение которого посетитель банковского отделения тратит свое время на проведение необходимой ему операции.

⊗ Объект «**Sink**» уничтожает заявки на обслуживание. Обычно он используется в качестве конечной точки потока клиентов – уход из банка.

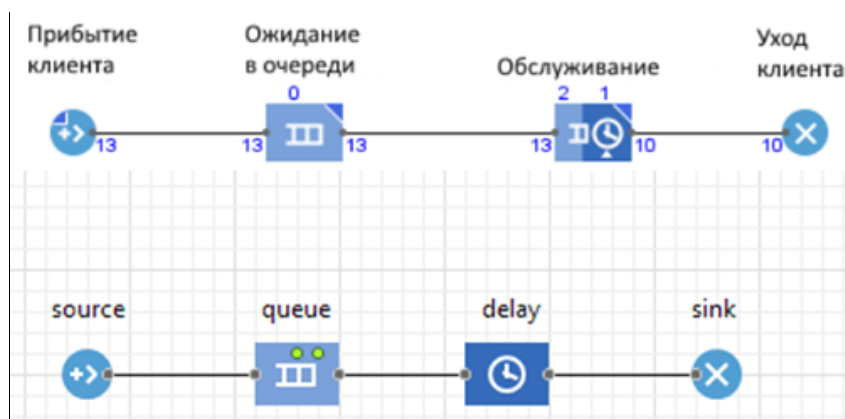


Рис. 3.35. AnyLogic-модель системы массового обслуживания «**отделение банка**»

После загрузки в модель входных данных, имитация запускается, и появляется возможность посмотреть работу отделения банка в динамике, что позволяет обработать и проанализировать результаты. Если средний размер очереди клиентов превысил установленный предел, то количество доступных сотрудников увеличивают, и эксперимент выполняется заново. Этот процесс повторяется, пока не будет найдено оптимальное решение. Изменяя входные данные модели, можно быстро исследовать множество вариантов. Их можно исследовать в динамике и сопоставить друг с другом. В случае необходимости

можно дополнить модель потоком клиентов, которые по разным причинам уходят из банка без обслуживания.

3.6. КЛЕТОЧНЫЕ АВТОМАТЫ И АГЕНТНОЕ МОДЕЛИРОВАНИЕ

Модели на основе **клеточных автоматов** – это яркий пример имитационного моделирования. Можно считать модели типа «клеточный автомат» предшественниками агентного моделирования. Модели вида «клеточный автомат» представляют собой алгоритм функционирования системы, который не имеет аналога в виде математической модели. Динамика изменения состояния автомата может быть исследована только путем модельного **эксперимента**. Заранее предсказать какие-либо закономерности развития невозможно. Еще в 1970 г. А.Н. Колмогоровым давался прогноз, что «с развитием современной вычислительной техники будет во многих случаях разумно изучение реальных явлений вести, переходя прямо к дискретным моделям».

История развития нового направления моделирования начата математиками с изучения необычной абстрактной конструкции – клеточного автомата, который был реализован в форме игры «Жизнь», предложенной математиком Джоном Конвеем в 1970 г. Название связано с тем, что игра имитирует рост, распад и различные изменения в популяции живых организмов, которые происходят по определенным правилам.

Клеточным автоматом называется сеть из дискретных элементов (клеток), меняющих свое состояние в дискретные моменты времени. Чаще всего рассматриваются двумерные клеточные автоматы, элементами которых являются квадратные клетки. Впрочем, форма клеток может быть иной: треугольник, шестиугольник и т.п.

Каждая клетка может находиться в конечном числе состояний. Время в данной модели является дискретным множеством тактов. Состояние каждой клетки автомата в следующий момент времени определяется ее собственным состоянием и состоянием ближайших соседей в предыдущий момент времени.

Среда предполагается однородной, т. е. правила изменения состояний для всех клеток одинаковы. Если это правило зависит от случайных факторов, то автомат называется **стохастическим**, в противном случае – **детерминированным**.

Оказалось, что мир подобных систем необычайно богат, и с их помощью целесообразно моделировать такие сложные процессы как самоорганизация, морфогенез, турбулентность и даже социальные процессы. Кроме того, результаты моделирования на основе клеточных автоматов имеют простую наглядную графическую форму отображения.

Возникающие в процессе игры «Жизнь» ситуации очень похожи на реальные процессы, происходящие при зарождении, развитии и гибели колоний (популяций) живых организмов.

Генетические законы эволюции клеток таковы, что не существует ни одной исходной конфигурации, для которой возможен неограниченный рост популяции. Примеры клеточных структур представлены на рис. 3.36.

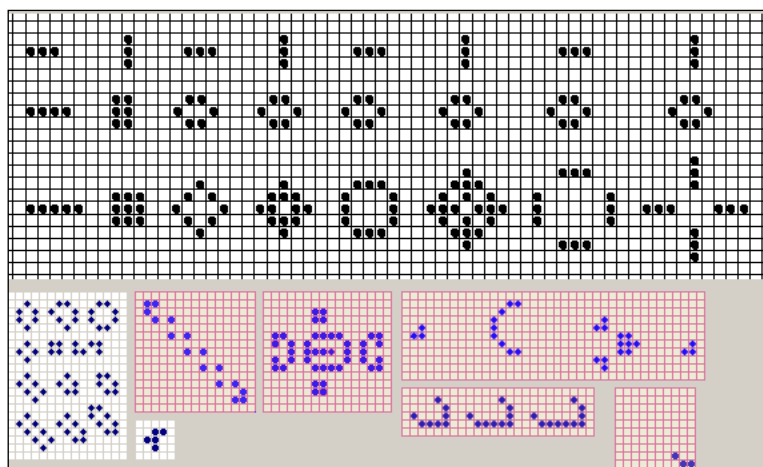


Рис. 3.36. Примеры структур, порождаемых клеточным автоматом «Жизнь»

Начальные конфигурации, которые в течение определенного промежутка времени претерпевают разнообразные изменения, могут закончить свою эволюцию одним из трех способов:

- полностью исчезают из-за перенаселенности или разреженности жизненного пространства, т. е. популяция вырождается;
- переходят в устойчивую конфигурацию – эволюция вышла на стационарный режим;

- выходят на колебательный режим, при котором совершают некий бесконечный цикл превращений с определенным периодом.

Генетические законы следующие:

Правило выживания. Каждая клетка, у которой имеются две или три соседних клетки, выживает и переходит в следующее поколение.

Правило рождения. Если число клеток, с которыми граничит какая-либо пустая клетка, равно трем, то на ней происходит рождение нового организма.

Правило гибели. Каждая клетка, у которой оказывается более трех соседних, погибает от перенаселения. Каждая клетка, вокруг которой свободны все клетки или занята только одна, погибает от одиночества.

Гибель и рождение всех организмов происходит одновременно. Выжившие и вновь рожденные организмы образуют одно поколение эволюции, причем различные начальные конфигурации порождают ряд типовых периодических или стационарных структур. Можно сказать, что клеточные автоматы дают качественно новый тип моделей, которые оказались весьма гибким инструментом для исследования сложных нелинейных процессов в разнообразных системах, в которых из простых правил поведения элементов возникают сложные закономерности. Подобные процессы традиционными математическими методами исследовать чрезвычайно сложно, если вообще это возможно.

Модели на основе клеточных автоматов нельзя отнести к математическим моделям, они описывают алгоритм функционирования системы в рамках специфической организации модельной системы в виде множества взаимодействующих элементов-клеток.

Следующим шагом развития подобного подхода является агентно-ориентированное моделирование (АОМ). Про агента можно сказать следующее:

- **Агент** – индивидуум, состоящий из отдельных частей с определённым набором характеристик и правил, управляющих его поведением и отвечающих за принятие решений.

- **Агент** помещён в определённую среду, живёт в ней и взаимодействует с другими агентами. Агенты способны распознавать и различать различные черты других агентов.

- **Агент** целеустремлён, имеет цели, которые необходимо достичь, исходя из его поведения.

Таким образом, агент – некая сущность, которая имеет автономное поведение, может принимать решения в соответствии с набором правил, может взаимодействовать с окружением и другими объектами. Программной реализацией агента является объект некоторого класса.

- **Агенты не являются клеточными автоматами** и не обязательно обитают в дискретном пространстве (как в игре «Жизнь»). Во многих агентных моделях пространство может вообще отсутствовать. Когда пространство все же необходимо, оно чаще всего является непрерывным (это может быть карта мира или план здания).

- **Агенты – не обязательно люди.** Агентом может быть все, что угодно: транспортное средство, оборудование, проект, организация или даже идея.

- **Агентом может быть объект, кажущийся абсолютно пассивным.** Например, в модели нефтепровода вы можете представить сегмент трубы как агента, задав для него графики техобслуживания, вероятности происхождения аварий, логику проведения ремонтных работ, затраты и т.д.

- **Агентов в модели может быть как много, так и мало.** При этом агенты могут быть как одного типа, так и разных типов (многоагентные модели).

- **Существуют агентные модели, в которых агенты вообще не взаимодействуют друг с другом.** Например, в моделях развития хронических заболеваний индивидуальная динамика агента зависит только от его личных параметров и, в только некоторых случаях, от среды.

При помощи агентов моделируют рынки, конкуренцию и цепочки поставок, деятельность населения и многое другое. Агентные модели позволяют получить представление об общем поведении системы исходя из предположений о поведении ее элементов при отсутствии знания о глобальных законах, то есть в наиболее общем случае.

В имитационном моделировании **агентные модели** могут быть как очень детализированными, когда агенты представляют физические объекты, так и предельно абстрактными, когда с помощью агентов моделируются конкуриру-

ющие компании или правительства государств. Аппарат агентного моделирования – математическая логика с дискретным представлением времени.

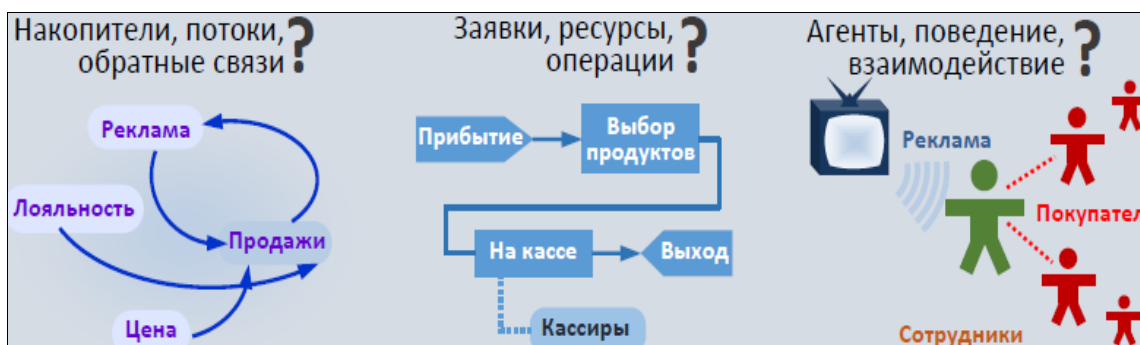


Рис. 3.37. Выбор типа модели

Прежде чем выбрать метод моделирования, следует тщательно исследовать моделируемую систему и цели моделирования. На схеме (рис. 3.37) показано, что конкретная задача, стоящая перед разработчиком, во многом определяет подход к моделированию супермаркета.

Разработчик может построить диаграмму процессов, в которой участвуют покупатели-заявки и кассиры-ресурсы; агентную модель, в которой на покупателей-агентов влияет реклама и общение между собой и с сотрудниками-агентами компании; диаграмму потоков и накопителей, в которой продажи связаны с рекламой, качеством сервиса, ценами и лояльностью клиентов.

Идея агентного моделирования состоит в следующем:

- Сложная система состоит из множества простых объектов.
- Объекты моделируются агентами.
- Смена внутреннего состояния объекта имитируется изменением состоянием агента.
- Агент взаимодействует с другими агентами и со средой обитания подобно объектам реального мира.
- Агент имеет внутреннее представление окружающего мира.
- Агент имеет набор правил поведения для реакции на внешние события.

Главное свойство агентной модели состоит в том, что сложное поведение системы, складывается из комбинации простых правил поведения агентов.

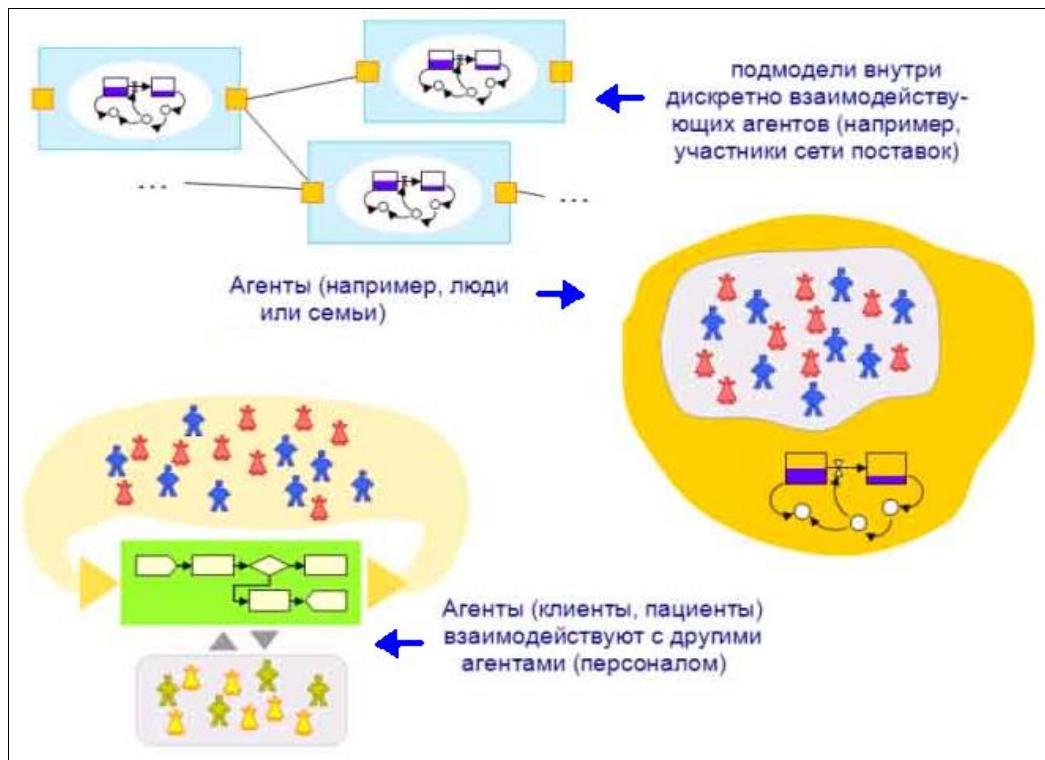


Рис. 3.38. Примеры систем, которые моделируются AOM

Имитация покупателей, розничных и оптовых торговцев, дистрибьютеров, производителей как агентов представлена на рис. 3.39.

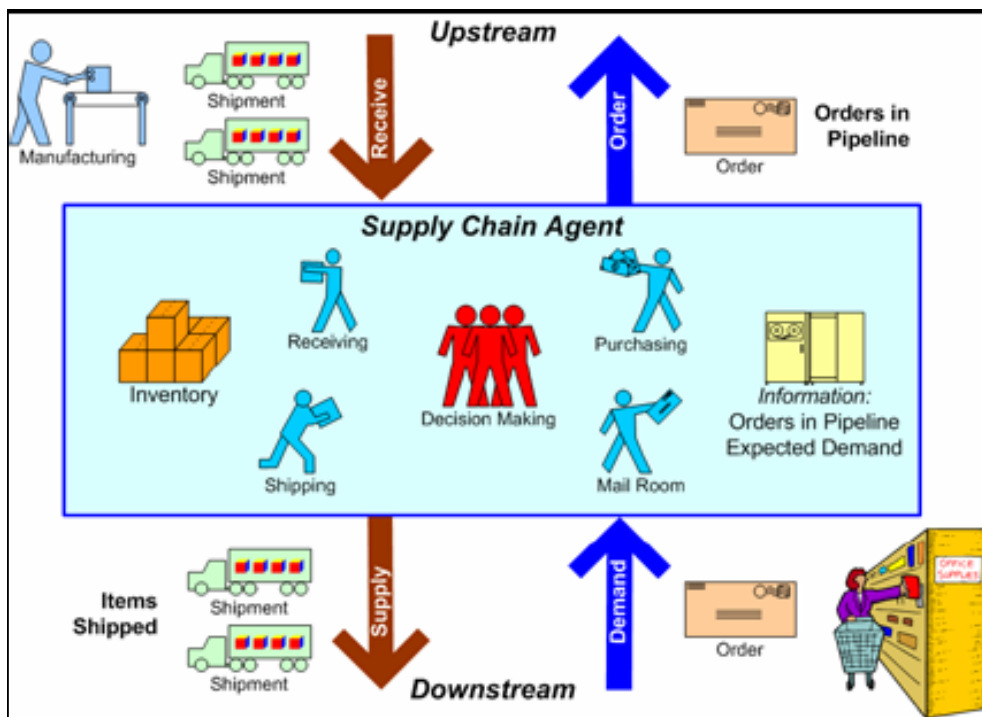


Рис. 3.39. Моделирование цепочки поставок

Для имитации волчьей стаи (рис. 3.40), в которой действуют правила коллективного поведения, обусловленные необходимостью выживания в суровых зимних условиях и эти правила наиболее ярко выражены. Для агентов (волков) правила просты:

- Следовать к средней позиции своих соседей.
- Избегать локального скопления вместе с соседями.
- Следовать среднему направлению своих соседей.



Рис. 3.40. Имитация стаи волков

Имитация экологической системы «хищник-жертва» сводится к заданию правил поведения «хищников», которые питаются «жертвами» и перемещаются для их поиска. «Жертвы» питаются растительным кормом и перемещаются для его поиска и стараются избежать встречи с «хищниками» (рис. 3.41).

Кормовая база для каждого вида ограничена численностью «жертв» и объемом растительного корма, который возобновляется, в ареале обитания «жертв» по мере роста растений. Более сложные взаимодействия в экосистеме отражены на рис. 3.42.

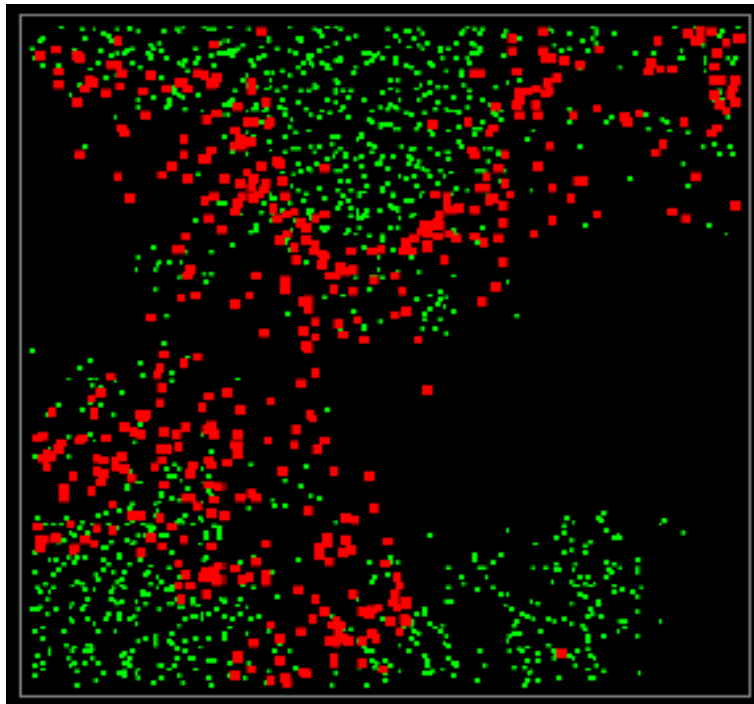


Рис. 3.41. Взаимодействия в системе «хищник-жертва». AnyLogic-модель

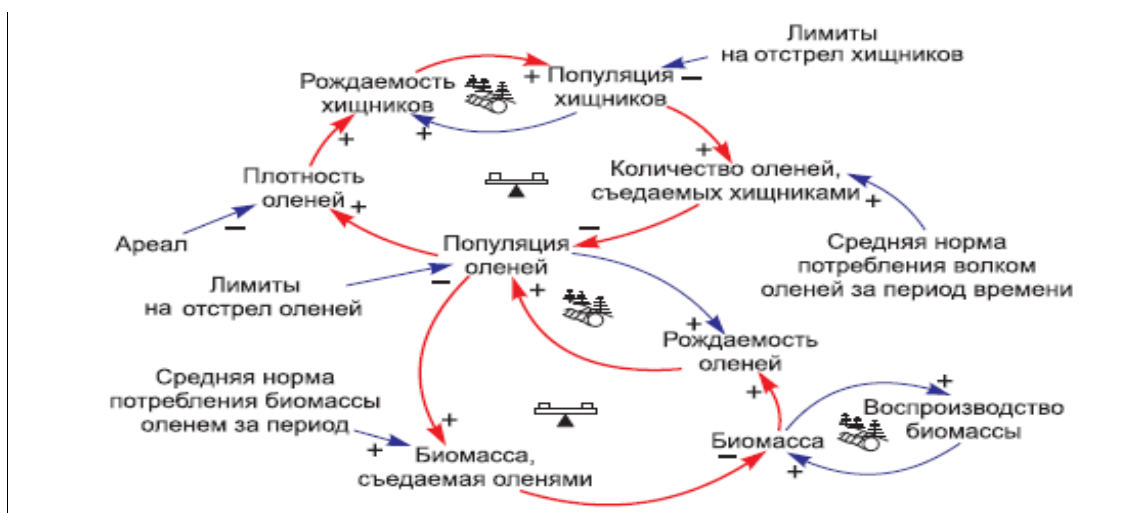


Рис. 3.42. Взаимодействия в системе «хищник-жертва» с учетом отстрела

Таким образом, области применения агентного моделирования это:

- Экономические и логистические процессы.
- Развитие цивилизации.
- Социальные отношения.
- Политика.
- Когнитивные аспекты общения.
- Физические, химические и биологические процессы.
- Экология.

3.7. РЕГРЕССИОННЫЕ МОДЕЛИ И КОРРЕЛЯЦИОННЫЙ АНАЛИЗ

В настоящем разделе мы познакомимся с методами построения моделей типа «**черный ящик**». Значение моделей данного вида при моделировании сложных систем было раскрыто в предыдущих разделах.

В имитационном моделировании сложных систем с применением моделей «**черный ящик**» прослеживается одно из характерных явлений современной науки – стремление перейти к изучению сложных, больших и плохо организованных систем.

В последние десятилетия четко определился подход в изучении подобных систем – применение имитационного моделирования и использование идей и методов математической статистики для построения моделей элементов системы. Методы математической статистики по существу хорошо обоснованная система формализации эмпирических наблюдений, когда исследователь сознательно отказывается от детального традиционного изучения механизмов всех протекающих в системе процессов.

Одним из результатов таких исследований является построение регрессионной модели. При этом немедленно возникают проблемы выбора стратегии эксперимента, обработки результатов эксперимента и т.п. Это типичные задачи математической статистики. Некоторые методы статистического подхода рассматриваются нами в данном и следующем разделах.

Пусть эксперимент поставлен и проведен, результаты наблюдений получены. Теперь их необходимо представить в какой-то компактной форме, удобной для практического применения. Пусть явление или процесс характеризуется двумя величинами X и Y , значения которых фиксируются в процессе эксперимента или наблюдений и имеют случайный характер. Задачей исследования является установление так называемой статистической зависимости между случайными величинами, при которой изменение одной из величин влечет за собой изменение другой величины. Поскольку X и Y – случайные величины, то для заданного значения X величина Y не может быть предсказана точно, а может быть указана лишь с определенной вероятностью. Такая вероятностная

функциональная зависимость между случайными величинами называется регрессией.

Задачами регрессионного анализа является определение собственно функции регрессии. Впрочем, следует помнить, что уравнение регрессии лишь формально устанавливает соответствие между переменными, хотя реально такой зависимости может и не быть (ложная регрессия). Уравнение регрессии представляет собой математическую форму зависимости измеряемой величины от влияющих на нее факторов.

Выбор той или иной формы зависимости определяет точность, с которой модель описывает реальную действительность. Регрессионный анализ сводится к тому, что на основе экспериментальных данных или данных наблюдений определить коэффициенты модели и выполнить оценку ее адекватности.

Существуют различные виды регрессий:

1. Простая и множественная регрессия. Простая регрессия отражает зависимость между двумя переменными, т.п. уравнение регрессии представляет собой функцию одного аргумента, для множественной регрессии уравнение регрессии есть функция многих аргументов.

2. По форме зависимости различают линейную и нелинейную регрессии.

Построение регрессионной модели связано с корреляционным анализом. В регрессионном анализе исследуется форма связи, а в корреляционном анализе оценивается ее сила. Корреляция в широком смысле означает взаимосвязь между объективно существующими явлениями. Если случайные величины **причинно обусловлены**, тогда в вероятностном смысле можно утверждать об их связи, т.е. о **корреляции**.

Корреляционный анализ проводится с использованием коэффициента корреляции, который и характеризует силу связи между величинами:

$$r_{xy} = \frac{\sum_{i=1}^N [(x_i - \bar{x}) \cdot (y_i - \bar{y})]}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2 \cdot \sum_{i=1}^N (y_i - \bar{y})^2}}$$

Здесь \bar{x} и \bar{y} – средние значения соответствующих величин, r_{xy} – коэффициент корреляции. Значение коэффициента корреляции заключены в пределах от -1 до $+1$. В случае $r_{xy}=0$ связь между X и Y отсутствует. Анализ выполняется с использованием **корреляционной функции R_{xy}** (в электронных таблицах Excel-функция КОРРЕЛ).

Построение регрессионной модели обычно проводится в несколько этапов:

1. Априорное исследование проблемы. Проводится конкретизация описания процессов и явлений, формулируются гипотезы о взаимозависимости явлений.

2. Формирование перечня факторов и их анализ. Исходя из содержательного описания объекта и предварительной информации теоретического и практического характера выявляются параметры модели, производится их разделение на зависимые и независимые. На основе информации о функционировании объекта формулируется гипотеза о форме связи между параметрами модели: линейная, нелинейная, простая или множественная.

На данном этапе в модель включаются все факторы, которые, по мнению исследователя, могут оказать какое-либо влияние на свойства системы, т.е. на ее зависимые переменные. Таких факторов может оказаться достаточно много. Действительно, какие факторы являются **главными**, а какие **второстепенными** еще предстоит выяснить.

Таким образом, возникает дилемма: мало факторов – модель может иметь низкую адекватность; много факторов – модель, как правило, неустойчива; малые изменения параметров или объема выборки, по которой строится регрессионная модель, могут привести к существенным изменениям.

Пусть исходная информация о свойствах объекта, полученная в ходе экспериментов, представлена таблицей 3.1. Здесь y – зависимый параметр, а X_1, X_2, \dots, X_n факторы, определяющие, **по мнению исследователя**, свойства объекта, т.е. значение y . По этим данным необходимо построить регрессионную зависимость: $y=f(X_1, X_2, \dots, X_n)$.

Для снижения размерности модели, т.е. уменьшения количества факторов, необходимо провести анализ их значимости и взаимозависимости (мульт-

тиколлинеарности), т.е. их взаимной парной корреляции. Так как на этапе предварительного анализа в число факторов могут быть включены и малозначимые факторы, которые оказывают несущественное влияние на величину зависимой переменной.

Результаты эксперимента

Таблица 3.1

X1	X2	X3	X4	X5	Y
0,164663	2,619369	8,321767	0,119369	4,5285	22,68506
0,310293	3,424706	9,707507	0,924706	3,020979	15,2339
0,455923	2,726734	11,09325	0,226734	0,338279	1,944358
0,601553	2,575317	12,47899	0,075317	0,322542	2,023601
0,747183	2,542886	13,86473	0,042886	0,152486	1,362907
0,892813	3,266111	15,25047	0,766111	0,062932	1,134693
1,038443	2,56584	16,63621	0,06584	3,093885	16,53766
1,184073	2,981757	18,02195	0,481757	2,241391	12,55099
1,329703	3,188881	19,40769	0,688881	0,013524	1,714155
1,475333	2,789835	20,79343	0,289835	0,305511	3,502513
1,620963	3,426532	22,17917	0,926532	0,410757	4,382427
1,766593	3,220105	23,56491	0,720105	1,798101	11,69751
1,912223	2,685862	24,95065	0,185862	4,145915	23,83909
2,057853	2,605505	26,33639	0,105505	0,017298	3,622187
2,203483	2,506872	27,72213	0,006872	0,226433	5,117287

Для отбора главных и исключения взаимозависимых факторов необходимо построить корреляционную матрицу, которая будет иметь следующий вид (таблица 3.2):

Корреляционная матрица

Таблица 3.2

	A	B	C	D	E	F	G
1		x1	x2	x3	x4	x5	y
2	x1	1,0000	0,1831	1,0000	0,1831	-0,0977	0,7895
3	x2	0,1831	1,0000	0,1831	1,0000	-0,0756	0,1138
4	x3	1,0000	0,1831	1,0000	0,1831	-0,0977	0,7895
5	x4	0,1831	1,0000	0,1831	1,0000	-0,0756	0,1138
6	x5	-0,0977	-0,0756	-0,0977	-0,0756	1,0000	0,5109
7	y	0,7895	0,1138	0,7895	0,1138	0,5109	1,0000

Качественная оценка силы связи двух величин может быть выполнена по шкале Чеддока (табл. 3.3):

Сила связи	Значение коэффициента корреляции при наличии:	
	Прямой связи	Обратной связи
Слабая	0,10–0,30	(–0,10)–(–0,30)
Умеренная	0,30–0,50	(–0,30)–(–0,50)
Заметная	0,50–0,70	(–0,50)–(–0,70)
Высокая	0,70–0,90	(–0,70)–(–0,90)
Весьма высокая	0,90–0,99	(–0,90)–(–0,99)

Мультиколлинеарная зависимость факторов присутствует, если коэффициент парной корреляции $|R_{x_i x_j}| \geq 0,7$. Опыт показывает, что если для двух факторов $|R_{x_i x_j}| \geq 0,7$, то одну из переменных можно исключить. Обычно в модели оставляют тот фактор, который является хорошо управляемым, т.е. исследователь представляет каким образом можно изменить его значение. Кроме того, факторы, для которых $|R_{x_i y}| \leq 0,1$, т.е. фактически не связанные с y , также подлежат исключению как малозначимые.

На этапе построения регрессионной модели выбирается класс функций, в рамках которого строится регрессионная модель и методом наименьших квадратов определяются коэффициенты модели. Модель строится уже без учета исключенных факторов.

Рассмотрим простой пример построения регрессионной модели. Предположим, выполнено N измерений величин X и $Y - (x_i, y_i)$. Допустим, что зависимость y от x предполагается линейной: $f(x) = a x + b$.

Результаты измерений (x_i, y_i) содержат определенную (неизвестную) случайную величину, которая обусловлена погрешностями измерений и неконтролируемыми случайными факторами. Требуется построить такую линейную зависимость, которая наилучшим образом описывает результаты наблюдений. Иными словами, требуется определить параметры линейной функции a и b .

Для решения данной задачи наиболее часто используется метод наименьших квадратов (МНК). Параметры линейной функции должны иметь такие значения, чтобы в целом точки $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ лежали как можно ближе к прямой.

Назовем отклонением разность вида: $f(x_i) - y_i$. Подберем параметры a и b таким образом, чтобы сумма квадратов отклонений была минимальна:

$$\sum_{i=1}^N (f(x_i) - y_i)^2 \rightarrow \min .$$

В этом и состоит требование метода наименьших квадратов. Данная сумма есть функция искомых параметров a и b :

$$F(a, b) = \sum_{i=1}^N ((a \cdot x_i + b) - y_i)^2 .$$

Для отыскания min найдем частные производные функции F и приравняем их нулю:

$$\frac{\partial F}{\partial a} = 0, \quad \frac{\partial F}{\partial b} = 0 .$$

Последние соотношения дают систему линейных уравнений относительно неизвестных a и b , решение которой представляется следующими формулами:

$$a = \frac{N \cdot \sum_{i=1}^N x_i y_i - \sum_{i=1}^N x_i \cdot \sum_{i=1}^N y_i}{N \cdot \sum_{i=1}^N x_i^2 - (\sum_{i=1}^N x_i)^2}, \quad b = \frac{\sum_{i=1}^N y_i - a \cdot \sum_{i=1}^N x_i}{N} .$$

Пример регрессионной модели представлен на рис. 3.43.

Для оценки точности регрессии используется коэффициент детерминации, который характеризует степень соответствия уравнения регрессии, имею-

щимся эмпирическим данным:

$$D = 1 - \frac{\sigma_{ост}^2}{\sigma_{общ}^2} .$$



Рис. 3.43. Линейная регрессионная модель, полученная с использованием функции Excel «линия тренда»

Величина $\sigma_{общ}^2$ – общая дисперсия, характеризует разброс наблюдаемых величин y_i относительно среднего значения \bar{y} ; $\sigma_{ост}^2$ – остаточная дисперсия, характеризует отклонения наблюдаемых величин y_i от значений, полученных по регрессионной зависимости $f(x)$:

$$\sigma_{общ}^2 = \frac{\sum_{i=1}^N (y_i - \bar{y})^2}{N - 1}, \quad \sigma_{ост}^2 = \frac{\sum_{i=1}^N (y_i - f(x_i))^2}{N - 1}.$$

На основе коэффициента детерминации может быть проведена оценка адекватности регрессионной модели. Если $D=1$, значит $\sigma_{ост}^2 = 0$, т.е. все наблюдаемые точки соответствуют построенной регрессионной зависимости. В противном случае при $D \approx 0$ имеет место ложная регрессия.

Для построения уравнения регрессии необходимо создать таблицу числовых данных. Затем по таблице необходимо построить диаграмму «точечная». При этом выбрать вариант диаграммы, которая отображает табличные данные в виде точек. Далее необходимо выбрать команду «Добавить линию

тренда» (пункт меню «**Диаграмма**»). В открывшемся диалоговом окне «**Формат линии тренда**» (рис. 3.44) выбрать тип зависимости.

На вкладке «**Формат линии тренда**» (рис. 3.44) выбрать опции «**Показывать уравнение**» и «**Поместить величину достоверности**». Результат представлен на рис. 3.43.

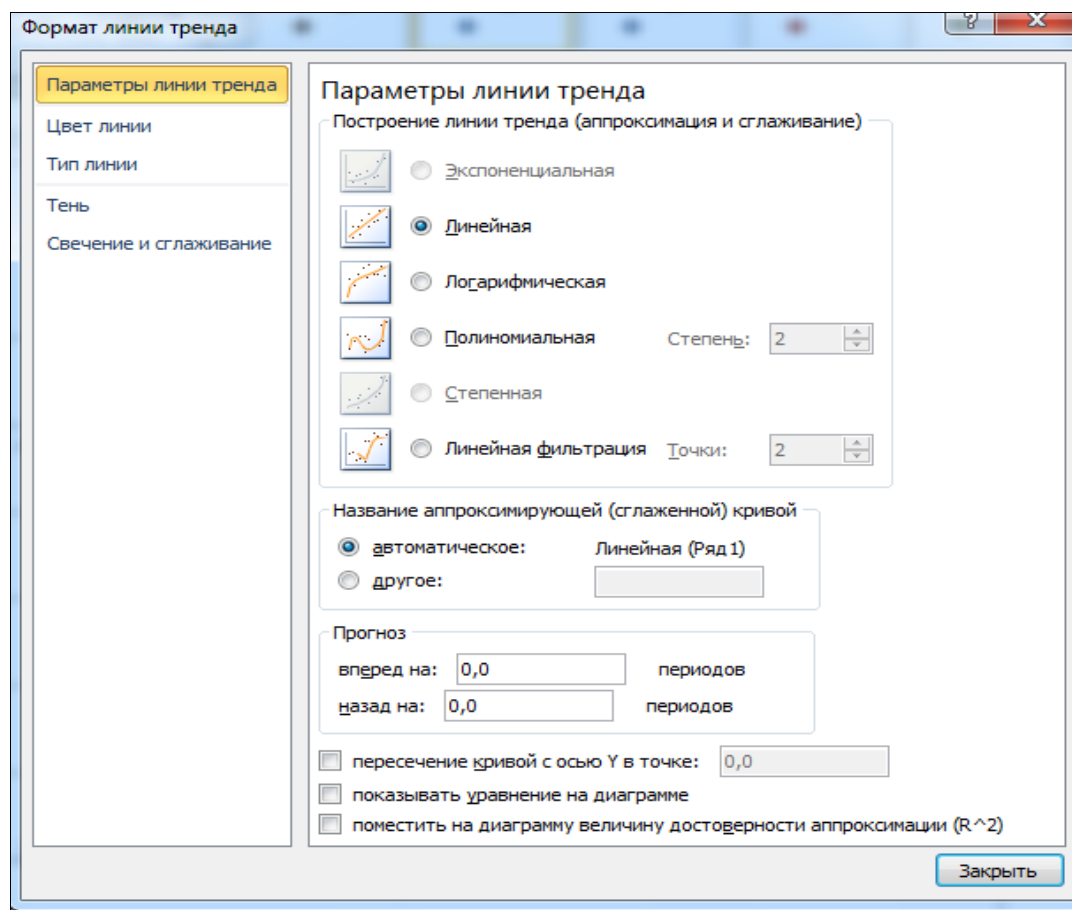


Рис. 3.44. Диалоговое окно «**Формат линии тренда**»

3.8. ПЛАНИРОВАНИЕ ЭКСПЕРИМЕНТОВ

Ранее отмечалось, что в ряде случаев математические модели строятся путем статистической обработки результатов эксперимента с объектом моделирования. Такой подход применяется, когда законы протекания моделируемых процессов сложны и слабо изучены. Естественно, конечной целью подобного эксперимента является установление количественных зависимостей неко-

торых свойств объекта от параметров, которые предположительно влияют на эти свойства.

Со времен Ньютона и до начала XX века точные науки стремились иметь дело с системами, в которых можно было выделить явления или процессы одной физической природы, зависящие от небольшого числа параметров. Результаты экспериментов представлялись простыми и хорошо интерпретируемыми математическими зависимостями, которые получали статус научных законов.

В течение более 200 лет считалось, что единственно правильной является методология **однофакторного эксперимента**. Предполагалось, что исследователь мог с необходимой степенью точности стабилизировать все независимые переменные (факторы) своей системы, затем, поочередно варьируя некоторыми из них, установить интересующие его зависимости.

В начале XX века на основе методов математической статистики были сделаны первые шаги по изучению плохо организованных систем, в которых нельзя выделить отдельные физически однородные явления и разграничить их действие. Это обусловлено тем, что путем моделирования стали исследоваться реальные производственные и технологические процессы и системы. Примерами могут служить любые химико-технологические процессы, которые включают в себя процессы теплопередачи, диффузии, переноса и химической кинетики протекающих реакций.

В последние десятилетия четко определился подход в изучении подобных систем – применение имитационного моделирования и использование идей и методов математической статистики для построения моделей элементов системы. Методы математической статистики по существу хорошо обоснованная система формализации эмпирических наблюдений, когда исследователь сознательно отказывается от детального традиционного изучения механизмов всех протекающих в системе процессов.

Эксперимент с реальным объектом, как правило, достаточно сложное и дорогостоящее мероприятие, поэтому он должен выполняться по определенному **плану**. Основная задача планирования эксперимента – обеспечение по-

лучения необходимой информации об исследуемой системе при ограничениях на ресурсы, которые могут использоваться в эксперименте.

К настоящему времени сложилась математическая **теория планирования эксперимента**, которая имеет методы, позволяющие повысить эффективность эксперимента. Рассмотрим основные понятия теории планирования эксперимента.

Модель, которая будет построена путем обработки результатов эксперимента – модель типа «черный ящик». При таком подходе у исследуемого объекта выделяют только входные и выходные переменные, которые называются **факторами** и **откликами**. Факторы – это управляемые переменные, значения которых в ходе экспериментов можно менять. Каждый фактор в эксперименте может принимать определенные значения, которые называются уровнями. Набор **уровней** значений факторов определяет одно из возможных состояний системы в эксперименте.

Классический эксперимент является однофакторным: варьируется один из факторов, а остальные поддерживаются на постоянном уровне. Ясно, что в условиях большого числа факторов такая схема проведения эксперимента непригодна. В этом случае единственным возможным является статистический подход к планированию эксперимента, когда исследователь может изменять все факторы одновременно.

В ходе эксперимента регистрируются некоторые величины, характеризующие свойства изучаемого процесса или явления. Такие величины называются **функциями отклика**. С аналитической точки зрения эксперимент представляет собой выявление связи отклика y с рядом независимых переменных x_1, x_2, \dots, x_k . Эта связь в итоге выражается с помощью уравнения регрессии:

$$y = f(x_1, x_2, \dots, x_k).$$

Уравнение регрессии определяет некоторую поверхность в $(k+1)$ -мерном пространстве. Таким образом, целью эксперимента является получение такого количества точек в **пространстве факторов**, которое будет с достаточной степенью точности характеризовать **поверхность отклика**. Планирование экспери-

мента при этом заключается в выборе **оптимального количества точек** (опытов) и в размещении их в пространстве факторов таким образом, чтобы уравнение регрессии было определено с наибольшей точностью.

Суть одного из видов активного эксперимента, который называется **полным факторным экспериментом** (ПФЭ), заключается в том, что каждый фактор варьируется в эксперименте вместе со всеми другими факторами. При этом чтобы исследовать k факторов на m уровнях, требуется выполнить m^k опытов. Простейшие методы планирования эксперимента предполагают изменение каждого фактора на двух уровнях.

Планирование эксперимента можно предельно формализовать и упростить. Теория планирования эксперимента предполагает получение регрессионной модели в унифицированном виде. Эта модель может быть пригодна для любых откликов и факторов. Подобной моделью может служить алгебраический многочлен (отрезок ряда Тейлора):

$$y = b_0 + \sum_{i=1}^k b_i x_i + \sum_{i \neq j}^k b_{ij} x_i x_j + \sum_{i=1}^k b_{ii} x_i^2 + \dots$$

где b – коэффициенты регрессии, k – количество факторов, x_i представляет собой i -й фактор.

План экспериментов должен содержать, по крайней мере, такое же количество различных экспериментальных точек, какое количество коэффициентов модели требует определения.

Для построения плана эксперимента требуется определение границ варьирования переменных и основного (нулевого) уровня значений переменных. Для каждой переменной задается нулевой уровень x_{i0} , соответствующий наилучшим условиям построения модели. Для каждой переменной задается интервал варьирования Δx_i , который должен быть достаточно большим для статистической различимости уровней фактора на фоне ошибок его измерения. В рамках выбранной модели в виде алгебраического многочлена строится план

эксперимента путем варьирования каждого из факторов x_i на нескольких уровнях относительно нулевой точки, представляющей центр эксперимента.

Если выбранная модель включает в себя только **линейные** члены и **произведения факторов**, то для оценки коэффициентов модели используется план экспериментов с варьированием всех k факторов на двух уровнях. Такие планы ПФЭ называются планами типа 2^k , где $N = 2^k$ – число всех необходимых испытаний.

В качестве уровней варьирования выбираются значения факторов, симметрично расположенных относительно нулевой точки. Нижний и верхний уровни выбираются по соотношениям:

$$x_{in} = x_{i0} - \Delta x_i, \quad x_{iv} = x_{i0} + \Delta x_i.$$

Для упрощения записи плана проводится нормирование факторов, при этом нижнему уровню фактора будет соответствовать значение -1, а верхнему уровню значение +1. Преобразование проводится по формулам:

$$\bar{x}_i = (x_i - x_{i0}) / \Delta x_i.$$

Планы ПФЭ типа 2^2 и 2^3 представлены ниже в таблицах 3.5 и 3.6:

План ПФЭ типа 2^2

Таблица 3.5

Номер испытания	1	2	3	4
Фактор				
\bar{x}_1	-1	+1	-1	+1
\bar{x}_2	-1	-1	+1	+1

Полный факторный эксперимент дает возможность определить не только коэффициенты регрессии, соответствующие линейным членам, но и коэффициенты, соответствующие взаимодействиям факторов, которые проявляются при одновременном варьировании факторов. То есть действие одного фактора за-

висит от уровня на котором находятся другие факторы. Для оценки свободного члена и определения эффектов взаимодействий план ПФЭ типа 2^3 для модели

План ПФЭ типа 2^3

Таблица 3.6

Номер испытания	1	2	3	4	5	6	7	8
Фактор								
\bar{x}_1	-1	-1	-1	-1	+1	+1	+1	+1
\bar{x}_2	-1	-1	+1	+1	-1	-1	+1	+1
\bar{x}_3	-1	+1	-1	+1	-1	+1	-1	+1

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_{12}x_1x_2 + b_{13}x_1x_3 + b_{23}x_2x_3 + b_{123}x_1x_2x_3$$

расширяют путем добавления фиктивной переменной \bar{x}_0 и столбцов произведений типа x_1x_2 , x_1x_3 , ..., $x_1x_2x_3$ (таблица 3.7):

Таблица 3.7

Номер испытания	План ПФЭ								y_i
	\bar{x}_0		\bar{x}_2	\bar{x}_3	$\bar{x}_1\bar{x}_2$	$\bar{x}_1\bar{x}_3$	$\bar{x}_2\bar{x}_3$	$\bar{x}_1\bar{x}_2\bar{x}_3$	
1	+1	-1	-1	-1	+1	+1	+1	-1	-12
2	+1	-1	-1	+1	+1	-1	-1	+1	4
3	+1	-1	+1	-1	-1	+1	-1	+1	30
4	+1	-1	+1	+1	-1	-1	+1	-1	-6
5	+1	+1	-1	-1	-1	-1	+1	+1	2
6	+1	+1	-1	+1	-1	+1	-1	-1	-6
7	+1	+1	+1	-1	+1	-1	-1	-1	4
8	+1	+1	+1	+1	+1	+1	+1	+1	0

Таким образом, для определения коэффициентов представленной выше модели достаточно провести 8 экспериментов. Матрица планирования (табли-

ца 3.7) позволяет, пользуясь методом наименьших квадратов, вычислить коэффициенты регрессии независимо друг от друга по результатам всех испытаний.

Формулы вычислений по представленной матрице планирования (таблица 3.7) для модели

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_{12}x_1x_2 + b_{13}x_1x_3 + b_{23}x_2x_3 + b_{123}x_1x_2x_3$$

имеют вид:

$$b_0 = \sum_{t=1}^N x_0 y_t / N, \quad b_i = \sum_{t=1}^N x_{it} y_t / \sum_{t=1}^N x_{it}^2, \quad b_{ij} = \sum_{t=1}^N (x_i x_j)_t y_t / N,$$

$$b_{ijk} = \sum_{t=1}^N (x_i x_j x_k)_t y_t / N$$

Здесь индекс t означает номер испытания, N – общее количество испытаний, остальные обозначения соответствуют таблице 3.7. Результат обработки двухфакторного эксперимента представлен на рис. 3.45.

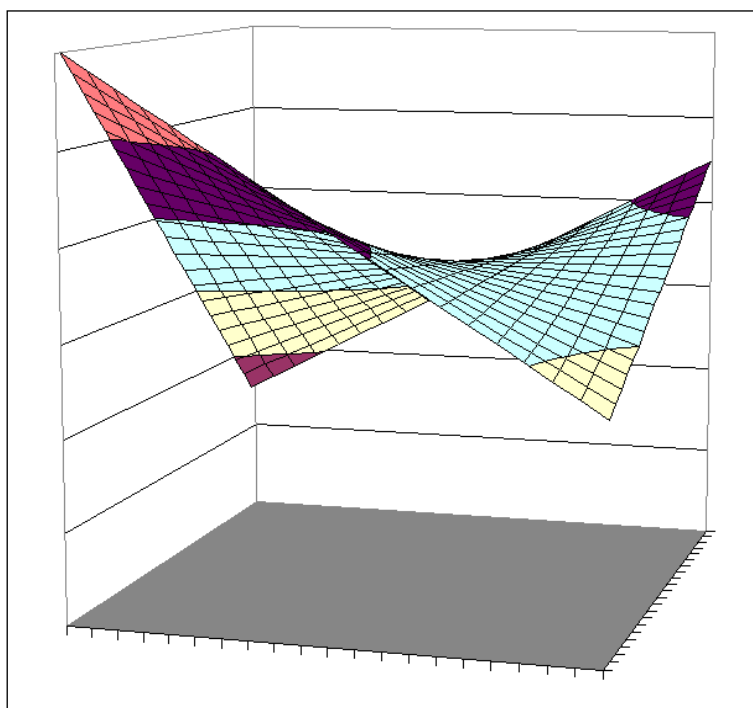


Рис. 3.45. Поверхность отклика – результат построения регрессионной модели

В данном разделе рассмотрен только один вид плана эксперимента. Однако существует достаточно большое количество других методов планирования, основанных на дисперсионном анализе и комбинаторике, целью которых является получение адекватных моделей, оптимальных в определенном смысле.

В заключение следует отметить, что применение активных методов планирования эксперимента позволяет построить модель процесса, когда информация о нем скудна, а сам процесс весьма сложен. Теория планирования эксперимента позволяет применять унифицированные методы к задачам самой разнообразной природы, проводить строгую статистическую обработку экспериментальных данных и оценку модели на основе статистических критериев значимости, повысить эффективность экспериментов за счет сокращения числа опытов до минимума.

3.9. ОПТИМИЗАЦИОННЫЕ МОДЕЛИ

На протяжении всей истории практической деятельности по созданию новых объектов человек вынужден искать ответы на вопросы: «Как создать изделие с наилучшими свойствами? «Как создать объект с заданными свойствами, но с наименьшими затратами?» При проектировании любого нового объекта всегда существует множество вариантов (альтернатив) выбора возможных конструктивных решений, вариантов компоновки, состава и способа соединения элементов системы и т.п. Выбрать наилучший вариант возможно формальным или творческим путем. Творческий путь основан на интуиции и опыте специалиста. Формальное решение проводится на основе **моделирования** и методов **оптимизации**.

Рассмотрим один из методов решения задачи выбора наилучшего (оптимального) решения, который проводится с помощью определенного **критерия** (целевой функции) – меры качества объекта. Суть критериального подхода состоит в следующем: формулируется критерий качества объекта (целевая функция). Критерий качества, по сути, является математической моделью этого объекта, т.к. отражает количественную характеристику определенного свойства объекта. Для каждой задачи целевая функция формулируется исходя из сути самой задачи. Например, в качестве целевой функции может быть выбрана формула вычисления прибыли от реализации плана, формула вычисления за-

трат на реализацию проекта, формула вычисления производительность технической системы и т.п. Так как значение критерия качества (целевой функции) дает количественную (численную) оценку каждого варианта решения, то это позволяет сравнивать различные варианты и выбрать наилучший вариант.

Таким образом, при решении задач оптимального выбора в рамках критериального подхода используются математические модели особого рода – **оптимизационные модели**, которые включают следующие компоненты:

1. Перечень независимых параметров объекта, которые определяют его свойства. Значения этих параметров можно варьировать при поиске оптимального решения.

2. Критерий качества объекта (целевая функция). Могут быть заданы следующие варианты поиска значения целевой функции: максимизация; минимизация; заданное значение. $F(x_1, x_2, \dots, x_n) \Rightarrow \text{Max}, (\text{Min})$.

3. Здесь x_1, x_2, \dots, x_n – независимые параметры системы, значения которых можно варьировать при поиске решения.

4. Ограничения, которые отражают взаимозависимости между параметрами объекта: $\varphi_i(x_1, x_2, \dots, x_n) \geq (= \leq) b_i$.

5. Граничные условия, определяющие допустимые пределы изменения параметров: $e_j \leq x_j \leq d_j$.

Решение, удовлетворяющее ограничениям и граничным условиям, называется допустимым. В данном случае решение – это набор значений параметров x_1, x_2, \dots, x_n . Среди допустимых решений необходимо найти такое, которое дает требуемое значение целевой функции (*max, min*). Это и есть оптимальное решение задачи. Получение оптимального решения позволяет обойтись без экспериментов с самим объектом. В ряде случаев целевая функция и ограничения задаются линейными соотношениями. Такие задачи называются задачами линейной оптимизации.

В электронных таблицах Excel с помощью надстройки «Поиск решения» можно решать задачи оптимизации. Первоначально необходимо включить

надстройку. Последовательность подключения надстройки показана на рис. 3.46–3.49. В главном меню Excel выберем пункт «**Файл**» затем пункт «**Параметры**» (рис. 3.46).

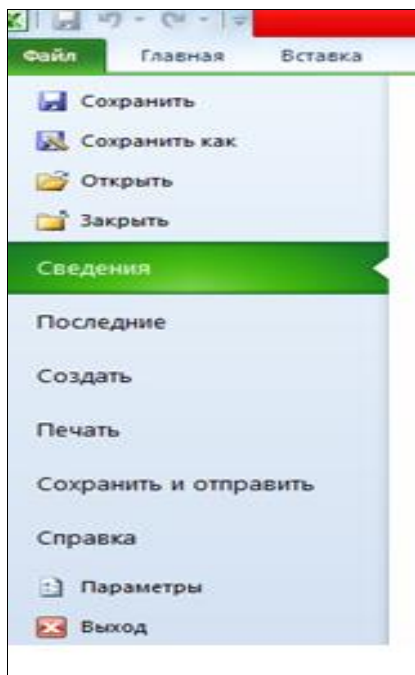


Рис. 3.46. Выбор пункта «Параметры»

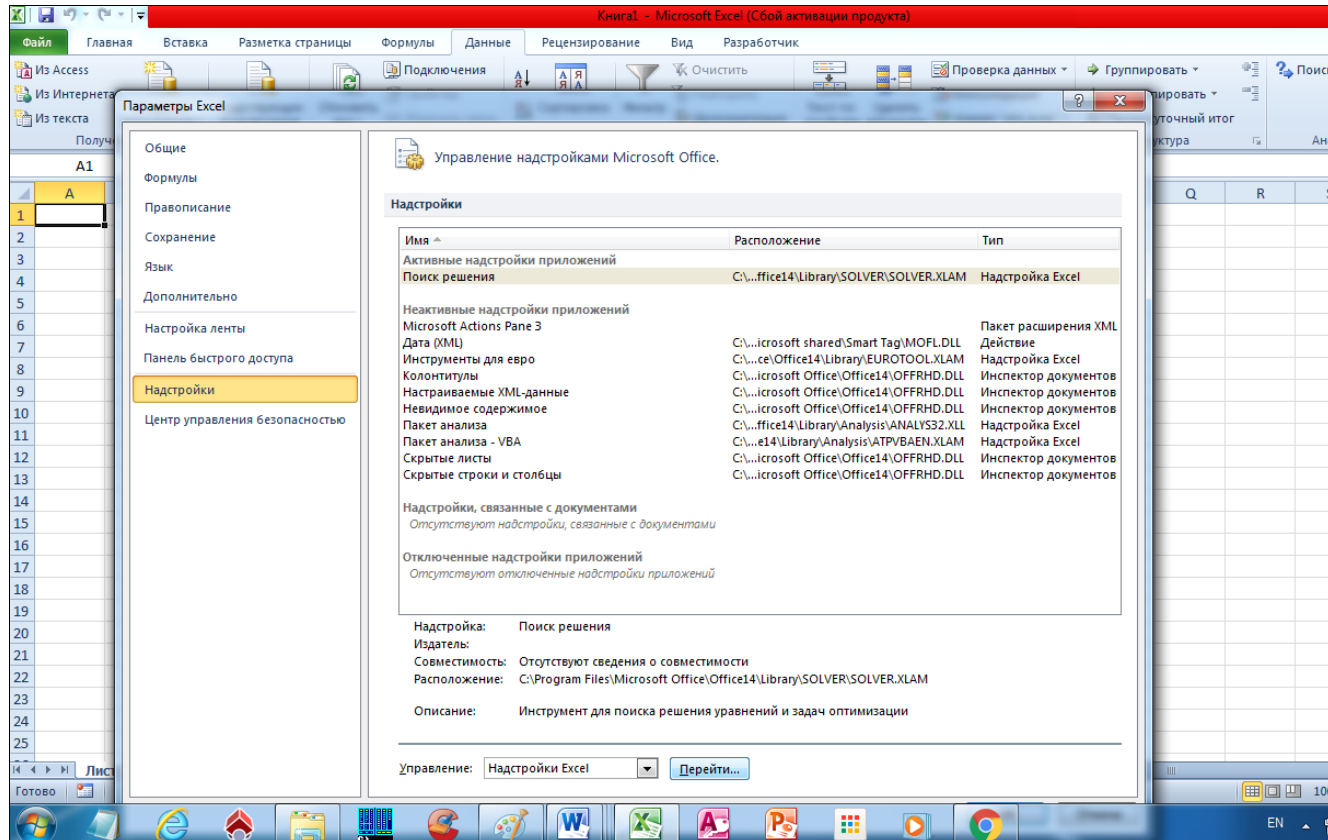


Рис. 3.47. Пункт меню «Надстройки» кнопка «Перейти»

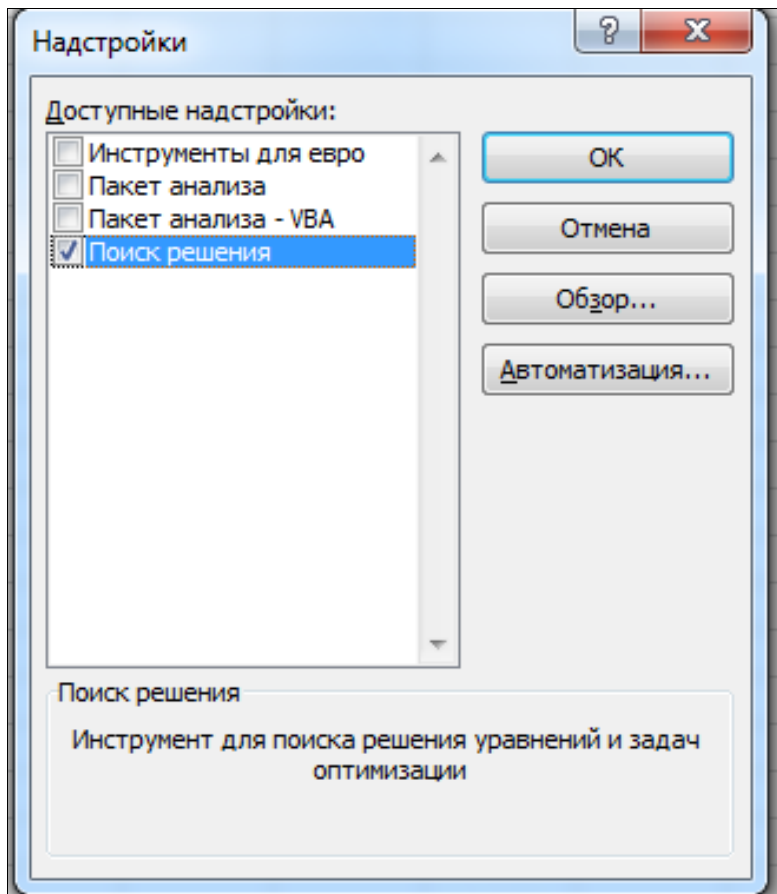


Рис. 3.48. Выбор надстройки «Поиск решения»

Отправной точкой при поиске оптимального решения является модель вычисления, созданная в рабочем листе в виде электронной таблицы.

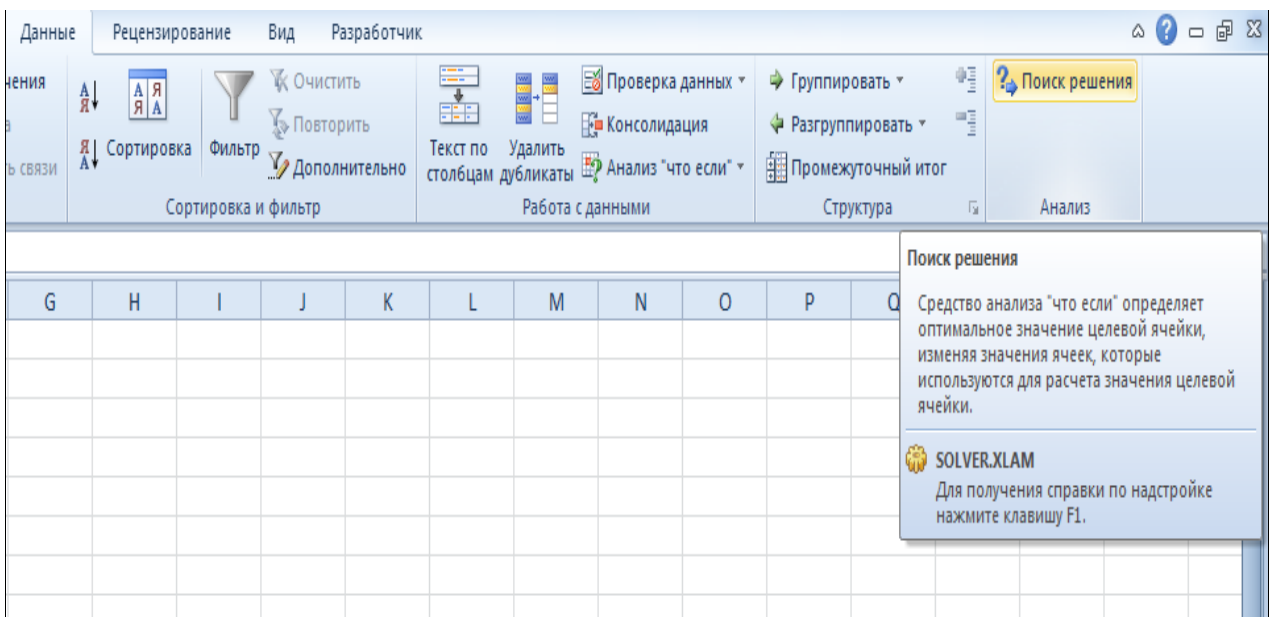


Рис. 3.49. Доступ к надстройке «Поиск решения», вкладка «Данные»

В надстройке «**Поиск решения**» необходимо задать следующие данные:

1. **Изменяемые ячейки** – это ячейки, которые будут содержать искомые значения параметров. При этих значениях целевая функция получит экстремальное (минимальное или максимальное) значение.

2. **Целевая ячейка** – это ячейка таблицы, значение в которой должно быть максимизировано или минимизировано. Она должна содержать формулу, которая прямо или косвенно ссылается на изменяемые ячейки. Эта формула представляет собой **целевую функцию**.

3. **Значения в изменяемых ячейках** будут автоматически последовательно изменяться до тех пор, пока не будет получено нужное значение в целевой ячейке. Эти ячейки, следовательно, прямо или косвенно должны влиять на значение целевой ячейки.

4. Можно задать **ограничения и граничные условия**.

Надстройка предоставляет возможность задать параметры, определяющие процесс поиска решения. После задания всех необходимых параметров можно запустить «**Поиск решения**».

Порядок решения задачи с помощью надстройки «**Поиск решения**» состоит в следующем:

1. Создать оптимизационную модель, которую следует представить в виде электронной таблицы.

2. Выделить в модели целевую ячейку. Соответствующая ссылка затем будет автоматически отображена в окне диалога «**Поиск решения**».

3. Выполнить переход **Данные/Поиск решения**. В открывшемся окне диалога в поле **Целевая ячейка** будет представлена ссылка на активную ячейку.

4. В группе «**Равной**», «**Максимум**», «**Минимум**» установить переключатель в нужное положение в зависимости от того, должно ли в результате оптимизации значение в целевой ячейке быть максимальным, минимальным или же равняться определенному заданному значению.

5. В поле «**Изменяя ячейки переменных**» укажите ссылки на изменяемые ячейки. Поместите курсор в это поле и выделите соответствующие ячейки.

6. При необходимости задайте нужные ограничения (рис. 3.50), используя описанную ниже процедуру определения ограничений.

7. Установите параметры решения задачи (кнопка «**Параметры**»).

8. Нажмите кнопку «**Найти решение**». Надстройка определит решение, выполняя последовательные вычисления и изменяя значения в соответствующей

щих ячейках. Если программа нашла решение, то это будет отображено в окне диалога «**Результаты поиска решения**» (рис. 3.51) и в модели.

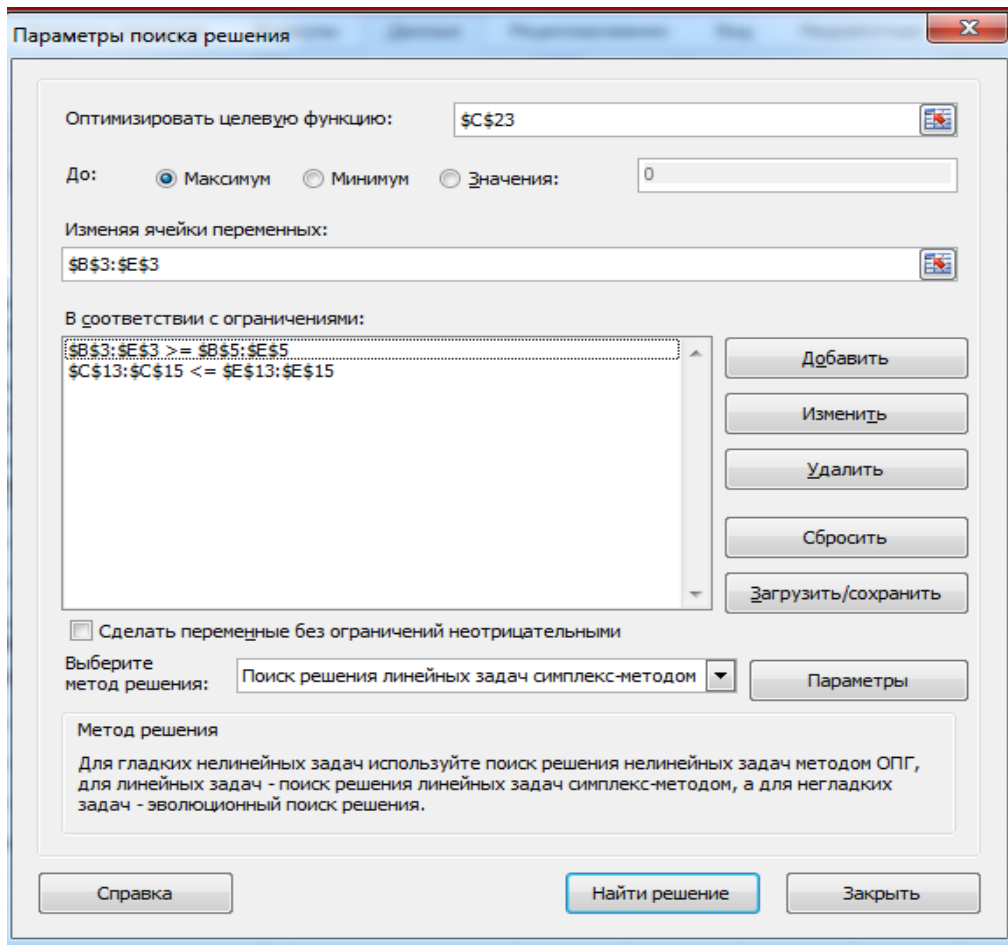


Рис. 3.50. Окно «**Поиск решения**»

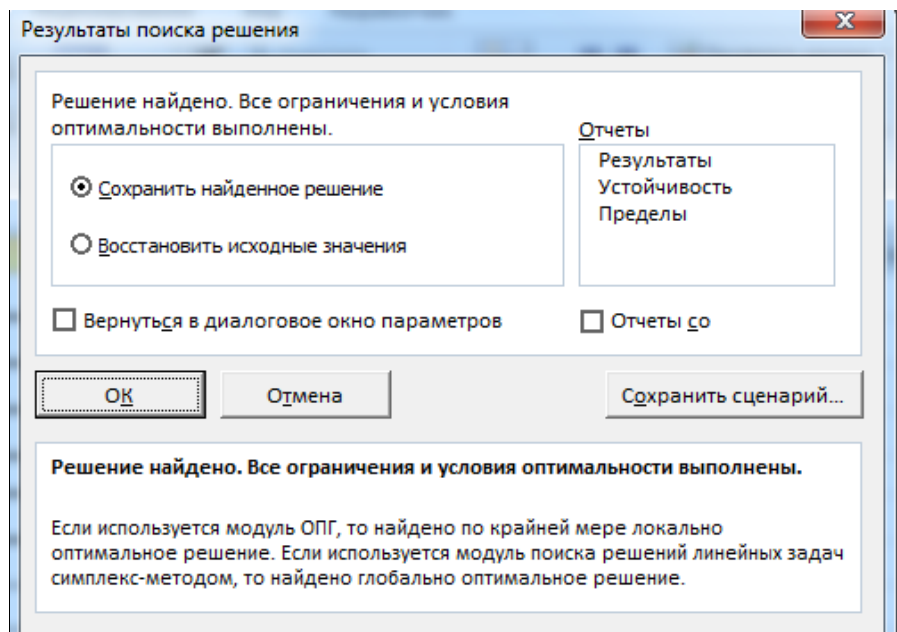


Рис. 3.51. Окно «**Результаты поиска решения**»

Если поиск решения не привел к нахождению оптимального результата, то в окне диалога будет отображено соответствующее сообщение. Ограничения – это условия, которые должны быть выполнены при поиске решения при решении задачи оптимизации. Ограничения определяют допустимые значения переменных, которые варьируются в процессе поиска решения.

1. В окне диалога «**Поиск решения**» в поле «**В соответствии с ограничениями**» нажмите кнопку «**Добавить**». На экране будет отображено окно диалога для определения ограничения (рис. 3.52).

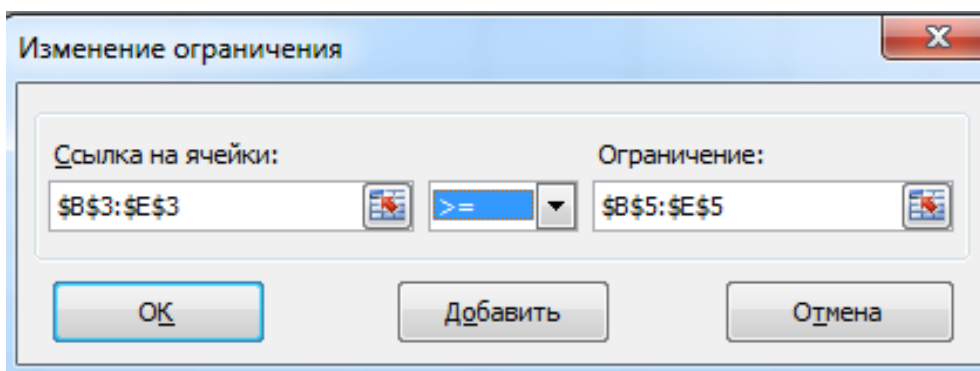


Рис. 3.52. Окно «**Добавление/изменение ограничений**»

2. В поле «**Ссылка на ячейку**» укажите ссылку на ячейку или диапазон ячеек, для которой должно действовать ограничение.

3. Выберите из списка нужный режим сравнения: =, <=, >=. Дополнительно в распоряжении имеется режим «**Цел**» (для переменной допустимы только целые значения) и «**Двоич.**» (допустимы только значения 0 или 1).

4. В поле «**Ограничение**» укажите верхнюю или нижнюю границу. Вы можете указать как число, так и ссылку на ячейку, диапазон ячеек, а также формулу.

5. Нажмите кнопку «**Добавить**». При этом окно не будет закрыто.

6. Аналогичным способом задайте и другие ограничения.

7. Необходимо выбрать метод решения: для линейной или нелинейной задачи.

8. Завершите определение последнего ограничения нажатием кнопки «**Найти решение**».

Рассмотрим пример решения задачи линейной оптимизации:

Имеется задача линейной оптимизации. Требуется найти значения переменных x_1, x_2, x_3, x_4 , при которых достигается максимум функции $f = 3x_1 + 2,5x_2 + 5x_3 + 4x_4$.

При ограничениях:

$$\begin{cases} 3x_1 + 5x_2 + 2x_3 + 4x_4 \leq 60 \\ 22x_1 + 14x_2 + 18x_3 + 30x_4 \leq 400 \\ 10x_1 + 14x_2 + 8x_3 + 16x_4 \leq 128 \\ x_1, x_2, x_3, x_4 \geq 0 \end{cases}$$

Решение задачи состоит из следующих этапов:

– Создание электронной таблицы, которая содержит все формулы и исходные данные (рис. 3.53).

– Ячейки **B5:E5** – изменяемые ячейки. В линейных задачах их начальные значения не влияют на результаты и алгоритм поиска.

– Диапазон ячеек **B11:E11** следует заполнить коэффициентами левых частей зависимостей для ограничений.

– Диапазон ячеек **E15:E17** надо заполнить значениями, составляющими правые части неравенств.

– В диапазон ячеек **B20:E20** следует записать коэффициенты целевой функции.

Ввод формул задачи:

• Для целевой функции в ячейку **C23** записываем формулу: **=СУММПРОИЗВ(B5:E5; B20:E20)**

• Для левых частей ограничений в ячейки **C15:C17** вводим формулы, начиная со знака равенства:

=СУММПРОИЗВ(B5:E5; B11:E11)

=СУММПРОИЗВ(B5:E5; B12:E12)

=СУММПРОИЗВ(B5:E5; B13:E13)

Процесс создания собственно оптимизационной модели с помощью надстройки «Поиск решения» описан выше.

Аналогичным образом могут быть решены задачи оптимального использования ресурсов, выбора оптимального плана перевозок и т.п. Большое число задач оптимизации связано с некоторыми структурами (графами). Например, задача поиска кратчайшего маршрута между заданными вершинами графа, задача оптимального плана перевозок по некоторой транспортной сети, поиск критического пути, задача «коммивояжера» и т.п.

	A	B	C	D	E
3		Переменные			
4		X1	X2	X3	X4
5		0	0	16	0
6		Нижняя граница переменных			
7		0	0	0	0
8		Верхняя граница переменных			
9					
10		Коэффициенты левой части ограничений			
11		3	5	2	4
12		22	14	18	30
13		10	14	8	16
14			Левая часть ограничения	Знак ограничения	Правая часть ограничения
15			32	<=	60
16			288	<=	400
17			128	<=	128
18					
19		Коэффициенты целевой функции			
20		3	2,5	5	4
21					
22		Целевая функция			
23		80			

Рис. 3.53. Электронная таблица для решения задачи оптимизации

На практике оценка альтернатив одним критерием затруднена. Любая система характеризуется многими положительными или отрицательными качествами. Например, производительность, мощность, энергоемкость, стоимость, трудоемкость изготовления и т.п. Обычно оценка решения делается с учетом нескольких частных критериев. Такая задача называется задачей **многокритериальной оптимизации**. Как правило, нет решения, соответствующего экстремальному значению всех альтернатив, так как частные критерии качества имеют конкурирующий характер. Улучшение одного критерия ведет к ухудшению другого и т.д. В этом случае при выборе решения требуется принятие определенного **компромисса**. Рассмотрим некоторые варианты методов принятия компромиссных решений.

1. Метод уступок. Этот прием основан на том, что с точки зрения решения конкретной задачи частные критерии неравнозначны между собой: одни более важны, чем другие. Пусть частные критерии упорядочены по степени их важности. Возьмем первый, самый важный критерий и найдем по этому критерию наилучшее решение. Затем определим уступку, т. е. величину, на которую мы согласны изменить полученное значение самого важного критерия, чтобы за счет этого попытаться улучшить значение следующего критерия и т.д.

2. Введение интегрального критерия, т.е. сведение многокритериальной задачи к однокритериальной. Важным элементом подобной оптимизации является назначение коэффициентов веса для каждого частного критерия. При построении таких критериев используются аддитивные и мультипликативные функции, которые определяют вклад каждого частного критерия в интегральный критерий. Однако полученное подобным способом решение может быть весьма чувствительным к изменению весовых коэффициентов.

3. Определение множества Парето. Этот метод позволяет существенно сократить поле поиска оптимального решения. Суть метода состоит в установлении предпочтения между вариантами решения (альтернативами). Предпочтение имеет место, если одна альтернатива по некоторым частным критериям не уступает другим альтернативам, а по остальным частным критериям их превосходит. Таким образом, выявив предпочтения между альтернативами, худшие альтернативы можно отсеять. Оставшиеся альтернативы образуют множество Парето. Затем для выбора единственной альтернативы можно применить, например, метод экспертных оценок.

Главный вывод, который мы должны сделать из представленного обзора методов состоит в том, что для задач многокритериальной оптимизации не существует единственного метода решения. Частные варианты приводят, как правило, к разным результатам.

В заключение данного раздела отметим, что оптимизация получила широкое применение в практике проектирования технических систем, а также в административной и управленческой деятельности. И это не удивительно, так как повышение эффективности любой целенаправленной деятельности – есте-

ственное стремление человека. Нахождение оптимальных вариантов особенно важно для оценки состояния современной техники, определения перспектив ее дальнейшего развития. Часто оказывается, что оптимальное решение всего на несколько процентов превосходит существующее. Но нередко оптимизация вскрывает значительные резервы улучшения качества систем.

При всей очевидной полезности оптимизации требуется осторожное обращение с подобным подходом. Основания для такого заключения получены в ходе практического решения задач оптимизации и состоят в следующем:

- Оптимальное решение часто оказывается весьма чувствительным к незначительным изменениям в условиях задачи.
- В понятии оптимальности важную роль играют ограничения. Не задав всех необходимых ограничений, мы можем одновременно с оптимизацией по основному критерию получить нежелательные сопутствующие эффекты.

Итак, отношение к оптимизации можно сформулировать следующим образом: практическое применение решений задач оптимизации прямо зависит от того, насколько хороша исходная модель. Для сложных систем моделирование является всегда приближительным, и использовать средства поиска оптимальных решений необходимо тем осторожнее, чем сложнее исследуемая система. Оптимизация элементов системы не даст оптимальности всей системе.

3.10. ГЕОМЕТРИЧЕСКИЕ КОМПЬЮТЕРНЫЕ МОДЕЛИ И САПР

Геометрическое моделирование изучает и применяет методы построения моделей, описывающих геометрические свойства различных объектов и систем. К геометрическим свойствам относятся пространственные отношения между реальными объектами их **форма и размеры, положение, ориентация** в пространстве. Геометрическое моделирование позволяет отражать эти свойства систем, не вдаваясь в подробности других свойств. Основной проблемой графики является отображение трехмерных объектов на плоской поверхности.

Сам термин «**компьютерная графическая модель**» говорит о том, что отражение свойств объекта-оригинала осуществляется средствами компьютерной графики, т.е. в наименовании модели отражен способ ее построения и реализации. Подобные модели, как наиболее наглядные, широко используются в практике моделирования.

Теоретической основой компьютерного геометрического моделирования являются методы построения подобных моделей, которые разрабатываются несколько веков:

1. Геометрия Евклида: построения с помощью циркуля и линейки.
2. Аналитическая геометрия: применение алгебраических методов в геометрии.
3. Начертательная геометрия, проективная геометрия.
4. Вычислительная геометрия (40–70-е гг. XX в.): разработка теории и прикладных методов геометрического моделирования на основе сплайн-функций, методов сплайн-аппроксимации трехмерных поверхностей.

Таким образом, современное геометрическое моделирование, основанное на компьютерных технологиях, базируется на аналитической и дифференциальной геометрии, вычислительной математике, вариационном исчислении и топологии, а также разрабатывает собственные методы. Описание геометрических свойств объектов происходит путем построения **математических моделей**, отражающих эти свойства, что позволяет проводить различные преобразования, редактировать и строить графические отображения этих объектов.

Исторически идеи создания какого-либо изделия воплощались за чертежным столом, проверялись при изготовлении материальной модели, проверялись в лабораториях и на испытаниях. Поэтому, традиционные курсы инженерной графики (до внедрения САПР) рассчитаны на обучение использованию чертежных инструментов, основам геометрических построений, различным видам проецирования (в основном ортогональной проекции), выполнению комплексных эскизов и чертежей, нанесению размеров и теории допусков и посадок.

Научное содержание графических курсов до появления САПР существенно не изменялось. Развитие аппаратных и программных графических средств,

включая твердотельное 3D-моделирование, изменило методологию, т.е. структуру, логическую организацию, методы и средства инженерной разработки проекта. Сегодня чертежные инструменты, такие как рейсшины, треугольники, готовальни и многое другое, в большинстве чертежных залов считаются устаревшими, хотя владение традиционными инструментами выполнения чертежей останется необходимым умением еще на долгие годы.

Эффективное использование всей мощи новых компьютерных инструментов требует от студентов широкого диапазона разнообразных навыков. Пространственное воображение становится все более важным, так как компьютерные трехмерные модели все чаще заменяют изготовление материальных моделей. Инженер должен представлять, какой вид будет иметь создаваемое им изделие, и должен уметь представить его как традиционным методом на бумаге, так и с помощью программ САПР.

Термин «САПР» (система автоматизированного проектирования) определяется как проектирование, осуществляемое взаимодействием человека и ЭВМ. Под проектированием здесь понимается процесс составления описания, необходимого для создания в заданных условиях еще не существующего объекта.

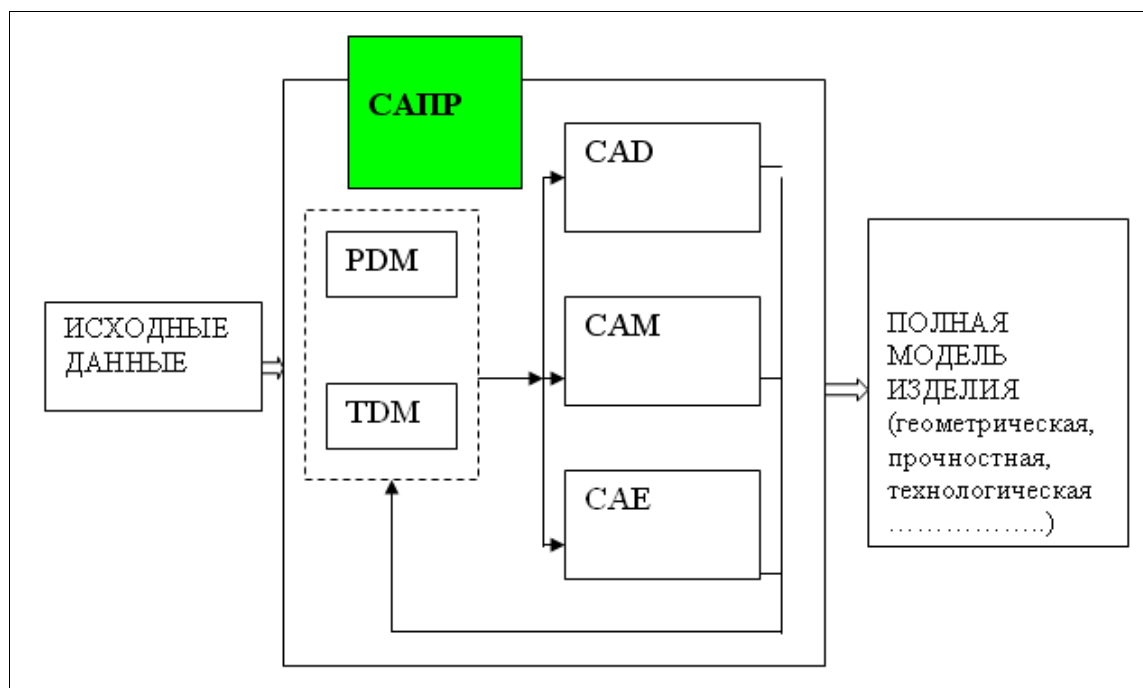


Рис. 3.54. Структура САПР

1. В САПР создается объектно-ориентированный цифровой документ.
2. В САПР создается математическая модель геометрии объекта, которая может быть передана для обработки в другие приложения: получение заготовок чертежей, построение наглядного изображения по чертежам, исследование свойств модели математическими методами, подготовка управляющих программ для станков с ЧПУ, подготовка конструкторской документации и др.
3. В САПР создается электронный цифровой документ (со всеми его преимуществами). Средства САПР предоставляют пользователю «виртуальные» электронные инструменты построения и моделирования, свойства которых значительно превосходят возможности традиционных инструментов черчения.
4. В системе САПР методами «визуального» построения создаются математические модели двумерных чертежей и трехмерных объектов, которые могут быть переданы в другие системы для последующей обработки: исследование, подготовка программ обработки на станках ЧПУ, подготовка конструкторской документации и т.п.

Отдельные модули САПР – CAD, CAM, CAE, TDM – решают весь спектр производственных задач (рис. 3.54).

CAD (Computer Aided Design) – модуль компьютерного геометрического (проектирования) моделирования.

CAM (Computer Aided Manufacturing) – модуль технологической подготовки производства.

CAE (Computer Aided Engineering) – модуль компьютерного инженерного анализа.

PDM (Product Data Management) – модуль, позволяющий управлять данными о продукции на протяжении всего жизненного цикла изделия при проектировании и подготовке производства.

TDM (Technical Data Management) – модуль управления базами данных, включая документооборот конструкторской и технологической документации.

Термин САПР связан с английской аббревиатурой CAD. Эта аббревиатура соответствует программным комплексам: CAD (Computer Aided Drawing) – это

черчение с помощью компьютера; CAD (Computer Aided Design) – проектирование с помощью компьютера. Аббревиатура CAM (Computer Aided Manufacturing) означает компьютерную технологическую подготовку производства.

Аббревиатура CADD означает черчение и проектирование с помощью компьютера. Сокращение CAE (Computer Aided Engineering) означает компьютерный инженерный анализ. Аббревиатуры PDM (Product Data Management) обозначает компьютерные системы управления данными о продукции на протяжении всего жизненного цикла изделия при проектировании и подготовке производства; TDM (Technical Data Management) – это управление документооборотом конструкторской и технологической документации. Система PLM+ (Product Lifecycle Management system) – управление жизненным циклом изделия. Итогом работы таких систем являются не только чертежи деталей, а в первую очередь программы, для изготовления этих деталей на станках с ЧПУ.

Исторически разработка и развитие CAD-систем определялась, в первую очередь, потребностями аэрокосмической, автомобильной и военной промышленности. Основные успехи CAD-систем связаны с развитием компьютерной графики, геометрического моделирования и компьютерного аппаратного обеспечения:

1. При построении модели создается цифровой документ. Средства САПР значительно превосходят возможности традиционных инструментов черчения.

2. Геометрическая модель объекта фактически представляет собой математическую модель формы объекта, которая может быть передана для обработки в другие приложения для получения чертежей заготовок, подготовки конструкторской документации, исследования свойств модели математическими методами, подготовки управляющих программ для станков с ЧПУ и др.

3. Возможна интеграция технологий трехмерного моделирования с другими методами в частности с методами численного моделирования процессов в проектируемом объекте.

В настоящее время известно множество CAD/CAM-систем различного уровня.

Система **CATIA** (Computer Aided Three-dimensional Interactive Application), система автоматизированного проектирования французской фирмы Dassault Systèmes. – одна из самых распространенных САПР высокого уровня. Это комплексная система автоматизированного проектирования (CAD), технологической подготовки производства (CAM) и инженерного анализа (CAE), включающая в себя передовой инструментарий трёхмерного моделирования, подсистемы программной имитации сложных технологических процессов, развитые средства анализа и единую базу данных текстовой и графической информации. Система позволяет эффективно решать все задачи технической подготовки производства: от предварительного проектирования до выпуска чертежей, спецификаций, монтажных схем и управляющих программ для станков с ЧПУ. В феврале 2008 г. Dassault Systèmes анонсировала новую версию системы – CATIA V6, которая поддерживает программы моделирования для всех инженерных дисциплин и коллективные бизнес-процессы на протяжении жизненного цикла изделия. Новая концепция получила название «PLM 2.0 на платформе V6». Суть концепции – трёхмерное моделирование и коллективная работа в реальном времени. Для связи между людьми, находящимися в разных точках мира, предусмотрены средства простого подключения к интернету.

AutoCAD – одна из первых САПР, ориентированная на решение задач машиностроения. В настоящее время это – одна из мощных CAD-систем для разработки конструкторской документации практически в любой сфере промышленного производства. Имеется множество приложений, интегрированных с AutoCAD и образующих САПР решения многочисленных и разнообразных задач проектирования. Начиная с версии 2010 в AutoCAD реализована поддержка параметрического черчения, то есть возможность накладывать на объект геометрические или размерные зависимости. Это гарантирует, что при внесении любых изменений в проект, определённые параметры и ранее установленные между объектами связи сохраняются.

T-FLEX CAD – профессиональная универсальная система параметрического двухмерного и трёхмерного геометрического моделирования в интересах, прежде всего, машиностроительного производства. Разработчик – россий-

ская фирма «Топ системы». В настоящее время выпущена новая отечественная САПР T-FLEX CAD 17 с приложениями.

Система позволяет полностью решить проблему подготовки технической документации: чертежей, схем, спецификаций и т.д. Обладая мощным параметрическим геометрическим ядром, позволяет существенно повысить скорость типового проектирования. Система позволяет создавать трёхмерные модели практически любого уровня сложности и чертежи на основе проекций трёхмерных моделей. Система имеет пакет анализа, который позволяет проводить прочностные расчеты, используя построенные трехмерные модели. Комплекс программ T-FLEX CAD/CAM/CAE/CAPP/PDM/CRM/PM/MDM/RM позволяет эффективно организовать работу на всех этапах жизненного цикла изделия. T-FLEX PLM+ (Product Lifecycle Management system) расширяет стандартные границы PLM-решений дополнительными возможностями по управлению всеми процессами, сопутствующими выпуску продукции.

САПР **ADEM** (Automated Design, Engineering & Manufacturing – Автоматизированное проектирование, расчет и изготовление) предназначена для автоматизации решения конструкторских и технологических задач производства. Система ADEM объединяет все известные методы геометрического моделирования, что позволяет сформировать конечный вид изделия и создать необходимую конструкторскую документацию. Начиная с 7-ой версии можно строить любые виды, разрезы и сечения без непосредственного обращения к 3D-модели. Изображение стрелки вида приводит к автоматическому созданию соответствующего вида или сечения. В настоящее время фирма-разработчик внедряют версию «Программный комплекс ADEM-VX 2020 в составе модулей: PDM, CAD, CAM, CAPP, SIM, GPP, NTR, I-RIS». Основным направлением деятельности группы компаний ADEM является автоматизация проектной конструкторско-технологической подготовки производства и управление инженерными данными на базе современных программно-технических комплексов.

Группа была основана в 1994 г. в результате объединения усилий коллективов компаний с целью разработки первой российской САПР для сквозного цикла проектирования-изготовления.

CAD-система **SolidWorks** позволяет строить самые разнообразные комплексы автоматизации проектирования, инженерного анализа и технологической подготовки производства. Для оптимизации состава комплексов и их функциональных возможностей в соответствии с решаемыми задачами в систему включен базовый модуль специальных API-функций для программирования прикладных задач. В результате во многих популярных прикладных системах появились средства прямого доступа к моделям SolidWorks. Более того, специально для SolidWorks было создано большое число модулей, работающих непосредственно в его среде.

Преимущества такого построения сквозного интегрированного решения очевидны. Возможность прямой передачи данных между различными приложениями позволяет создать гибкий программный комплекс, в котором могут быть задействованы лучшие в своем классе приложения. И сейчас пользователь SolidWorks может выбирать из нескольких сотен программных и аппаратных партнеров такие дополнительные модули, которые решат именно его задачи с минимальными финансовыми издержками. И тот факт, что отдельные модули созданы самыми разными производителями, для конечных пользователей остается незаметным. SOLIDWORKS 2020 добавляет новые возможности и усовершенствования, которые помогают максимально увеличить производительность ресурсов проектирования и производства, а также ускорить выпуск инновационных продуктов.

CAD-система «КОМПАС-3D» (разработчик российская фирм Ascon) предназначена для создания трёхмерных параметрических моделей и сборок и последующего создания их чертежей, содержащих все необходимые виды, разрезы и сечения, имеет определённые преимущества перед другими системами:

1. Система русскоязычна изначально, термины и определения полностью соответствуют терминологии отечественного конструирования.
2. Заложено выполнение всех требований стандарта ЕСКД.
3. Имеется очень широкий и одновременно практически необходимый набор функций редактирования изображений.

4. Система включает прикладные библиотеки (конструкторские библиотеки, справочники материалов и др.), ориентированные на отечественное производство.

5. Система постоянно модернизируется. В настоящее время внедряется профессиональная версия V.19, а версия V.12 LT предназначена для решения учебных задач и первоначального обучения на любом уровне.

Во всех современных системах трехмерного моделирования построение моделей производится с помощью операций **объединения, вычитания и пересечения**, выполняемые с **трехмерными объектами**. Эти объекты создает пользователь по определенным правилам при построении модели. Многократное применение этих операций суть построения трехмерной модели.

Для создания объемных элементов используются трехмерные операции, которые связаны с перемещением плоских фигур в пространстве. При этом ограничивается часть пространства, которая определяет форму элемента: перемещение прямоугольника приводит к формированию призмы; в результате поворота линии вокруг оси симметрии будет сформировано тело вращения; перемещение окружности вдоль направляющей кривой формирует объемный элемент – круглый стержень и т.п. (рис. 3.55–3.58).

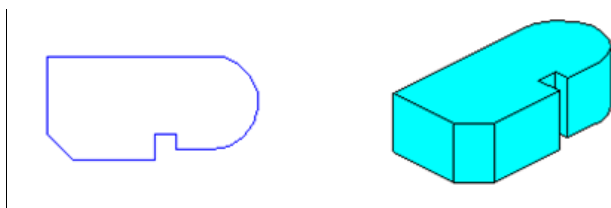


Рис. 3.55. Эскиз и элемент, образованный операцией «**Выдавливание**»

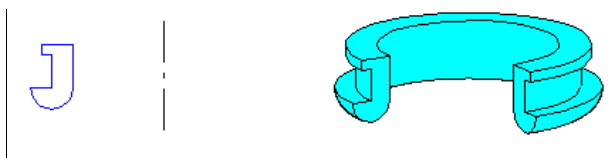


Рис. 3.56. Эскиз и элемент, образованный операцией «**Вращение**»

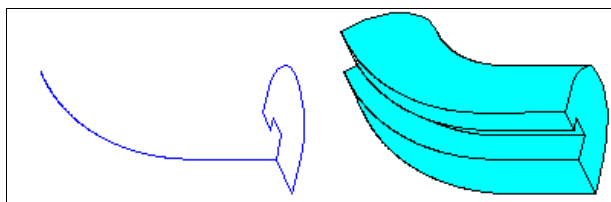


Рис. 3.57. Эскиз и элемент, образованный «Кинематической операцией»

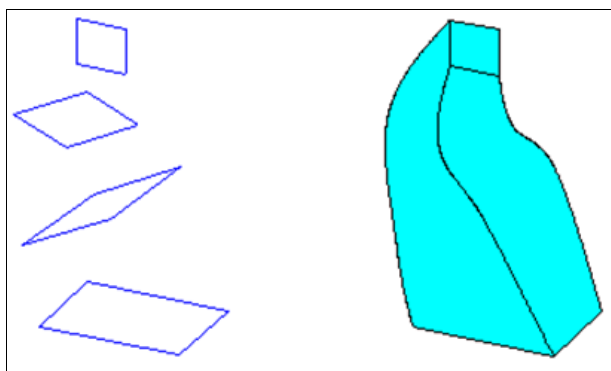


Рис. 3.58. Эскиз и элемент, образованный «Операцией по сечениям»

Плоская фигура, в результате перемещения которой образуется объемное тело, называется **эскизом**, а само перемещение – **операцией**. Эскиз может располагаться в одной из стандартных плоскостей проекций (фронтальная, горизонтальная, профильная), на плоской грани существующего тела или на плоскости, положение которой определено пользователем.

Операции, которые можно выполнять при построении модели:

1. Выдавливание – выдавливание в направлении, перпендикулярном плоскости эскиза.
2. Вращение – эскиза вокруг оси, лежащей в плоскости эскиза.
3. Кинематическая операция – перемещение эскиза вдоль направляющей.
4. Операции по сечениям – построение объемного элемента по несколькими эскизам, которые располагаются в нескольких параллельных плоскостях.

Имеется также операция «Вырезать», «Вырезать выдавливанием», «Вырезать вращением» и т.д.

Таким образом, процесс создания трехмерной модели заключается в многократном добавлении или вычитании дополнительных объемов, каждый

из которых представляет собой элемент, образованный при помощи операций над плоским эскизом. Объемные элементы, из которых состоит трехмерная модель, образуют в ней грани, ребра и вершины. Характеристики этих элементов представлены в табл. 3.8:

Характеристики элементов 3-D модели

Таблица 3.8

Элемент	Характеристика
Грань	Часть поверхности детали (часть плоскости)
Ребро	Прямая или кривая, разделяющая две смежные грани
Вершина	Точка на конце ребра
Тело детали	Замкнутая область пространства, ограниченная гранями детали

Для построения геометрических моделей используются идеализированные геометрические объекты: точка, линия, плоскость, поверхность и т.п., которые в отличие от реальных объектов обладают набором только наиболее существенных свойств. Ясно, что эти идеализированные объекты являются концептуальными моделями геометрии. Так, геометрическая точка отличается от реальной точки на чертеже тем, что имеет только координаты, но не имеет размеров, геометрическая линия не имеет толщины и цвета, геометрическая плоскость – не имеет толщины и т.д.

Из многих вариантов компьютерных графических моделей рассмотрим один, который имеет особую актуальность – отображение **трехмерных объектов на плоской поверхности**.

Еще задолго до появления компьютеров подобные задачи решались на основе законов перспективы и проекционного черчения. На базе точной геометрической интерпретации этих законов родилась целая техническая наука – начертательная геометрия. Однако и картина художника, и чертеж инженера дают статичное и одностороннее отображение объекта, с которым невозможно производить преобразования и различные манипуляции.

В практической деятельности человек имеет дело с трехмерными объектами, для описания которых вынужден был строить модели на плоской по-

верхности. Основной недостаток традиционного двумерного изображения (например, чертежа) состоит в том, что конструктор вынужден использовать проекции трехмерного объекта на базовые ортогональные плоскости. Ограничение двумерных графических моделей особо проявляются, когда поверхность детали имеет сложную криволинейную форму. Трудности восприятия по чертежу формы, взаимодействия и взаимного расположения различных деталей механизма в рамках двумерных графических моделей неустранимы.

Исторически возможности компьютерного графического моделирования претерпели существенное изменение, прежде чем они сравнялись с возможностями обычного рисунка или чертежа. Достаточно сказать, что первоначально отображение даже двумерных объектов (графиков, диаграмм и т.п.) производилось с помощью символов алфавита.

Следующий этап развития компьютерного графического моделирования связан с технологиями растровой и векторной графики. Естественно, что, с тех пор как компьютер «научился» рисовать на уровне пикселей, отображение плоских двумерных объектов не представляет особых трудностей.

Для кодирования изображения используется **векторная и растровая графика**. Растровый рисунок – **матрица цветных точек** (пикселей). Векторный рисунок – **совокупность графических объектов**.

Каждый графический объект имеет набор параметров и методов его построения. Перед выводом на экран объекта программа производит вычисления координат экранных точек в изображении объекта с учетом масштаба. Векторную графику называют параметрической или вычисляемой графикой. В основе векторной графики лежит математическое описание геометрических объектов. Каждая точка имеет координаты на координатной плоскости.

Математическая модель отрезка прямой представляется следующей зависимостью:

$$y = y_1 + \frac{y_2 - y_1}{x_2 - x_1} (x - x_1).$$

Параметры окружности: координаты центра, радиус, толщина линии, ее тип и цвет.

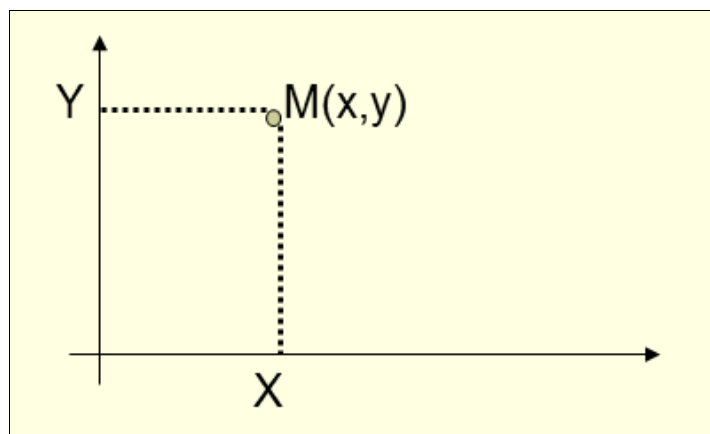


Рис. 3.59. Математическое описание точки

Особенности векторной графики

1. Масштабирование без потери качества изображения с проявлением мелких деталей.
2. Возможность проведения преобразований путем изменения параметров графических объектов.
3. Возможность программной обработки параметров.

Результат геометрического моделирования – математическая модель геометрии объекта. Математическая модель геометрии объекта позволяет:

- графически отобразить объект на экране;
- рассчитать его геометрические характеристики;
- рассчитать массовые, кинематические и другие характеристики;
- исследовать физические свойства;
- подготовить изготовление объекта.

Компьютерная графика – применение геометрической модели для получения изображения на экране. Изображение объекта – модель потока падающих и отраженных лучей света. Самый простой способ визуализации – отображение линиями. Отображается видимая часть объекта.

На современном уровне развития информационных технологий модели трехмерных объектов по принципу их построения можно разделить на три вида: **каркасные** модели; **поверхностные** модели; **твердотельные** (сплошные) модели.

Каркасная модель трехмерного геометрического объекта полностью описывается в терминах точек и линий (рис. 3.60). Элементами модели объекта

являются линии (ребра) и точки (вершины). Подобный вид моделирования давно известен в черчении – это аксонометрическая проекция.

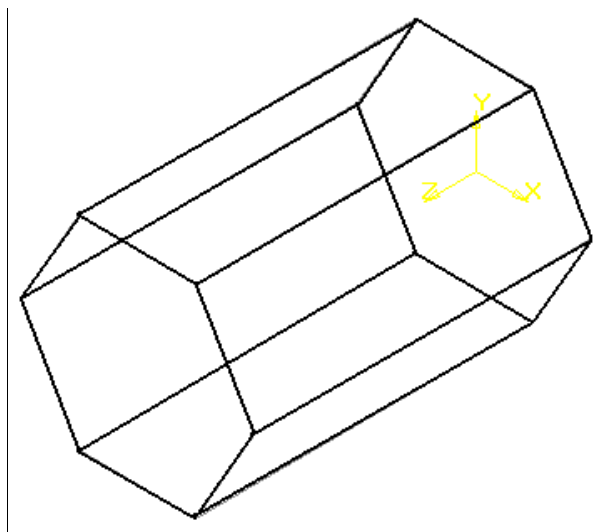


Рис. 3.60. Каркасная трехмерная модель

Главное ограничение каркасных моделей это недостаток информации о гранях. Они строятся наблюдателем чисто умозрительно. Вследствие чего невозможно однозначно распознать ориентацию и видимость граней трехмерного каркасного изображения. Этот эффект, которым иногда умышленно пользуются художники, обусловлен самой природой каркасной модели и может привести к непредсказуемым результатам. Для каркасной модели нельзя однозначно отличить видимые грани и ребра от невидимых. Еще сложнее в рамках каркасной модели обстоит дело с отображением криволинейных поверхностей. Подобная задача в общем виде в рамках каркасной технологии принципиально не имеет решения. Причем некоторые трехмерные объекты не имеют ребер, например сфера.

Каркасное моделирование – это моделирование трехмерных объектов самого низкого уровня. По своей сути оно является воспроизведением средствами компьютерной графики технологий трехмерного моделирования, которые были разработаны для черчения на бумаге.

Поверхностное моделирование трехмерных объектов связано с использованием точек, линий и поверхностей как графических примитивов. Результа-

том подобного моделирования является некоторая оболочка, которая описывает поверхность моделируемого объекта.

В поверхностном моделировании создаются и модифицируются поверхности, описывающие отдельные элементы объекта. Эти поверхности обрезают по линиям пересечения, сопрягают друг с другом и т.п.

Метод поверхностного моделирования наиболее эффективен для объектов, которые изготавливаются из листового материала и имеют сложные криволинейные поверхности, например: корпус автомобиля, корабля или самолета. В рамках этой технологии сложные поверхности образуются из элементарных геометрических поверхностей: поверхностей вращения, поверхностей, заданных аналитическими выражениями или поверхностей, образованных параллельным переносом линий (рис. 3.61).

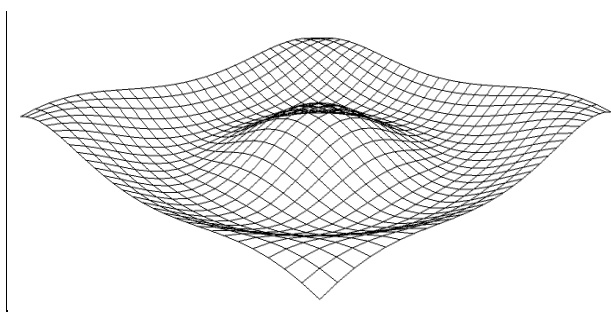


Рис. 3.61. Поверхностная трехмерная модель

Системы поверхностного моделирования представляют тело просто как совокупность поверхностей, соединенных друг с другом, которые ограничивают «пустой» объем. Применение данного метода моделирования, несмотря на ряд его достоинств, ограничено сложностью отображения внутренних полостей тела. Впрочем, при проектировании корпуса автомобиля или фюзеляжа самолета этого и не требуется.

Твердотельная модель. Основным элементом твердотельной модели является трехмерный объект как таковой. В твердотельном моделировании с самого начала построения модели работа производится с оболочками тел, а не с отдельными поверхностями. Оболочка полностью описывает поверхности объекта, отделяющие его внутренний объем от остальной части пространства.

В модели отражено, что одну часть пространства занимает объект, а другая часть находится вне объекта. Модель объекта может иметь одну или несколько оболочек: одна является внешней, другие ограничивают внутренние пустоты тела. Представление тела как совокупности оболочек, ограничивающих его объем, является наиболее общим подходом. Каждая оболочка строится из набора стыкующихся друг с другом поверхностей, содержащих полную информацию о своих границах и связях с соседями. Такое описание тел называется представлением с помощью границ. Оно дает возможность выполнять над телами множество операций, сохраняя при этом единый способ их «внутреннего» устройства. Такое представление тел позволяет моделировать объекты произвольной формы и сложности (рис. 3.62–3.63).

Например, трехмерную модель кубика можно спаять из проволоки (каркасная модель), склеить из бумаги (поверхностная модель) или вырезать из пенопласта (твердотельная модель).

Процесс построения твердотельной модели аналогичен процессу изготовлению материальной модели. Сначала создается некоторая простая заготовка. Далее заготовка изменяется требуемым образом путем присоединения или отсечения.

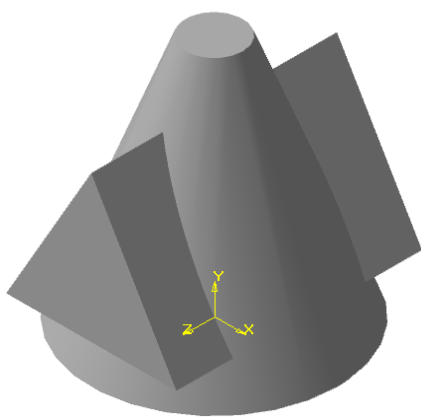


Рис. 3.62. Твердотельная модель.
Пересечение конуса и призмы.

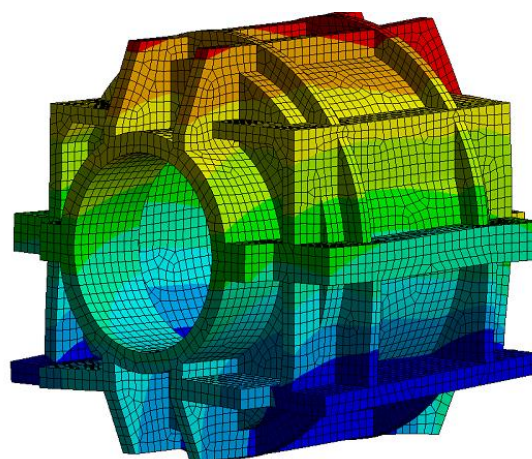


Рис. 3.63. Твердотельная модель
детали под нагрузкой

Технология твердотельного моделирования основана на конструировании модели из некоторого набора базовых трехмерных твердотельных про-

стейших объектов. Каждый такой объект строится пользователем на основе плоских эскизов с применением трехмерных операций. Данная технология актуальна, прежде всего, для машиностроения в частности для создания трехмерных моделей деталей и сборок.

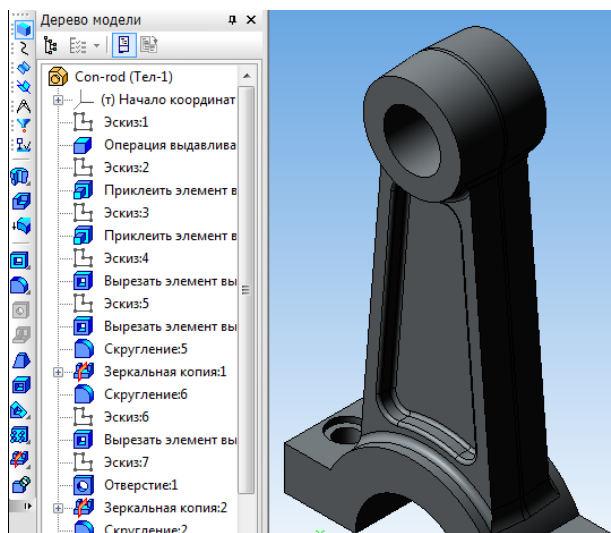


Рис. 3.64. «Дерево модели», 3D-модель в «КОМПАС»

Для редактирования геометрической модели тела необходима информация о последовательности построения модели. Поэтому в модель тела включено «дерево модели» (рис. 3.64).

В итоге результатом твердотельного геометрического моделирования некоего объекта является математическая модель его геометрии. Трехмерный объект определен многими изменяемыми параметрами. Вся эта информация содержится в математической модели его геометрических свойств, поэтому твердотельное моделирование называют **параметрическим**.

Компьютерное графическое изображение объекта является результатом использования его геометрической модели. Для того чтобы «увидеть» модель объекта нужно смоделировать **поток падающих и отраженных от его поверхностей лучей света**. Модель можно осветить с разных сторон светом различного цвета и интенсивности. Исходная информация для построения изображения на экране компьютера предоставляет геометрическая модель этого объекта.

Можно получить реалистическое изображение объекта, близкое к его фотографии. При этом поверхностям модели можно придать необходимый

цвет, зеркальность, прозрачность и другие оптические свойства. Итак, при графическом отображении твердотельная модель, полностью определяя форму объекта, обеспечивает автоматическое удаление невидимых линий, позволяет эффективно имитировать движение, управлять цветовой гаммой для получения тоновых эффектов.

С твердотельной моделью можно производить операции как с реальным объектом: перемещать, вращать, рассекать, приближать, или удалять, выполнять сборку из деталей и разборку на простейшие элементы, выполнять деформацию или отображать самые сложные движения трехмерного объекта в соответствии с законами механики и т.д.

Твердотельное моделирование позволяет не только отображать чисто геометрические свойства, оно позволяет, например, наделить модель свойством инерции (массой), отображать поведение и функции целостного объекта и происходящие в нем процессы (рис. 3.65). Информативность и наглядность компьютерного эксперимента возрастает на порядок, если его результаты представить не в виде традиционных числовых таблиц и графиков, а в виде наглядных образов.

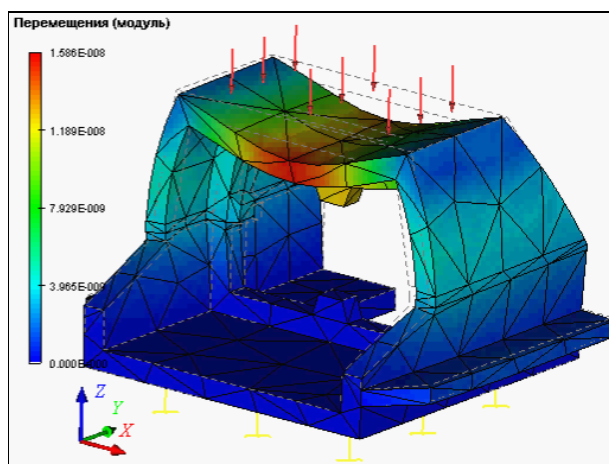


Рис. 3.65. Графическое представление напряжений и деформаций в объекте

Технологии 3D-моделирования позволяют полностью автоматизировать модельные исследования, например, температурных полей, полей напряжений и деформаций. Так, для объекта, представленного на рис. 3.65, после построения его трехмерной модели и задания внешних воздействий, расчет и

отображение состояния детали выполнены полностью автоматически. Теперь конструктору достаточно одного взгляда, чтобы определить наиболее нагруженные участки. Имея эту информацию можно принять необходимые меры для обеспечения надежности.

Численные результаты моделирования могут быть отображены в виде **виртуального движения** трехмерных твердотельных моделей объектов. Действительно, если твердотельная модель – это объект, то программным путем можно изменять его свойства и отображать поведение объекта, изменение его состояния и параметров на экране компьютера в виде 2D- или 3D-анимации (рис. 3.66).

Инструментальные системы моделирования MVS, RMD и т.п. позволяют использовать технологию виртуального движения. Аналогичные технологии использует современный кинематограф для создания сцен фантастических событий.

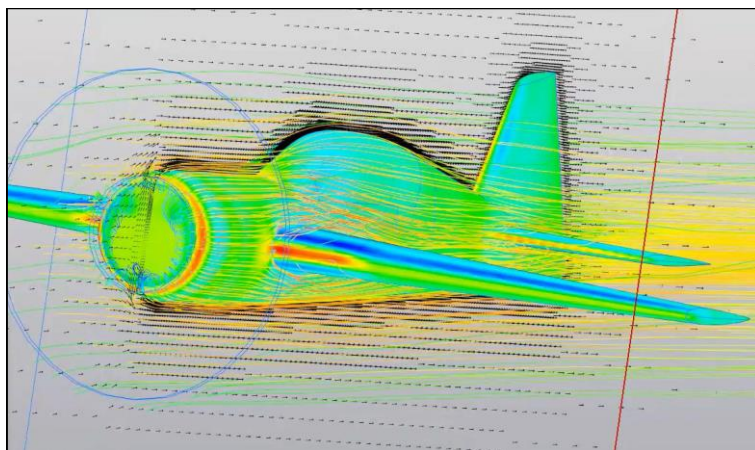


Рис. 3.66. Отображение обтекания самолета

Твердотельное моделирование позволяет создавать модели, состоящие из множества взаимодействующих объектов, так называемые **сборки** (рис. 3.67–3.68). Наглядность подобных моделей не нуждается в комментариях.

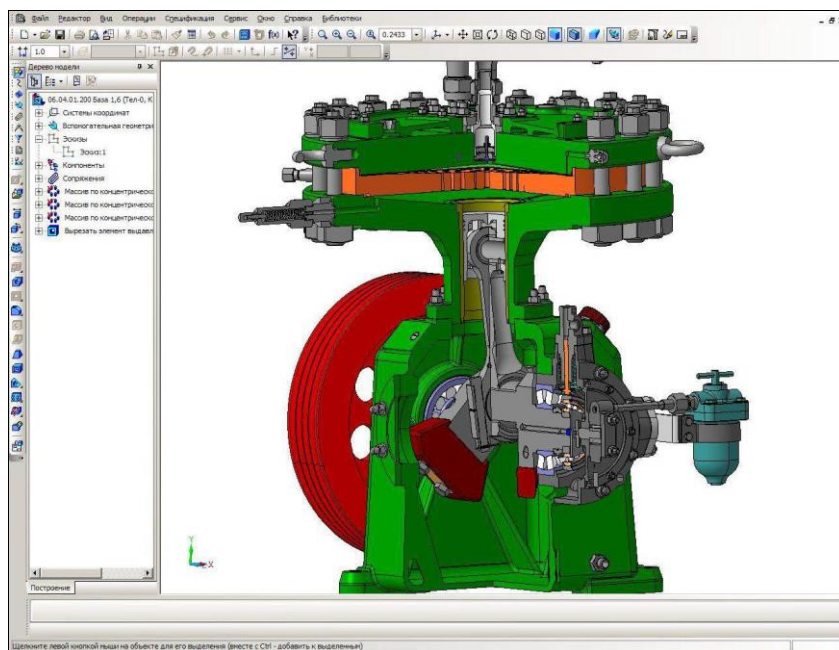


Рис. 3.67. 3D-сборка дизельного двигателя

Таким образом, современное компьютерное графическое моделирование объектов представляет собой синтез математического моделирования формы объекта и графического отображения геометрических свойств, а также позволяет отображать другие (не геометрические) свойства объекта.

В системах твердотельного трехмерного моделирования построение моделей производится с помощью последовательного создания трехмерных объектов и выполнения с ними типовых операций. Эти объекты создаются пользователем при построении модели. Создание трехмерных объектов связано с перемещением плоских фигур (эскизов) в пространстве.

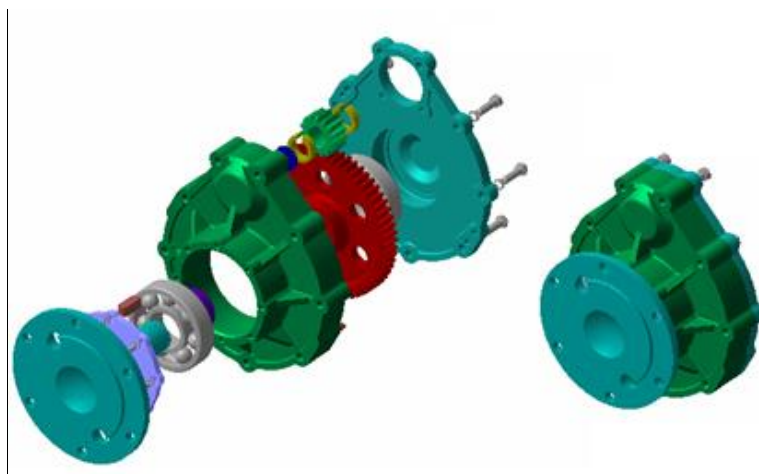


Рис. 3.68. Моделирование сборки редуктора средствами САПР «КОМПАС»

Эскиз может быть построен в одной из стандартных плоскостей (XY, ZX, ZY), на плоской грани существующего тела или на дополнительной плоскости, которая построена пользователем. Эскизы создаются средствами плоского черчения.

Основные операции, которые можно выполнять с эскизами при построении трехмерного объекта следующие: выдавливание перпендикулярно плоскости эскиза; вращение эскиза вокруг оси, лежащей в плоскости эскиза; перемещение эскиза вдоль направляющей; построение объемного элемента по нескольким эскизам, которые располагаются в нескольких параллельных плоскостях.

Создание трехмерной модели аналогично созданию реальной детали. Действительно, сначала выбирается (создается) заготовка, затем от нее отсекается лишнее или добавляются необходимые части. Процесс создания трехмерной модели заключается в многократном добавлении (приклеивании) или вычитании (вырезании) дополнительных трехмерных объектов, каждый из которых, образован при помощи вышеперечисленных операций над плоским эскизом. Порядок применения и параметры операций отражается в «**дереве модели**».

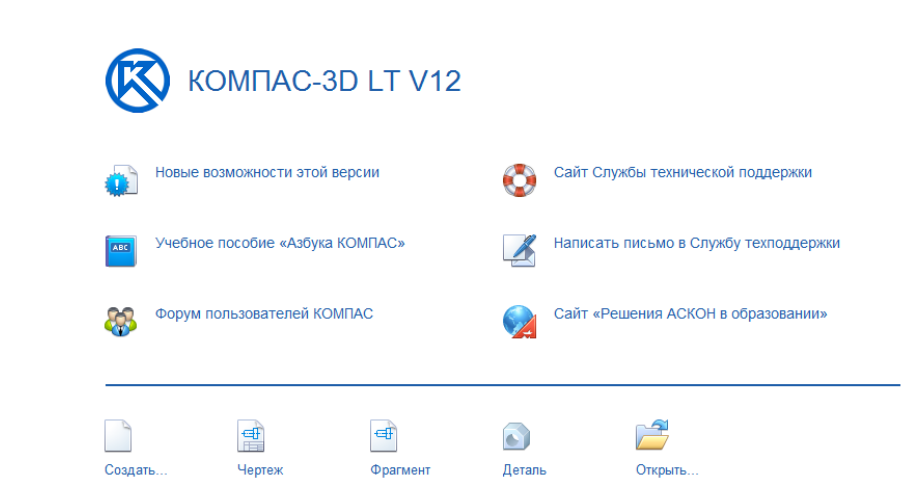


Рис. 3.69. Интерфейс системы КОМПАС после запуска

Построим трехмерную модель цилиндра, диаметр основания 70 мм, высотой 80 мм. Цилиндр (рис. 3.70) имеет продольное сквозное отверстие диаметром 30 мм.

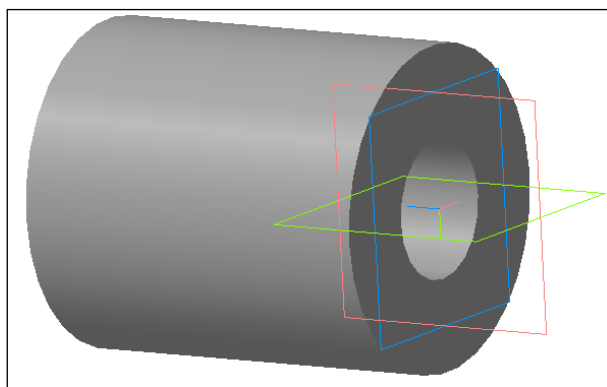


Рис. 3.70. Цилиндр с продольным отверстием

Первым шагом построения трехмерной модели является запуск подсистемы трехмерного моделирования, который производится при нажатии кнопки «Создать... Деталь» на стартовой странице «КОМПАС» (рис. 3.69).

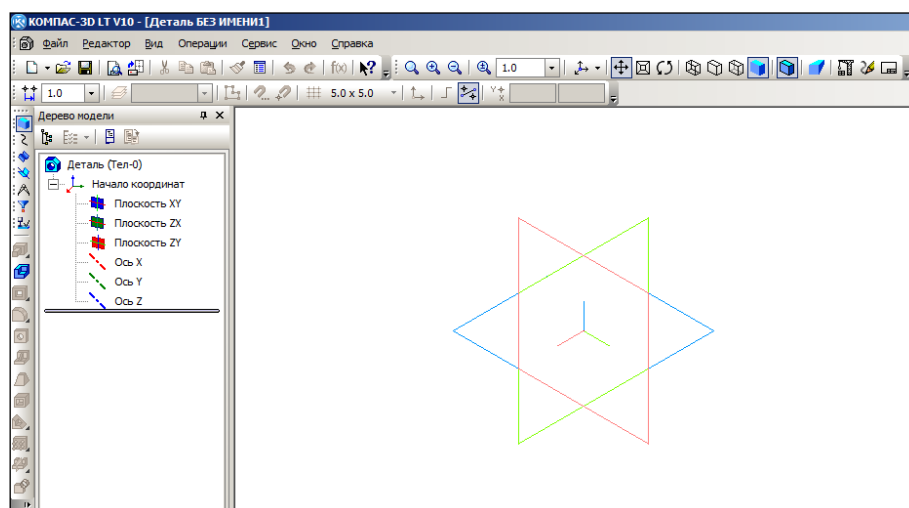


Рис. 3.71. Окно построения трехмерной модели (детали)

В результате откроется окно построения трехмерной модели (рис. 3.71). Первоначально познакомимся с функциональными кнопками подсистемы трехмерного моделирования.

Первоначально, в «**дереве модели**» (рис. 3.71) имеются система координат и три системные плоскости. Теперь следует выбрать одну из системных плоскостей (XY, ZX, ZY, рис. 3.72–3.74) для построения первого эскиза. Выберем плоскость XY (рис. 3.72, рис. 3.75).

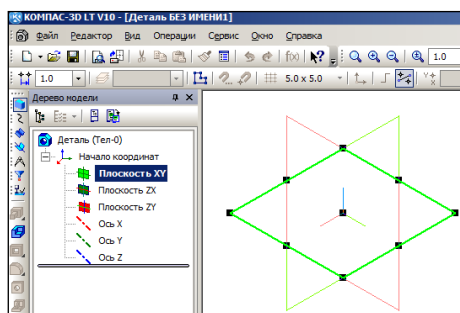


Рис.3.72. Выделена плоскость XY

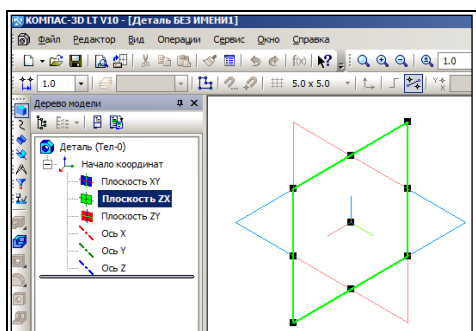


Рис.3.73. Выделена плоскость ZX

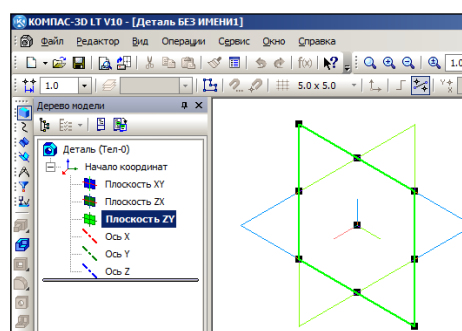


Рис.3.74. Выделена плоскость ZY

Далее следует задать ориентацию выделенной плоскости (рис. 3.75). Выберем вариант «**Нормально к...**». Новое положение выделенной плоскости представлено на рис. 3.76.

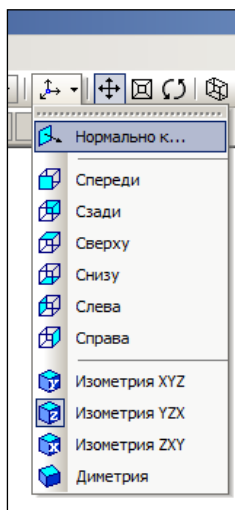


Рис. 3.75. Выбор ориентации плоскости эскиза

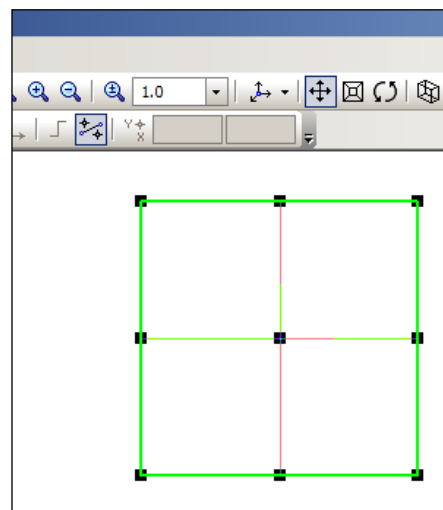


Рис. 3.76. Положение плоскости после выполнения команды «**Нормально к...**»

Включим режим редактирования эскиза (рис. 3.77), при этом установится режим плоского черчения (рис. 3. 78).

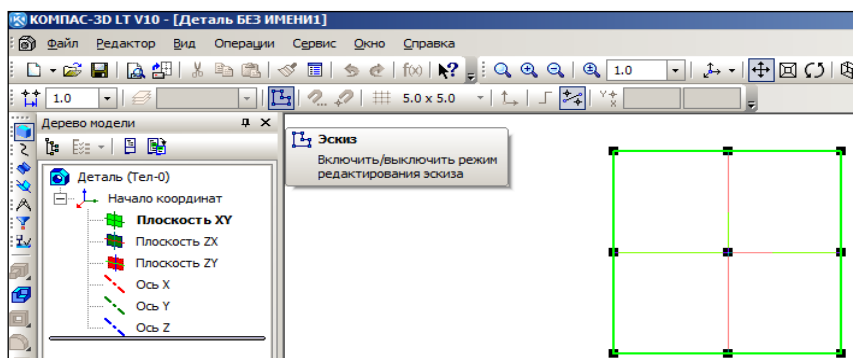


Рис. 3.77. Выполнение команды «Включить/выключить режим редактирования эскиза»

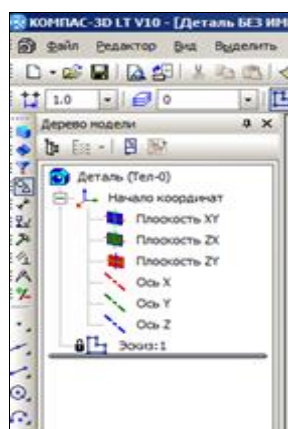


Рис. 3.78. Режим плоского черчения

Включим режим отображения сетки (рис. 3.79). Далее для точного построения эскиза окружности установим режим привязок по сетке (рис. 3.80). Для построения окружности выберем способ построения (рис. 3.81). Центр окружности привяжем к началу координат. Эскиз представлен на рис. 3.82.

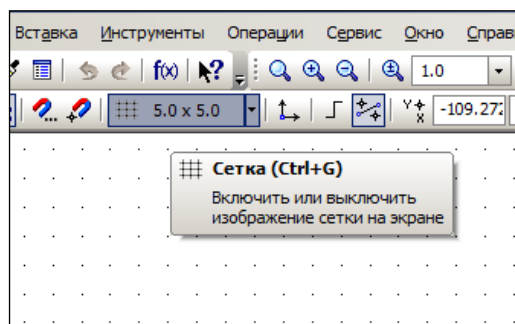


Рис. 3.79. Включение отображения сетки

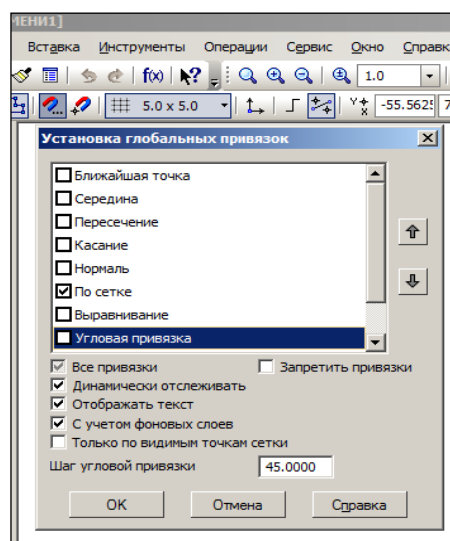


Рис. 3.80. Установка привязок

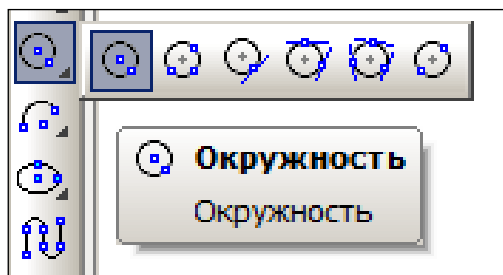


Рис. 3.81. Выбор способа построения окружности

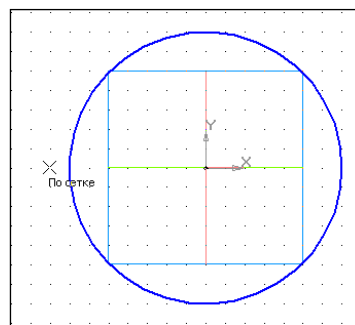


Рис. 3.82. Первый эскиз – окружность

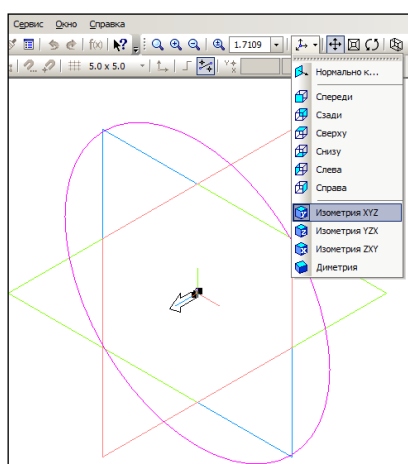


Рис. 3.83. Выбор ориентации эскиза «Изометрия XYZ»

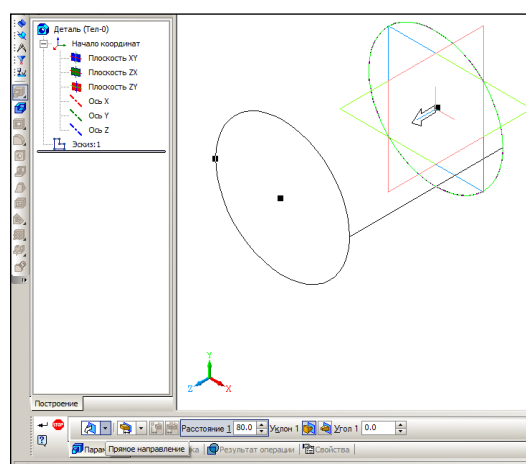




Рис. 3.84. Выполнение операции «Выдавливание»

Далее закончим редактирование эскиза и зададим его ориентацию «Изометрия XYZ» (рис 3.83). Выполним операцию выдавливания . Расстояние выдавливания 80 мм. На рис. 3.84 внизу рисунка представлены параметры операции и ее фантом. Выполнение операции происходит при нажатии кнопки **Создать** .

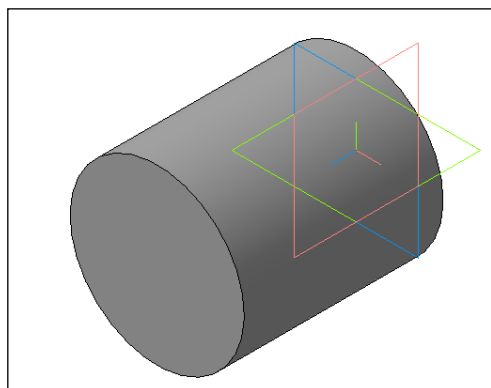


Рис. 3.85. Результат выполнения операции Выдавливание

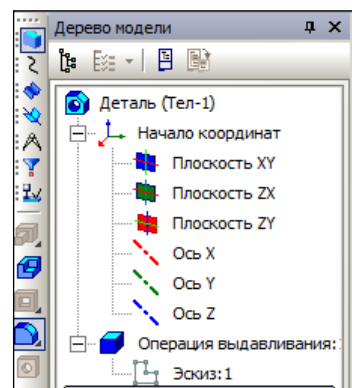


Рис. 3.86. Дерево модели

Результат показан на рис. 3.85. Выполнение операции приведет к изменению дерева модели (рис. 3.86). Таким образом, заготовка для будущей детали построена. Продолжим создание детали. Теперь необходимо вырезать продольное отверстие диаметром 30 мм.

Выделим плоскость XY. Выберем ее ориентацию «Нормально к...». Включим режим редактирования эскиза. Сетка и привязки по сетке уже установлены. Создадим новый эскиз (рис. 3.87); – окружность диаметром 30 мм.

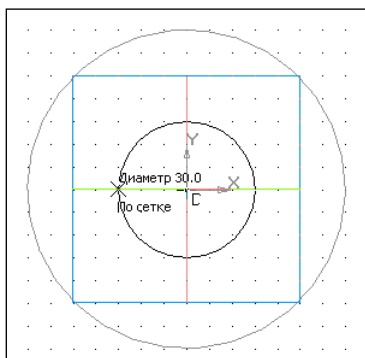


Рис. 3.87. Новый эскиз – окружность диаметром 30 мм

Завершим редактирование эскиза и зададим для него ориентацию «Изометрия XYZ». Выполним операцию «Вырезать выдавливанием» (рис.3.88–3.89). Обращаем внимание: необходимо правильно выбрать направление вырезания, иначе отверстие не будет вырезано. Расстояние вырезания следует задать несколько больше, чем длина детали. После выполнения операции деталь примет вид, представленный на рис. 3.90.

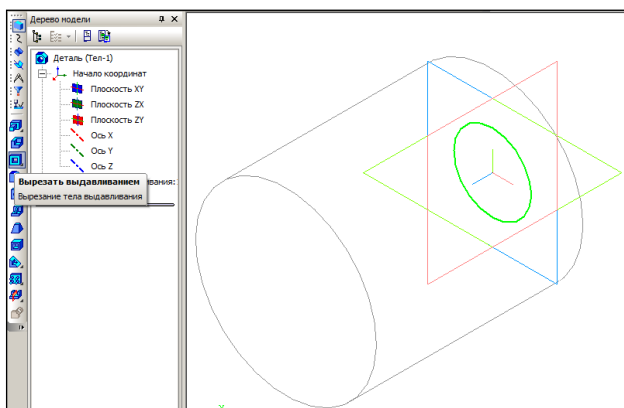


Рис. 3.88. Операция «Вырезать выдавливанием»

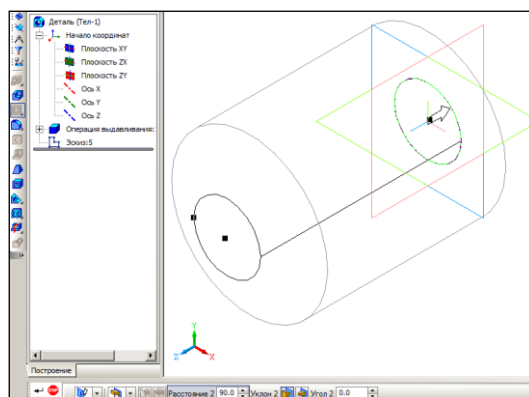


Рис. 3.89. Выполнение операции «Вырезать выдавливанием»

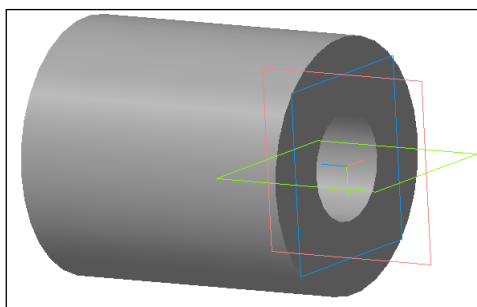


Рис. 3.90. Результат выполнения операции «Вырезать выдавливанием»

САПР на основе системы «КОМПАС» в настоящее время стремительно развивается и успешно конкурирует в России с другими системами. Получает применение в различных отраслях (рис. 3.91).

В заключение можно с уверенностью сказать, что на современном этапе компьютерное геометрическое и графическое 3D-моделирование развивается самыми быстрыми темпами.



Рис. 3.91. Сферы применения САПР «КОМПАС»

3.11. МОДЕЛИРОВАНИЕ СИСТЕМ УПРАВЛЕНИЯ

Потребность в автоматическом регулировании и управлении возникла и стала весьма актуальной с ростом сложности техники и расширением применения технических систем в производстве. Возникла необходимость управления процессами, которые подвержены действию помех и внешних воздействий, не только нарушающих точность функционирования, но и работоспособность системы в целом. Стало очевидно, что человек (оператор) не в состоянии обеспечить требуемую точность и оперативность регулирования режима работы технической системы. Например, необходимость регулирования актуальна для таких точных механических систем, как часы. Для них требуется обеспечить точность хода в условиях постоянного действия небольших помех.

Развитие устройств регулирования и управления началось в эпоху промышленной революции. Одним из первых промышленных регуляторов стал поплавковый регулятор питания котла паровой машины И.И. Ползунова и центробежный регулятор скорости вращения вала паровой машины Дж. Уатта. Паровая машина стала первым объектом применения регулирования, так как она не обладает свойством устойчивой работы. Известно множество случаев, когда оператор (человек) паровой машины был не способен выполнять функции регулирования, а наоборот, «раскачивал» колебания ее параметров. Именно потребности регулирования паровой машины стимулировали развитие теоретических исследований в данной области.

Фундаментальное значение для практики и теории регулирования имели работы Дж. Максвелла «О регуляторах», И.А. Вышнеградского «Об общей теории регуляторов» и «О регуляторах прямого действия». В данных работах реализован системный подход к проблеме. Регулятор и машина рассматривались как единая динамическая система. В этих работах заложен общий методологический подход к исследованию разнообразных по принципу действия систем. В работах И.А. Вышнеградского сформулированы основы теории устойчивости и установлены важные закономерности регулирования с использованием обратной связи.

В начале XX в., теория автоматического регулирования сформировалась как самостоятельная научная дисциплина с рядом прикладных разделов, которые связаны с конкретными техническими системами. Разрабатываются новые математические методы анализа линейных и нелинейных систем регулирования, методы исследования устойчивости и чувствительности, новые принципы управления: регулирование по возмущению, теория компенсации возмущения и инвариантности, принципы экстремального управления, теория оптимального управления и т.д.

Существенный вклад в развитие теории автоматического управления и регулирования внесли российские ученые Н.М. Крылов, Н.Н. Боголюбов, А.В. Михайлов, А.А. Ляпунов, А.А. Андронов, Л.С. Понтрягин, А.А. Крассовский и другие.

Значение теории автоматического управления актуально не только для технических систем, управляемые процессы имеют место в самых различных системах, например: в экономических, технологических, социальных, человеко-машинных и т.п. В настоящее время теория автоматического управления является научно-технической дисциплиной, которая изучает общие принципы управления разнообразными техническими системами на основе построения их математических моделей, разрабатывает собственные методы анализа и синтеза систем управления, методы анализа устойчивости, методы оценки свойств проектируемых систем и качества регулирования.

Итак, **управление** – это специальное воздействие на объект управления с целью обеспечить его оптимальное состояние или поведение. **Регулирование** – это частный случай управления: обеспечение определенного состояния объекта регулирования. Автоматическое управление и регулирование осуществляются без участия оператора (человека). При возникновении отклонений от требуемого режима функционирования объекта система регулирования (управления) должна воздействовать на объект так, чтобы устранить отклонения.

Различают внешнее и внутреннее управление в системе. Внутренняя функция системы – самоуправление. Внешнее управление обеспечивает необходимое функционирование системы.

На рис. 3.92 представлено устройство, предназначенное для поддержания постоянного числа оборотов вала паровой машины при изменении нагрузки.

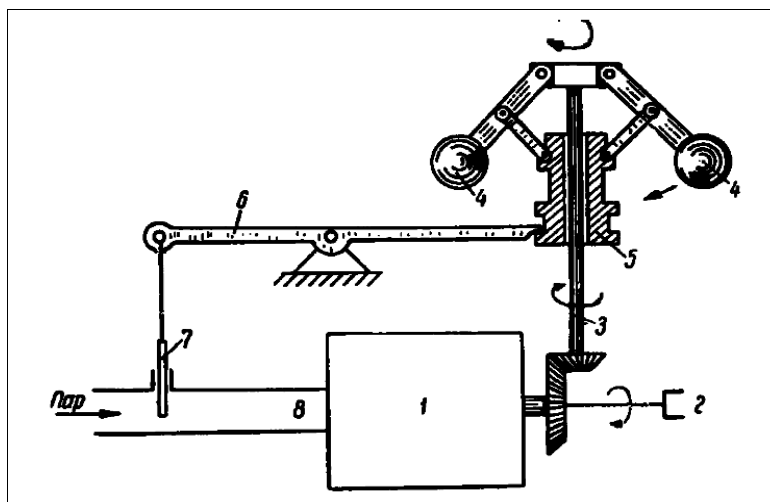


Рис. 3.92. Регулятор Уатта

Паровая машина предназначалась для привода ткацких станков, для работы которых требовалась постоянная скорость вращения вала станка. Потребность в автоматизации возникла вследствие того, что паровая машина требовала регулирования, а оператор (человек) не мог справиться с задачей регулирования работы паровой машины вследствие своих чисто физиологических возможностей.

Эта техническая система дала толчок для развития таких наук, как механика, термодинамика, теория автоматического регулирования и управления.

Задачи теории автоматического управления:

1. Анализ – исследование свойств системы управления.
2. Синтез – создание системы управления, которая обеспечит необходимое поведение объекта управления.

Следующим шагом было развитие кибернетики, основы которой Норберт Винер (1894–1964 гг.) изложил в книге «Кибернетика, или Управление и связь в животном и машине», 1948 г. Во время Второй мировой войны, он работал над математическим аппаратом для систем наведения зенитного огня и управление американскими силами противовоздушной обороны (рис. 3.93).



Рис. 3.93. Профессор Норбет Винер

Существуют следующие виды систем управления:

- **Системы стабилизации.** На начальном этапе развития техники системы стабилизации представляли единственный вид систем управления. Задачей данной системы является поддержание постоянства управляемого параметра.
- **Системы программного управления.** В этом случае алгоритм функционирования системы задан. В подобных системах существует датчик, который вырабатывает задающие воздействия по определенному алгоритму. Программное управление может быть организовано по любой схеме из представленных на рис. 3.94–3.96. Например, системы подобного рода применяются при программном управлении металлорежущими станками, где заложен алгоритм функционирования системы, который изменяет управляемую величину в соответствии с заранее заданной функцией времени.
- **Следящие системы.** Алгоритм функционирования в таких системах заранее неизвестен. Регулируемый параметр в таких системах должен воспроизводить изменение некоторого внешнего фактора, например движения цели.

Следящая система управления может быть организована также по любой из схем по рис. 3.94–3.96, если в ней будет присутствовать датчик слежения за изменением внешнего фактора.

- **Системы оптимального управления.** Определяют и поддерживают управляющие воздействия, соответствующие экстремальному значению управляемой величины.

- **Адаптивные системы управления.** Способны автоматически приспосабливаться к изменению внешних условий и свойств объекта управления.

Управляющие сигналы к объекту управления могут поступать непрерывно. Такие системы управления называются системами непрерывного действия. Примером является система управления паровой машиной на основе регулятора Уатта. Примером систем управления дискретного действия является система термостатирования, в которой обогрев или охлаждение включаются и выключаются в момент достижения определенных температур в термостате.

Виды управления представлены ниже:

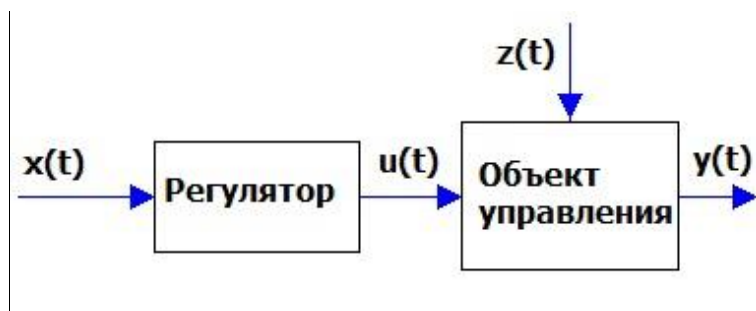


Рис. 3.94. Управление без обратной связи

При управлении без обратной связи (рис. 3.94) сигналы управления генерируются по определенной программе без учета состояния объекта управления. Примером системы с жестким управлением является светофор, который регулирует движения на перекрестке (объект управления – перекресток).

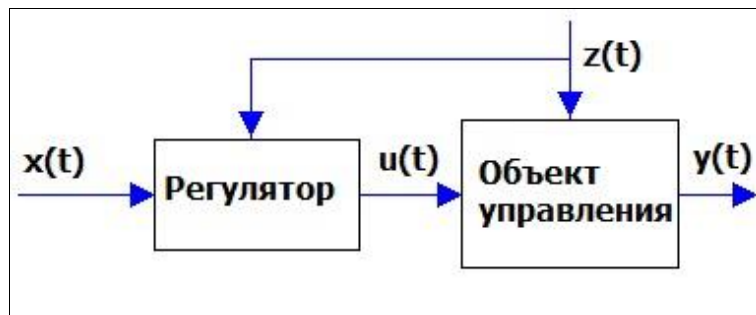


Рис. 3.95. Управление по возмущению

Схемы с управлением по возмущению (рис. 3.95) используются в случае, если возможно ввести коррективы в алгоритм управления для компенсации возмущения $z(t)$. Пример: биметаллическая система стержней в маятнике хронометра обеспечивает постоянную длину маятника при колебаниях температуры.

При управлении с обратной связью (рис. 3.96) система управления учитывает изменение состояния объекта управления $y(t)$. Его функционирование обеспечивается без измерения возмущения, только на основе информации о состоянии объекта управления. Примером может служить система регулирования паровой машины с регулятором Уатта.

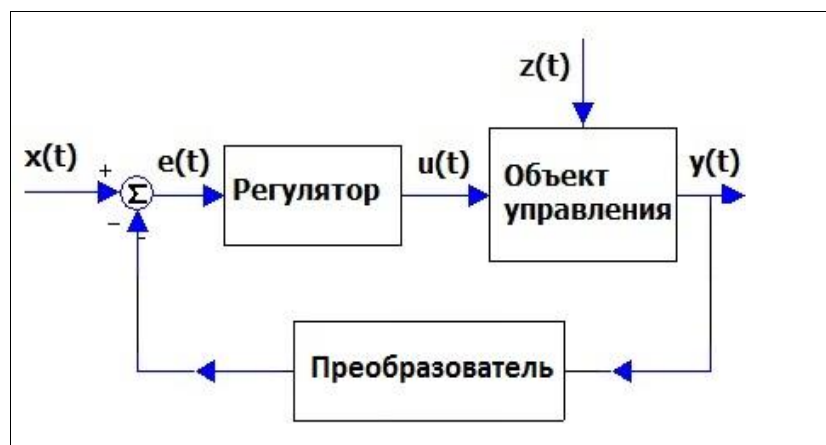


Рис. 3.96. Управление с отрицательной обратной связью

В подобных системах сигнал, поступающий по линии обратной связи, вычитается из задающего воздействия (отрицательная обратная связь). Если состояние объекта управления изменяется, то обратная связь стремится компенсировать изменения состояния объекта управления. Системы с обратной свя-

зью при определенных сочетаниях параметров, например при наличии запаздывания в линии обратной связи, могут быть неустойчивыми, т.е. способны дестабилизировать состояние объекта управления. Здесь: $x(t)$ – внешнее воздействие, $e(t)=x(t)-y(t)$, $z(t)$ – возмущение, $u(t)$ – воздействие регулятора на объект управления, $y(t)$ – регулируемый параметр.

Математическое моделирование систем управления – это классический подход к анализу подобных систем, который составляет основу теории автоматического управления и регулирования. Сегодня к чисто математическому моделированию добавилось компьютерное моделирование. В рамках теории автоматического регулирования были разработаны специфические методы анализа, которые затем использовались в других науках. Примером может служить теория устойчивости.

Методы математического моделирования, несомненно, актуальны в теории управления. Экспериментальный метод в данной области уместен лишь на этапе испытаний опытных образцов систем управления.

Первоначально в теории автоматического управления развивались аналитические и приближенные методы. В настоящее время разработаны специализированные программные комплексы компьютерного моделирования, предназначенные для моделирования систем управления и решения специальных задач этой области. Например, инструментальные системы компьютерного моделирования Simulink, VisSim и российский русскоязычный аналог программный комплекс MBTU (Моделирование В Технических Устройствах), разработанный в МГТУ им. Баумана. Последний программный комплекс предназначен для численного исследования на математических моделях нестационарных процессов в технических системах. Модели систем строятся как модели с сосредоточенными параметрами в виде обыкновенных дифференциальных и алгебраических уравнений. Этот программный комплекс позволяет решать различные задачи моделирования: анализа, синтеза, идентификации, контроля и управления.

Принцип моделирования программного комплекса MBTU состоит в создании и исследовании виртуальной модели реальной системы, которая представляет собой блок-схему.

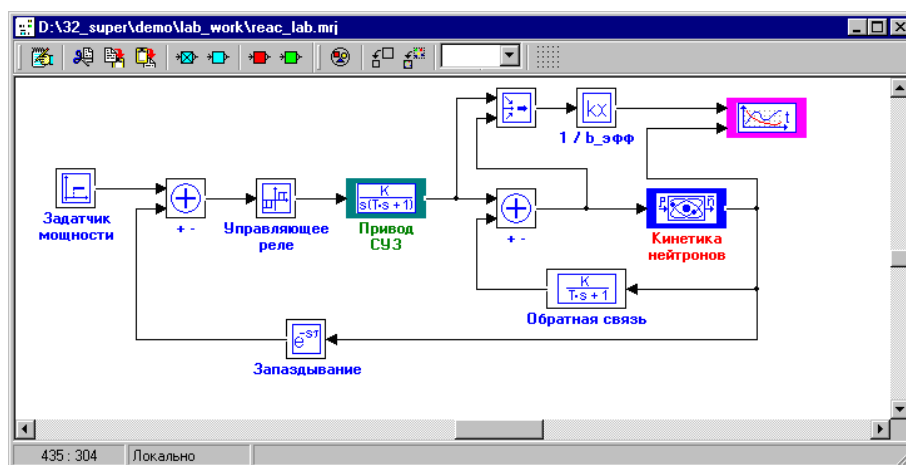


Рис. 3.97. Модель системы управления в среде комплекса моделирования MBTU

При моделировании не обязательно записывать уравнения модели в явном виде, модель конструируется из готовых блоков путем их копирования и соединения в блок-схему (рис. 3.97).

Принцип построения моделей в среде MBTU аналогичен технологиям построения моделей в программных комплексах VisSim и Simulink.

Программный комплекс VisSim предназначен для построения и исследования виртуальных моделей технических объектов, в том числе и систем управления. Этот комплекс разработан и развивается компанией «Visual Solutions». VisSim – это аббревиатура выражения Visual Simulator – визуальная среда моделирования. Он предоставляет пользователю графический интерфейс, с помощью которого исследователь создает модель из виртуальных элементов, что позволяет исследовать модели достаточно сложных систем. По технологии моделирования VisSim подобен другим аналогичным программным комплексам (рис. 3.98).

Особенностью этого пакета моделирования является ориентация на задачи анализа систем автоматического управления. При описании и последующем построении модели в среде VisSim численные методы реализации модели выбираются автоматически. Результаты решения могут быть представлены в виде

временных и фазовых диаграмм. При построении моделей в среде VisSim программирование не требуется, однако имеется возможность создания дополнительных моделей элементов и помещение этих моделей в библиотеку стандартных блоков VisSim.

Математическое и компьютерное моделирование систем управления основано на их декомпозиции, т.е. разделении на элементы (звенья).

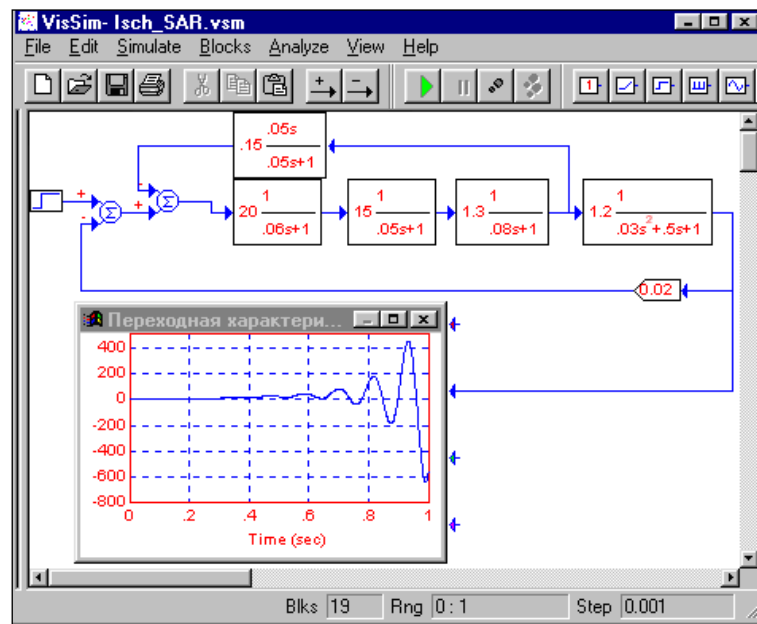


Рис. 3.98. Виртуальный стенд моделирования пакета VisSim

Математические модели элементов строятся на основе закономерностей процессов, которые в них протекают. Накопленный опыт исследования и разработки систем управления позволили выделить типовые математические модели элементов систем управления, из которых и создаются системы. Более сложные элементы могут быть получены путем соединения простейших типовых звеньев. Анализ свойств систем строится на результатах исследования свойств типовых элементов и свойств соединений элементов в более сложные агрегаты.

Дополнительные теоретические материалы по моделированию систем можно найти в монографиях, представленных в библиографическом списке: [3; 9; 10; 12; 14; 15–17; 18; 22–38, 42–52, 55–56].

3.12. КОНТРОЛЬНЫЕ ВОПРОСЫ К ГЛАВЕ 3

1. Что такое сложная система?
2. Что такое модель «черный ящик»?
3. Каковы особенности моделирования сложных систем?
4. В чем сущность метода статистического моделирования?
5. Что такое имитационное моделирование, в чем его актуальность?
6. Какие задачи решаются при моделировании сложных систем?
7. В чем сущность метода стохастического моделирования?
8. Каковы причины актуальности имитационного моделирования?
9. Каков принцип функционирования клеточного автомата?
10. Перечислите преимущества моделей типа «клеточный автомат».
11. Что такое агентное моделирование?
12. Как построить модель случайного события?
13. Как построить модель полной группы событий?
14. Как построить модель событий, которые могут проявляться совместно?
15. Какова цель корреляционного анализа?
16. Какова главная цель планирования эксперимента?
17. Какова структура простейшей системы массового обслуживания?
18. Каковы особенности моделирования систем массового обслуживания?
19. Что такое регрессионная модель?
20. В чем суть корреляционного анализа?
21. В чем суть полного факторного эксперимента?
22. Перечислите назначение и состав оптимизационных моделей.
23. Каковы особенности технологий трехмерного моделирования?
24. Что такое САПР?
25. Что такое управление?
26. Что такое регулирование?
27. Каковы виды управления?
28. Что такое управление с обратной связью и без нее?
29. Почему обратная связь должна быть отрицательной?

ГЛАВА 4. МОДЕЛИРОВАНИЕ С RAND MODEL DESIGNER И ANYLOGIC

4.1. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ МОДЕЛИРОВАНИЕ

Объектно-ориентированный подход в моделировании, который реализуется в RMD (Rand Model Designer) и AnyLogic, ассоциируется с объектно-ориентированным программированием (ООП). Однако первоначально он был использован в языке моделирования Simula-67. В дальнейшем этот подход развивался исключительно программистами. Итогом развития стал унифицированный язык моделирования UML (Unified Modeling Language).

Модификации ООП применительно к моделированию сложных многокомпонентных динамических систем называются **объектно-ориентированным моделированием** (ООМ) и представлены языком моделирования Modelica, а также входными языками пакетов моделирования AnyLogic, MVStadium, RMD и др.

Понятия класса и экземпляра являются ключевыми в ООМ.

Объект – это некоторая сущность, обладающая атрибутами и поведением.

Класс – это формализованное описание множества объектов, имеющих одинаковое функциональное назначение, атрибуты и поведение.

С понятием класса неразрывно связано понятие **экземпляра класса**, т.е. конкретного объекта из множества всех объектов, описываемых данным классом. Конкретный объект отличается от всех остальных объектов того же самого класса уникальными значениями **атрибутов**.

Например, пусть имеется класс «**Снаряд**», все экземпляры которого имеют одни и те же атрибуты: масса, горизонтальная и вертикальная координаты, горизонтальная и вертикальная скорости. Однако значения этих атрибутов для объектов «**Снаряд_1**» и «**Снаряд_2**» (экземпляров класса «**Снаряд**») различны.

Среди атрибутов выделяются **«параметры»** – такие атрибуты, значение которым присваивается только один раз при создании экземпляра класса и далее не меняется в течение всего времени существования объекта. У снаряда параметрами могут быть масса, начальная скорость и угол стрельбы.

В UML выделяются «активные» и «пассивные» объекты, и, соответственно, «активные» и «пассивные» классы, экземплярами которых они являются.

Активным объектом согласно UML называется «объект, который владеет потоком управления и может инициировать управляющие воздействия».

Пассивным объектом называется «объект, у которого нет собственного потока управления. Все его операции выполняются под управлением потока, прикрепленного к активному объекту».

В ООМ сложных динамических систем (СДС) определение «поток управления» заменяется определением «процесс, развивающийся во времени» и «управляющие воздействия» – «воздействия на другие объекты». Для компонентов СДС вводится специальное понятие **«активный динамический объект»** (АДО) – объект, имеющий деятельность, независимую от поведения других объектов. Интуитивно под АДО понимается некоторая совокупность атрибутов (констант, параметров, переменных) и поведения – непрерывного, дискретного или гибридного. Переменные соответствуют атрибутам активного объекта UML. Переменные можно разделить на внешние (входы, выходы, контакты, коннекторы) и внутренние. АДО взаимодействует с другими объектами через внешние переменные.

Класс, экземплярами которого являются АДО, называется **активным классом**. Поведение активного объекта языка UML задается «картой состояний» («**statechart**»), которая может описывать только дискретное поведение. Поведение АДО задается «картой поведения» («**behavior chart**» или «**B-chart**»), являющейся обобщением карты состояний.

Основным отличием карты поведения от карты состояний является возможность приписать состоянию непрерывную деятельность, задаваемую системой дифференциально-алгебраических уравнений. Более сложными понятиями ООМ являются наследование и полиморфизм.

Часто возникает необходимость создать новый класс «такой же, но...». Например, нужно описать бассейн с подогревом воды, дополнив описание стандартного бассейна нужными деталями. В этом новом классе «**Бассейн_с_подогревом**» описание интерфейса и динамики уровня воды будет точно таким же, как и в классе «**Бассейн**». К нему добавится свое описание тепловых потоков и динамики изменения температуры.

Можно просто перенести в описание нового класса элементы описания старого и добавить новые. Но можно объявить новый класс прямым потомком старого. В этом случае класс «**Бассейн**» будет являться **суперклассом** (родителем, базовым классом) для класса «**Бассейн_с_подогревом**», а тот в свою очередь будет являться **подклассом** (потомком, производным классом) по отношению к классу «**Бассейн**». В этом случае производный класс автоматически унаследует все элементы описания своего базового класса. Следует отметить, что **наследование** не означает простого копирования. Между классами возникает постоянная связь: если в классе «**Бассейн**» добавить новую переменную состояния (например, показатель хлорирования воды), то она автоматически появится в классе «**Бассейн_с_подогревом**».

Полиморфизм означает возможность использования вместо экземпляра блока некоторого базового класса экземпляром любого его производного класса.

Например, для радиолокационной станции все сопровождаемые объекты являются экземплярами класса «**Летательный_аппарат**» и характеризуются только положением и вектором скорости. На самом же деле эти объекты могут являться самыми разнообразными потомками класса «**Летательный_аппарат**»: от бомбардировщика В-52 до бумажного голубя.

RMD – это инструмент моделирования, позволяющий создавать компонентные модели непрерывных, дискретных и гибридных (непрерывно-дискретных) систем, проводить с ними вычислительные эксперименты и встраивать их в приложения.

Среда RMD работает на Intel-совместимых компьютерах под ОС Microsoft Windows (рис. 4.1).

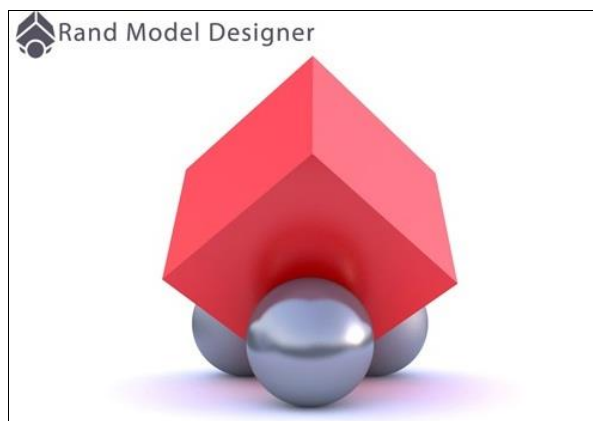


Рис. 4.1. Логотип RMD

Разработка моделей в среде RMD происходит исключительно на уровне математических зависимостей и диаграмм языка UML, не требуется использования программирования.

Среда RMD является компилирующей системой моделирования: выполняемая модель является двоичным выполняемым файлом Windows (DLL или EXE). Программный код выполняемой модели формируется автоматически по математическому описанию. Базовым формализмом для языка моделирования среды RMD является обобщенный гибридный автомат.

Работа со средой RMD начинается с того, что пользователь запускает интегрированную среду. Окно интегрированной среды выглядит, как показано на рис. 4.2. Интегрированная среда – это визуальное средство разработки моделей. Используя интегрированную среду, пользователь создает проект, являющийся описанием модели, и наполняет его модельными элементами: с помощью редактора уравнений вводит системы уравнений в естественной математической форме, с помощью редактора карт поведения рисует карты поведения и т.д.

Цель наполнения проекта – создание адекватной модели моделируемой системы. В процессе создания описания осуществляется контроль синтаксической и семантической корректности вводимых конструкций.

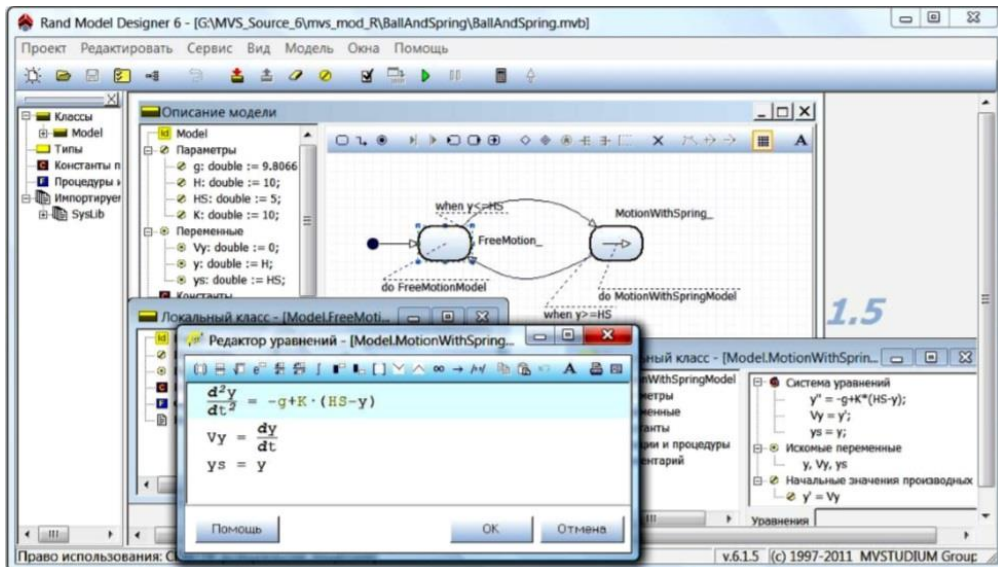


Рис. 4.2. Окно интегрированной среды моделирования RMD

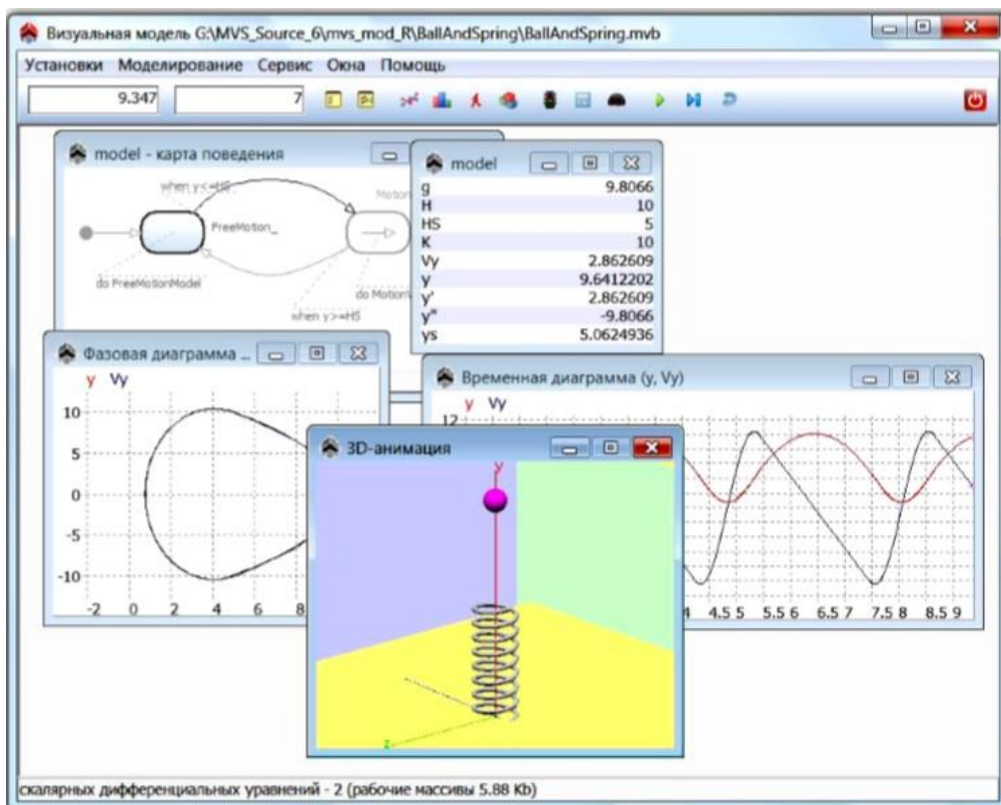



Рис. 4.3. Визуальная выполняемая модель

Когда описание модели готово, модель нужно отладить, проведя контрольные вычислительные эксперименты. Для этого пользователь запускает визуальную модель, нажимая на кнопку  главной панели инструментов или выбирая соответствующий пункт главного меню. В ответ на это среда RMD автоматически строит и запускает визуальную выполняемую модель. **Визуальная**

выполняемая модель – это самостоятельная программа, предоставляющая широкий спектр возможностей для интерактивного эксперимента с моделью. Пример визуальной выполняемой модели показан на рис. 4.3.

Визуальная модель обладает широким набором **возможностей**:

- анимация работы карт поведения;
- динамическая визуализация значений переменных;
- построение временных и фазовых диаграмм;
- построение гистограмм;
- 2D- и 3D-анимация;
- динамическая визуализация текущей системы уравнений;
- визуализация структуры решаемой системы уравнений.

В визуальной модели доступны **средства планирования компьютерного эксперимента и средства отладки**:

- составление плана простого эксперимента;
- интерактивное изменение значений переменных;
- создание виртуальных переменных;
- остановки по времени (непрерывному и дискретному);
- останов по входу в состояние;
- останов по срабатыванию перехода;
- останов по условию;
- сохранение и восстановление текущего состояния модели;
- сравнение текущего состояния модели;
- хронометрирование работы модели.

По завершении отладки готовая модель может быть:

- использована для анализа моделируемой системы;
- использована как пакет готовых стандартных компонентов (библиотека классов) для данной содержательной области;
- встроена в виде DLL во внешние приложения;
- сохранена как отдельное приложение для использования независимо от RMD.

4.2. ПРОЕКТ RMD «НЕПРЕРЫВНАЯ МОДЕЛЬ»

Проектом в RMD является совокупность данных, относящихся к определенной модели и хранящихся в отдельной папке. Проект соответствует языковой единице «пакет» («package») языка моделирования.

В качестве примера рассмотрим процесс создания и отладки модели математического маятника (материальная точка, подвешенная на невесомом нерастяжимом стержне длиной L , другой конец которого шарнирно закреплен, рис. 4.4). Такая модель является изолированной, однокомпонентной системой и требует только определения переменных, описывающих состояние модели, и задания уравнений, описывающих ее поведение.

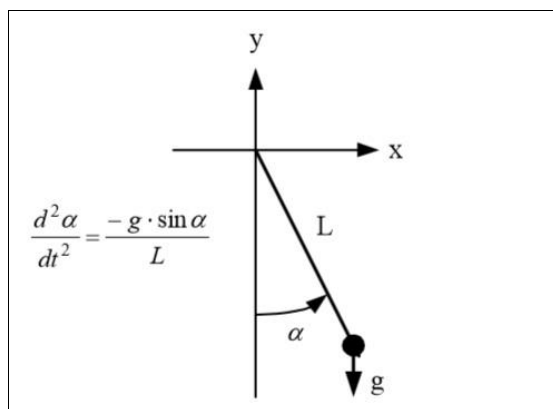


Рис. 4.4. Математический маятник

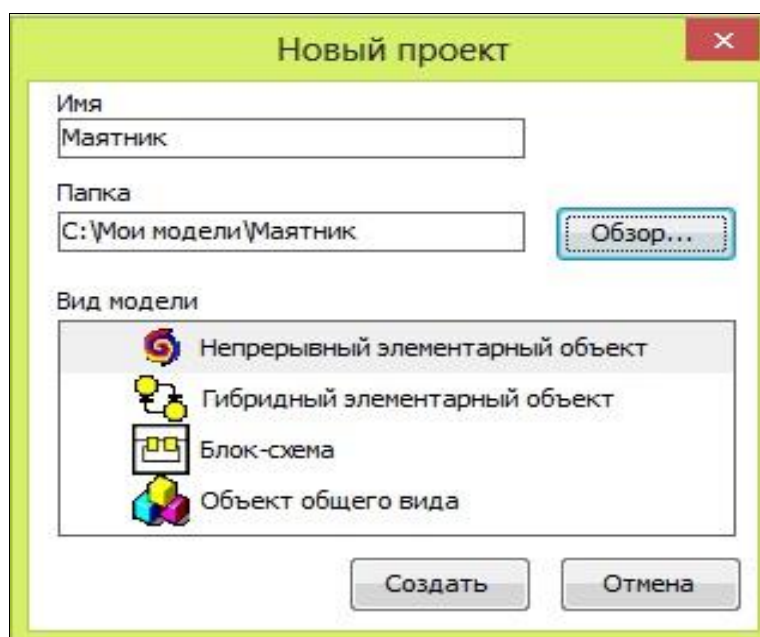


Рис. 4.5. Новый проект

Создадим новый проект с именем «Маятник». Для этого в интегрированной среде Rand Model Designer выполните команду главного меню «**Проект/Новый...**» или нажмите кнопку на инструментальной панели. В появившемся диалоге (рис. 4.5) выберите имя папки, «**С:\Мои модели**», введите имя проекта, «**Маятник**» и выберите вид модели (в данном случае «**непрерывный элементарный объект**»).

После нажатия кнопки «**Создать**» в выбранной папке появится файл «**Маятник.mvb**», содержащий базу данных проекта. Для удобства мы будем каждую модель размещать в отдельной папке, однако с помощью команды «**Проект-Сохранить как**» можно сохранять в одной папке различные модификации проекта (например, «**Маятник_1.mvb**»). В процессе работы над проектом в папке «**С:\Мои модели\Маятник**» появятся еще два файла, в которых сохраняются соответственно установки проекта и установки визуальной выполняемой модели. На рис. 4.6 показан начальный вид окон редактора моделей – главное окно интегрированной среды RMD.

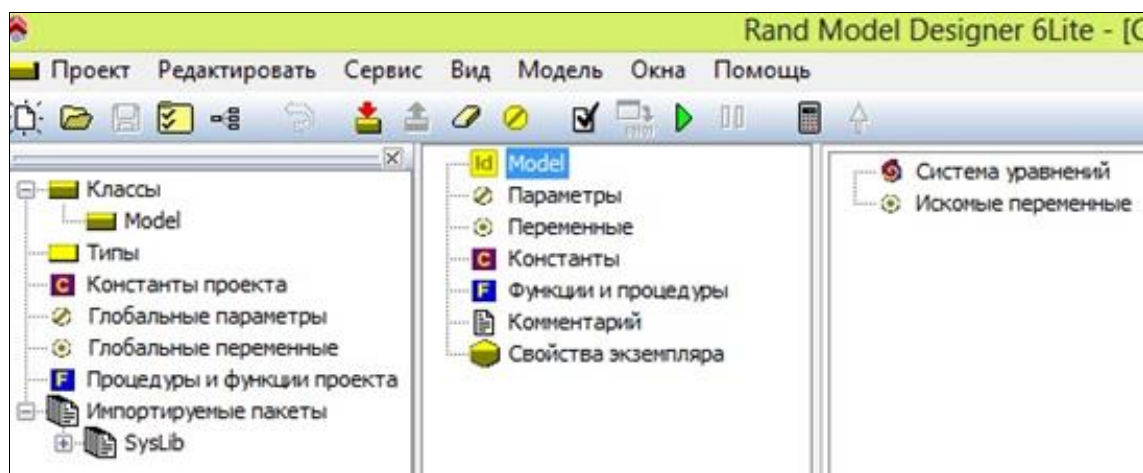


Рис. 4.6. Главное окно проекта

Основными элементами главного окна являются:

Главное меню (выделено красным цветом, рис. 4.7). Содержит в себе пункты меню, позволяющие осуществлять все основные манипуляции с проектом (открытие/закрытие проекта, сохранение проекта, установки проекта, запуск визуальной модели и т.д.);

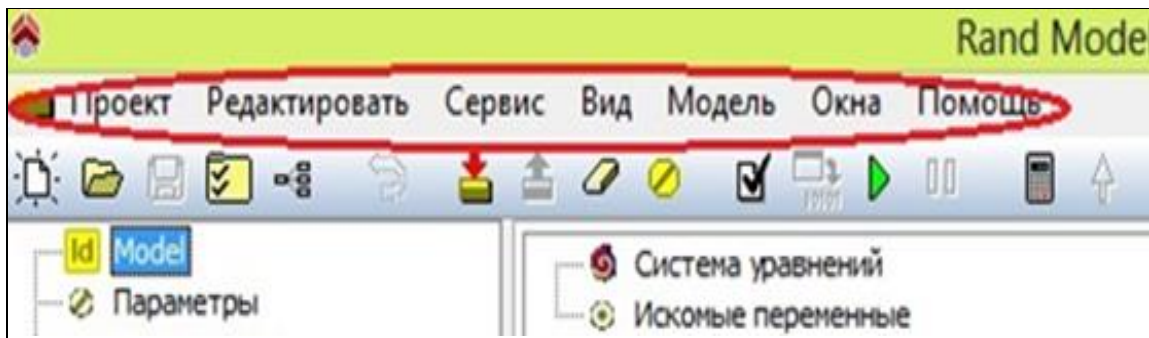


Рис. 4.7. Главное меню

Главная панель инструментов (рис. 4.8). Содержит в себе кнопки быстрого доступа к наиболее часто используемым командам **главного меню**.

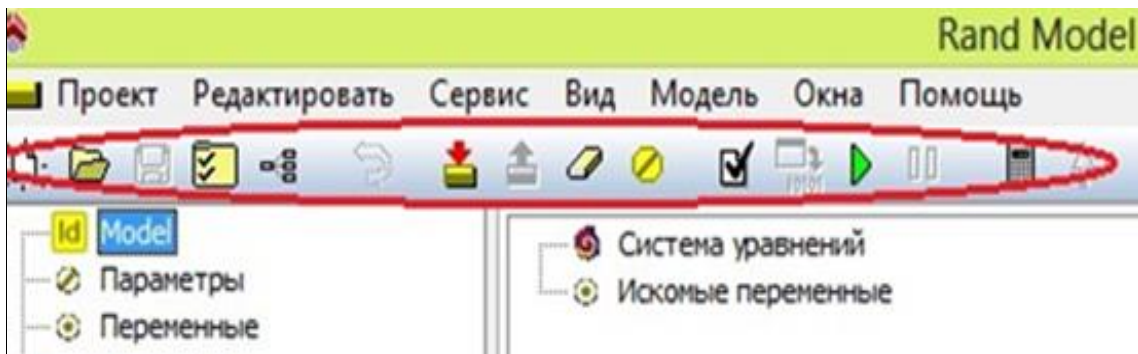


Рис. 4.8. Главная панель инструментов

Структура главного окна и количество окон редактора моделей зависит от выбранного вида модели. В нашем простейшем случае используется одно максимизированное окно редактора класса «**Model**». Особый класс Model – неотъемлемая часть любого проекта RMD. Проект представляет собой описание модели, с которой будет в дальнейшем проводиться вычислительный эксперимент. Выбранный при создании проекта «вид модели» определяет стереотип этого класса. В дальнейшем вы можете изменить его с помощью команды «**Преобразовать в...**» всплывающего меню, после нажатия правой кнопки мыши в поле окна «Model». В общем случае главное окно интегрированной среды включает в себя менеджер проекта в левой части окна (рис. 4.9) и многооконную область редакторов классов (рис. 4.10).

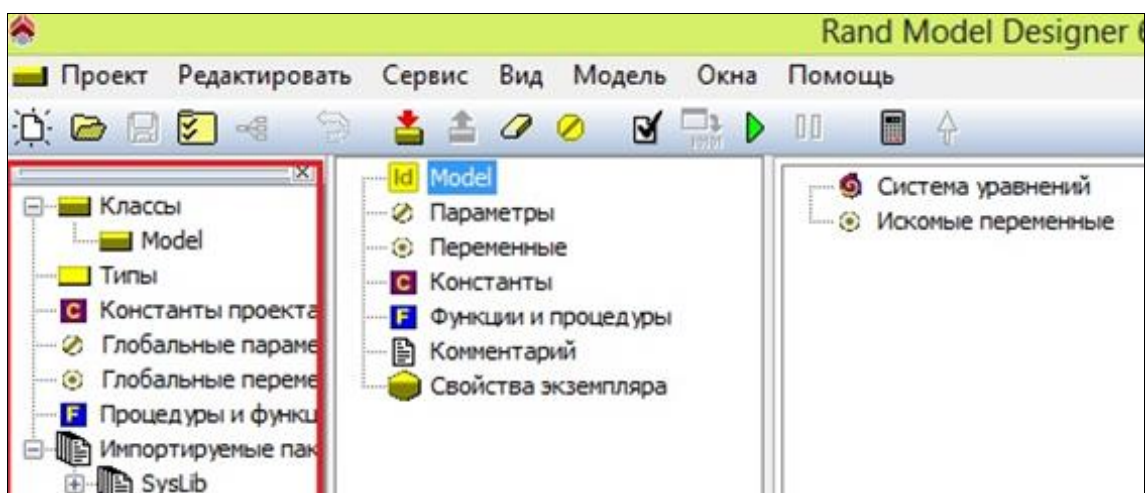


Рис. 4.9. Менеджер проекта

«Менеджер проекта» – это дерево компонентов проекта (классов, констант, процедур и функций), а также импортируемых пакетов. В новом проекте всегда присутствует класс «Model» и импортируемый пакет стандартных классов «SysLib».

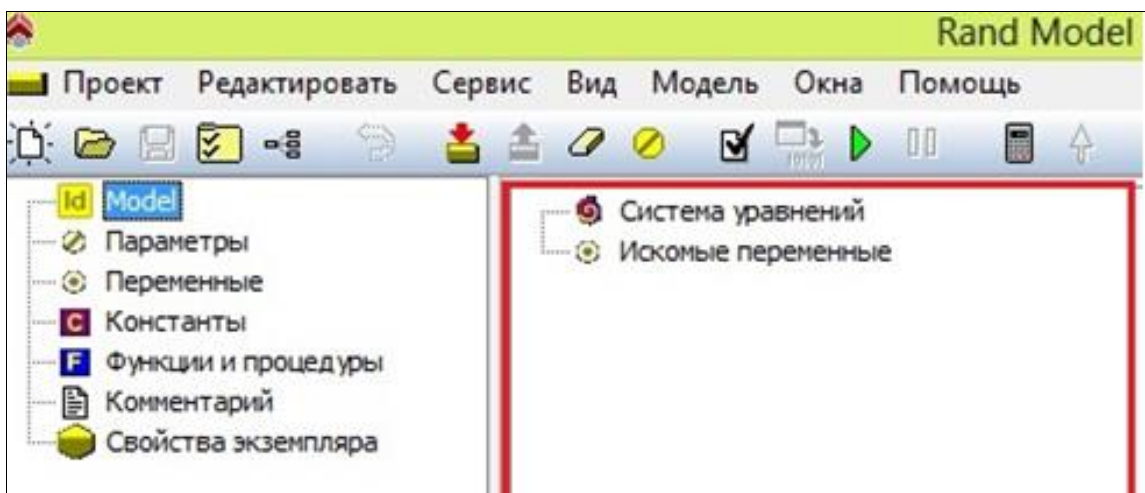


Рис. 4.10. Область редактирования классов

Вы можете открыть/закрыть окно менеджера проекта с помощью кнопки панели инструментов или команды меню «**Вид / Менеджер проекта**».

Площади, занимаемые окном менеджера проекта и редакторами классов, можно изменить с помощью перемещаемого разделителя (сплиттера), совмещенного с правой границей окна менеджера проекта.

Поведение маятника описывается дифференциальным уравнением второго порядка (рис. 4.4) относительно угла отклонения маятника, в котором есть два параметра: длина маятника (L) и ускорение свободного падения (g). Уравнение второго порядка требует также задания двух начальных условий: значения угла отклонения и значения угловой скорости маятника в начальный момент времени $t=0$.

При создании любой модели вы должны решить, к какому виду (стереотипу) относится та или иная величина. **Константы** не меняют своего значения никогда. Значения **параметров** могут быть изменены при создании объекта – экземпляра данного класса, но не могут меняться в процессе функционирования этого объекта. Значения **переменных** могут изменяться в процессе функционирования, и они могут быть искомыми переменными уравнений.

В данном случае угол отклонения α – переменная (поскольку греческих букв RMD не поддерживает, дадим этой переменной имя *Alpha*). Начальное отклонение *Alpha0* и начальная скорость *Omega0* должны быть параметрами, если мы захотим в дальнейшем одновременно наблюдать поведение нескольких маятников с различными условиями. Длину стержня L можно также назвать параметром, поскольку возможен эксперимент с несколькими маятниками разной длины. Наконец, ускорение g может быть как константой, если мы предполагаем, что все наши маятники будут расположены на уровне моря, так и параметром или переменной, если мы допускаем использование маятника на разных высотах или разных планетах.

Явно определим параметры L , $g0$, *Alpha0*, *Omega0* и переменные *Alpha* и g в окне класса Model. Можно было бы сразу перейти к написанию системы уравнений без предварительно определения переменных в классе Model. В этом случае редактор уравнений все вводимые пользователем имена при написании уравнений и формул будет рассматривать как скалярные переменные типа double с нулевыми начальными значениями. После ввода и проверки уравнений останется придать переменным нужные начальные значения. Для добавления нового параметра в описание класса Model перейдем в «Мене-

джер проекта», нажмем правую кнопку мыши на секции «**Параметры**» и выполним команду «**Новый параметр**» (рис. 4.11).

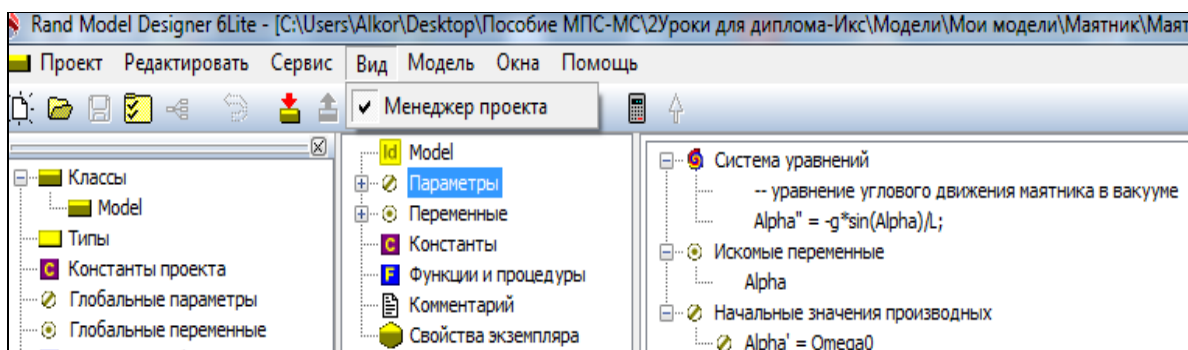


Рис. 4.11. Добавление нового параметра

После этого укажем в появившемся диалоге имя, тип, начальное значение и, если нужно, добавим комментарий (рис. 4.12). Аналогичным образом введем остальные параметры, переменные и константы. Результаты редактирования

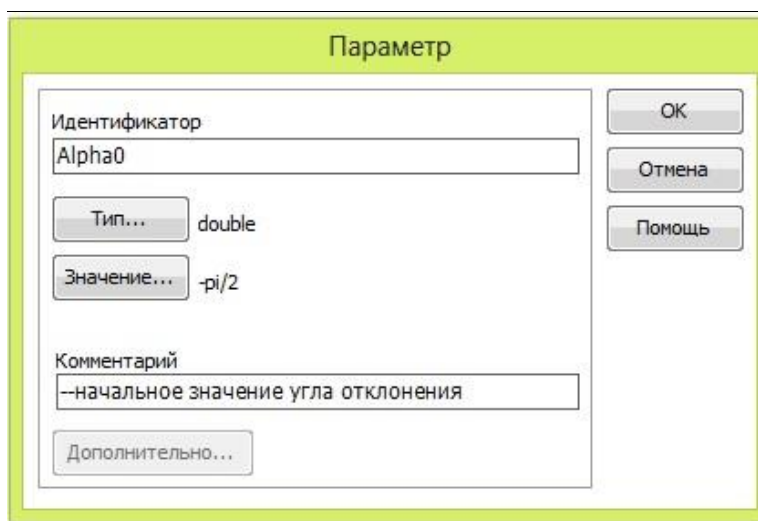


Рис. 4.12 Создание параметра

появятся в соответствующих элементах дерева проекта в главном меню проекта (рис. 4.13). Для редактирования определения переменной или параметра, выделите соответствующий элемент «дерева» и дважды щелкните мышью или выполните команду «**Изменить**» всплывающего меню. Если вы в дальнейшем захотите изменить стереотип, например сделать параметр g константой, то для

этого достаточно просто перетащить мышью методом «drag-and-drop» определение переменной из одной секции («**Параметры**») в другую («**Константы**»).

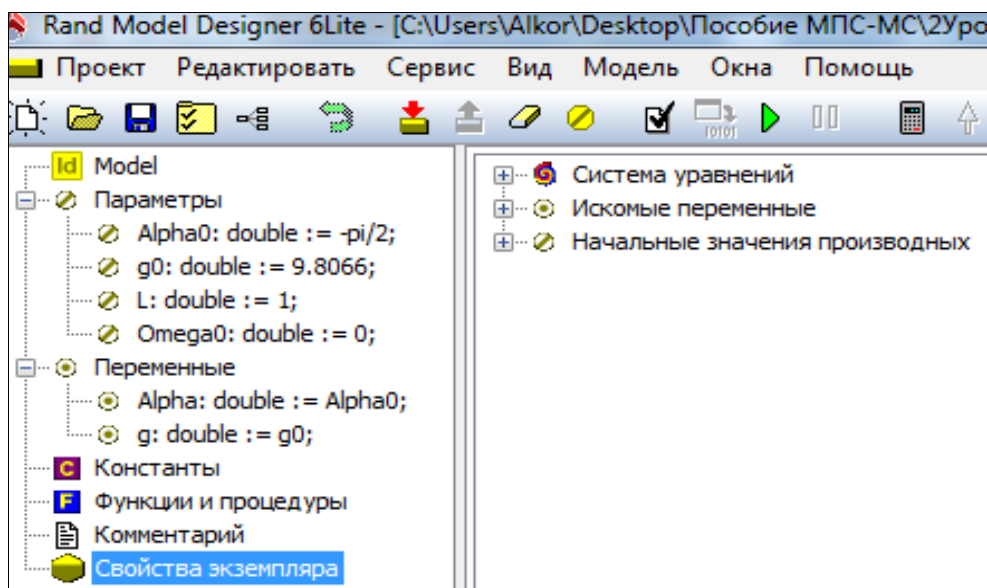



Рис. 4.13. Параметры и переменные проекта

Откроем уже существующий проект, если он был закрыт и сохранен. Для этого выберите в главном меню пункт «**Проект/Открыть...**», либо нажмите на кнопку  на главной панели инструментов, либо воспользуйтесь комбинацией клавиш <Ctrl+O>. Откроется окно «**Открыть**» (рис. 4.14).

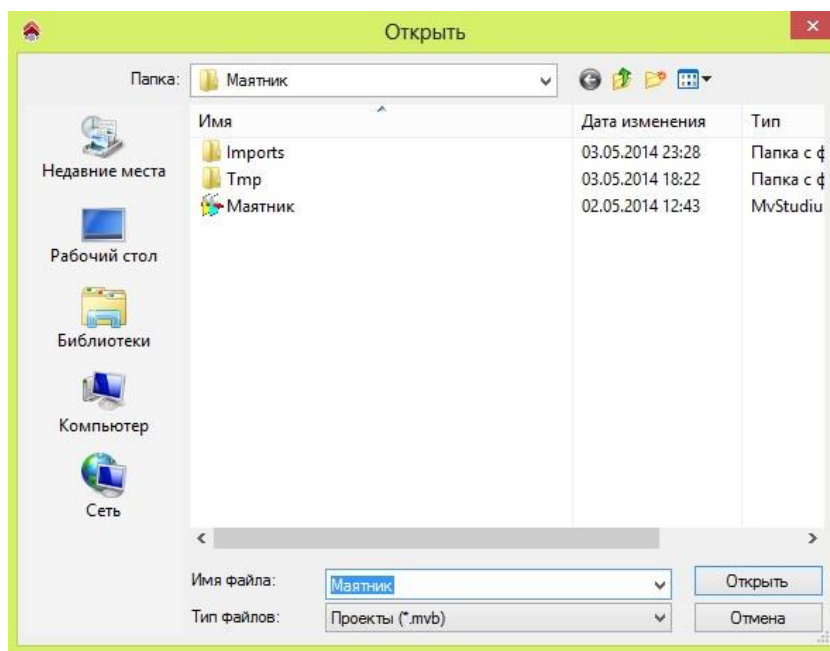


Рис. 4.14. Открытие существующего проекта

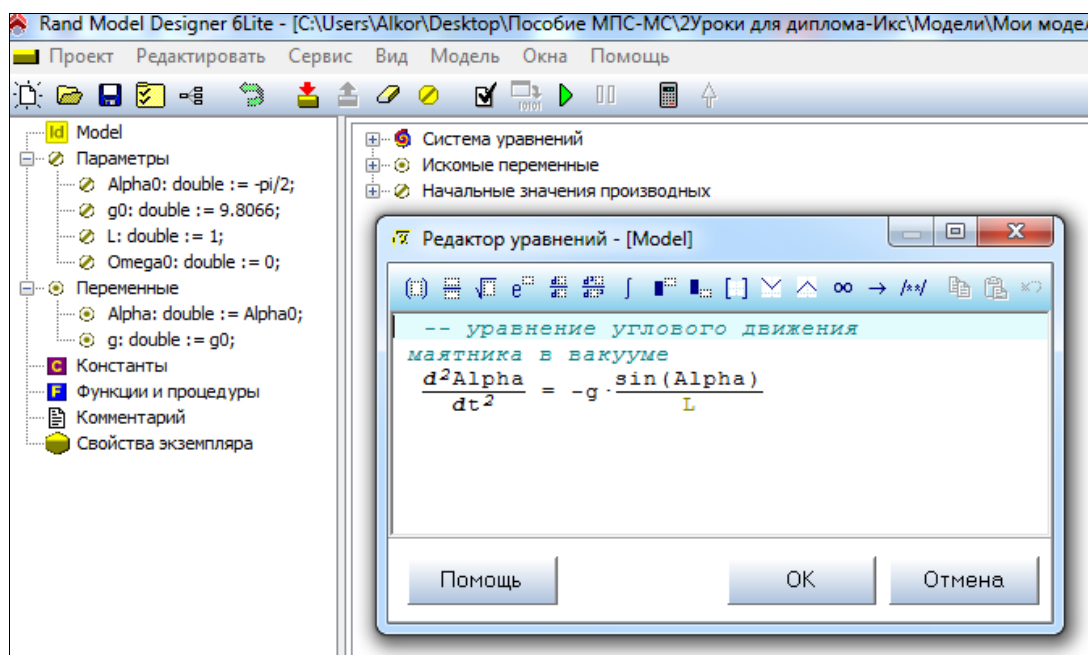


Рис. 4.15. Редактор уравнений

Для ввода или редактирования уравнений движения маятника дважды щелкните мышью на секции «**Система уравнений**» на страничке «**Уравнения**» или выполните команду «**Изменить**» всплывающего меню.

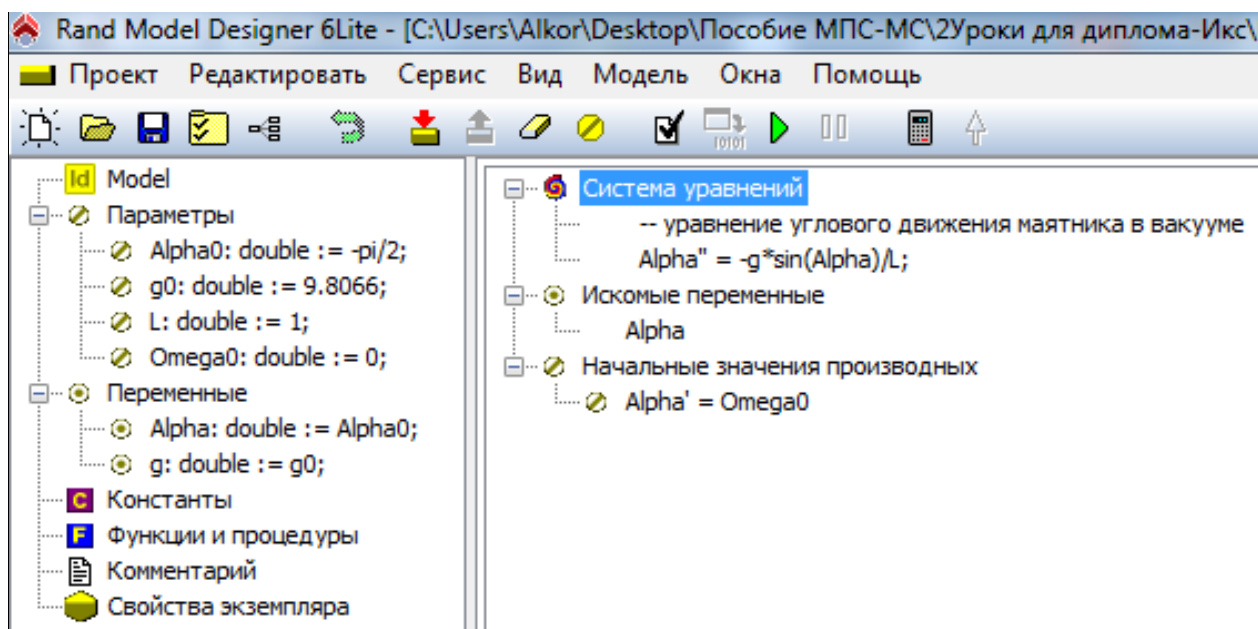


Рис. 4.16. Текстовое представление системы уравнений

В появившемся окне редактора уравнений можно ввести или изменить уравнения в естественной математической форме, с необходимыми коммента-

риями (рис. 4.15). Введите уравнения. После завершения редактирования системы уравнений нажмите кнопку ОК, и при отсутствии ошибок (при закрытии окна редактирования осуществляется синтаксический контроль) новый вид системы уравнений отобразится в текстовой форме на страничке «Уравнения» в окне редактора класса (рис. 4.16).

Отметим, что в текстовой форме производные обозначаются с помощью символа «'» ($Alpha'$, $Alpha''$). Обратите внимание на то, что производная искомой переменной $Alpha'$ появилась автоматически в поле начальных условий. «Редактор уравнений» пытается сам автоматически определить набор искомых переменных, и обычно это ему удается (в данном случае искомой может быть только переменная $Alpha$).

Если описание поведения содержит дифференциальное уравнение второго порядка относительно заданной переменной, то необходимо задать также начальное значение первой производной. Она не появляется в списке искомых переменных, но присутствует в поле начальных условий странички «Уравнения». По умолчанию значение производной принимается равным нулю. Чтобы изменить начальное значение производной, дважды щелкните мышью на соответствующей строке секции «Начальные значения производных» на страничке «Уравнения» и в появившемся диалоге введите нужное начальное значение (рис. 4.17).

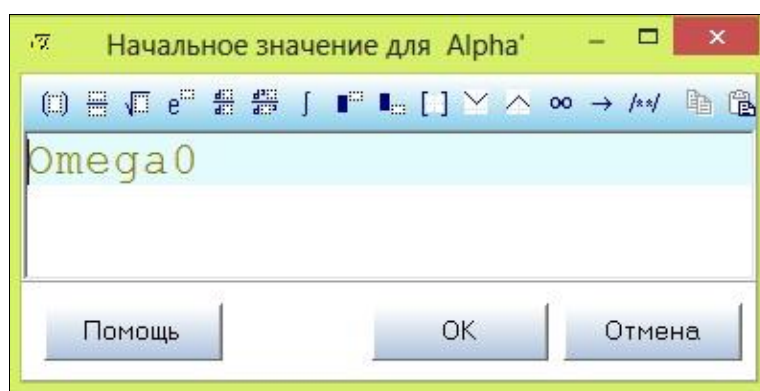





Рис. 4.17. Начальные значения производных

Графическая среда создает три типа моделей: визуальную модель, работающую под управлением среды; независимую от среды модель и «скрытые» модели в форме DLL.

С помощью команды **«Модель/Запустить»** главного меню или кнопки  на инструментальной панели создадим и запустим визуальную модель. Пакет RMD относится к категории «компилирующих»: для всех элементов модели генерируется «код» на промежуточном объектно-ориентированном языке программирования, который затем компилируется в машинный код и связывается с исполняющей системой RMD с помощью штатного компилятора командной строки этого языка. Вся работа с промежуточным кодом проводится в локальной папке «...**Тмп**» (в данном примере в «...**Модели\Маятник\Тмп**»), в ней же формируется файл **model.exe**. Обычно это занимает несколько секунд. При выходе из интегрированной среды все содержимое этой папки автоматически уничтожается. Для того чтобы сохранить модель как независимую от среды разработки выполняемую программу, сначала создайте визуальную модель с помощью команды **«Модель/Создать»** или кнопки  на инструментальной панели, а затем выполните команду **«Модель/Сохранить как»** главного меню. После создания визуальная модель запускается параллельно с интегрированной средой. В любой момент вы можете прервать выполнение модели из интегрированной среды (например, в случае ее «зависания») с помощью команды **«Модель/Стоп»** или кнопки  на инструментальной панели.

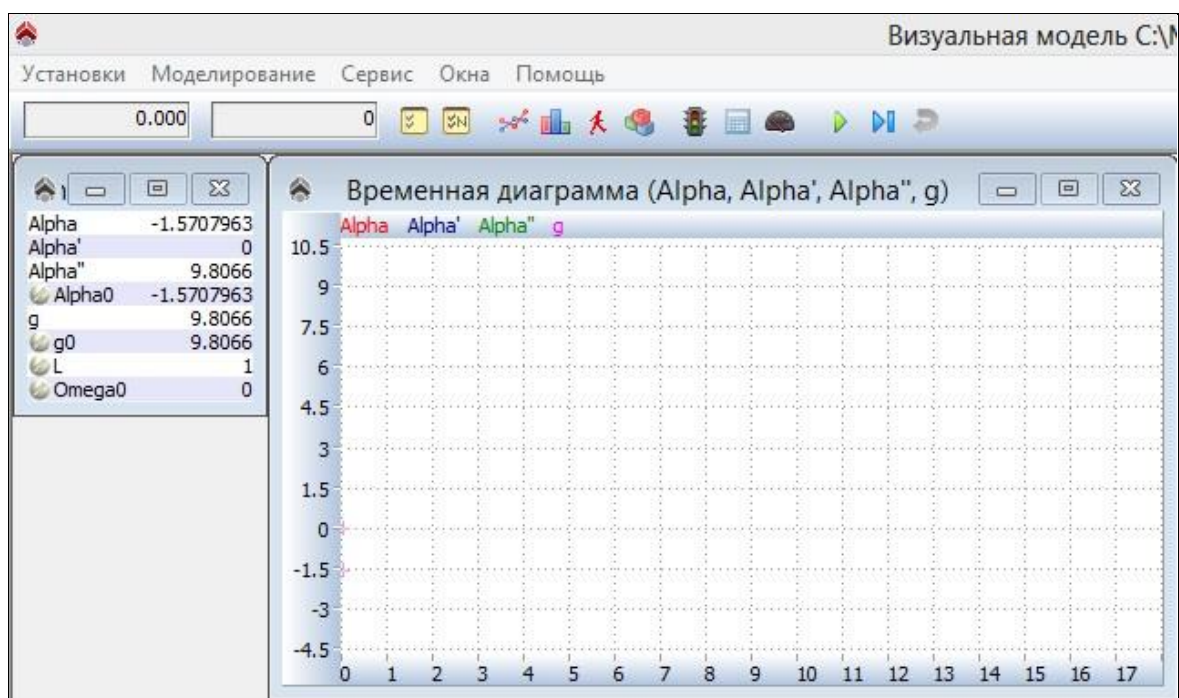







Рис. 4.18. Главное окно визуальной модели

На рис. 4.18 показано начальное состояние главного окна визуальной модели математического маятника при первом запуске. Модель находится в точке останова. Для непрерывной модели по умолчанию открываются:

- окно переменных (включая используемые производные);
- окно временной диаграммы для первых четырех искомым переменных.


В левой части инструментальной панели показаны непрерывная и дискретная составляющие модельного времени. Значения непрерывной составляющей модельного времени и окна переменных обновляются с частотой 10 Гц в реальном времени, значения дискретной составляющей изменяются при дискретных событиях.

Запустим продвижение модельного времени. Для этого нужно выполнить команду **«Моделирование/Пуск»** главного меню или нажать кнопку  на инструментальной панели. Система уравнений модели начнет численно решаться. По умолчанию в опциях численных методов (изменяются с помощью команды **«Установки/Численные методы»** или кнопки ) установлен автоматический выбор методов и погрешности, равные 10^{-6} . Эти установки подходят для большинства не очень сложных моделей. После запуска модели вы увидите как начнут изменяться значения непрерывной составляющей модельного времени и значения переменных в окне переменных, а также строиться графики на временной диаграмме. По умолчанию в опциях модели (изменяются с помощью команды **«Установки/Модель»** или кнопки  на инструментальной панели) вычисления происходят **«Так быстро, как возможно»**. При необходимости скорость течения модельного времени можно ускорить (пока позволяют возможности процессора) или замедлить.

В любой момент прогон модели можно остановить с помощью кнопки  или команды **«Моделирование/Стоп»**. Закончить прогон и начать новый можно с помощью кнопки  или команды **«Моделирование/Рестарт»**. В результате этих действий данный экземпляр класса Model уничтожается и создается новый, с новыми начальными значениями переменных, значение модельного времени устанавливается равным нулю (0,0).

В окне диаграммы отображаются временные или фазовые диаграммы переменных модели. По умолчанию окно диаграммы выглядит, как показано на рис. 4.18.

Окно диаграммы для переменных экземпляра класса Model открывается автоматически при запуске визуальной модели. Окно диаграммы для переменных других объектов может быть создано в процессе работы визуальной модели (рис. 4.18).

Создать новое окно диаграммы можно с помощью кнопки  на инструментальной панели или команды «**Окна/Новая диаграмма**» главного меню. По умолчанию эта диаграмма будет временной, т.е. по оси абсцисс будут откладываться значения непрерывной составляющей модельного времени, а по оси ординат – значения отображаемых переменных. В нашем простейшем примере искомые переменные уже были помещены в окно диаграммы автоматически. Для того, чтобы добавить новую переменную к уже отображающимся в окне, нужно новую переменную перетащить мышью методом «drag-and-drop» из окна переменных и поместить ее в поле окна диаграммы.

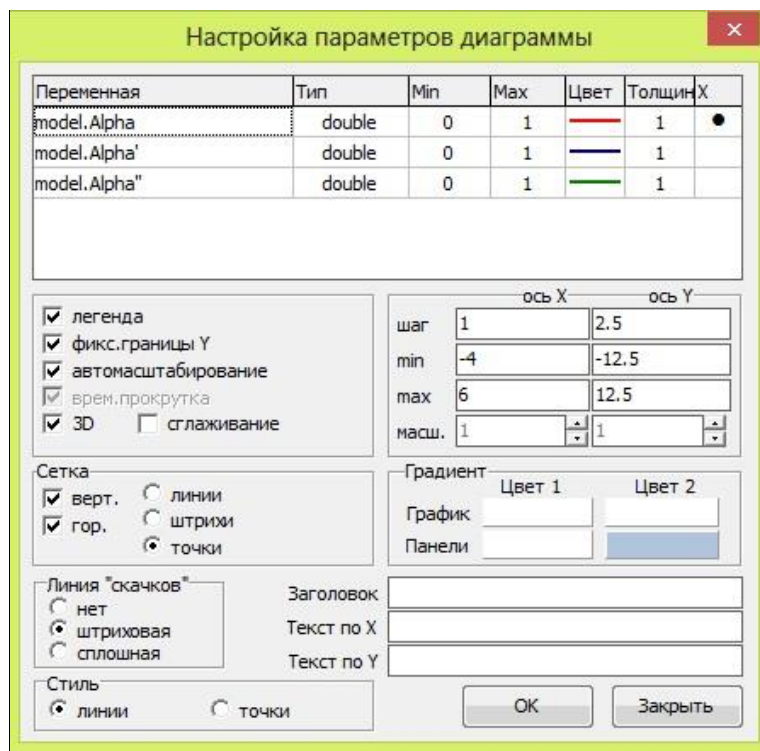


Рис. 4.19. Настройка диаграммы

Создадим в нашей модели маятника в дополнение к автоматически созданной временной диаграмме фазовую диаграмму, на которой отображались бы зависимости угловой скорости и углового ускорения маятника от угла отклонения. Создадим новую диаграмму и поместим на нее переменные $Alpha$, $Alpha'$ и $Alpha''$. С помощью двойного щелчка мышью в области диаграммы или команды «**Настройки**» всплывающего меню вызовем диалог настроек диаграммы (рис. 4.19).

В списке переменных окна выберем ту переменную, которая будет откладываться по оси абсцисс вместо модельного времени. Щелкнем мышью в последнем столбце X строки с именем выбранной переменной, например $Alpha$, появившийся знак «•» укажет, что эта переменная будет отображаться по оси абсцисс. Закроем окно, запустим модель и получим желаемую фазовую диаграмму (рис. 4.20).

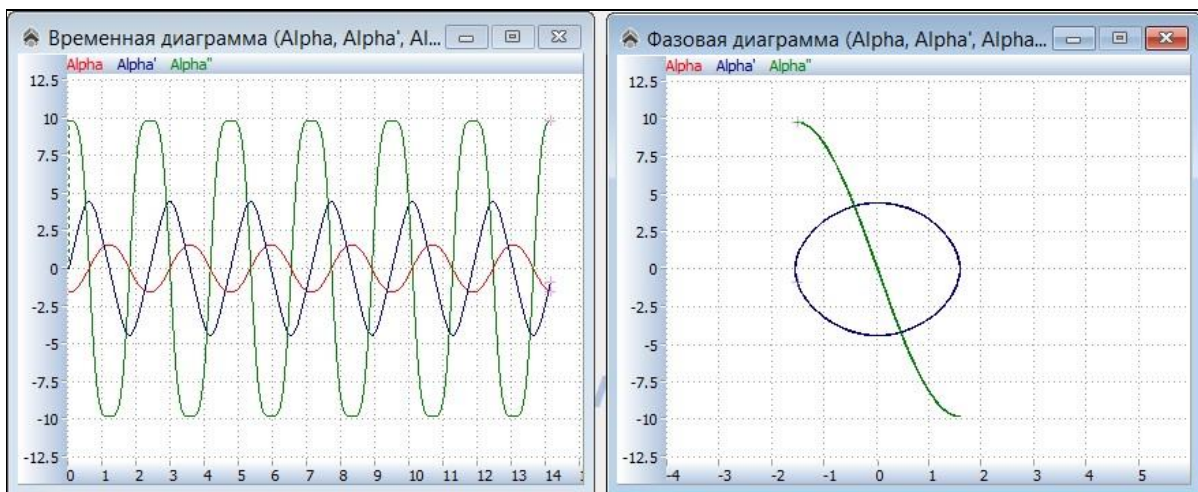


Рис. 4.20. Временная и фазовая диаграммы

Создадим новые переменные и дополним систему уравнени: $x = L \cdot \sin(Alpha)$, $y = -L \cdot \cos(Alpha)$ и $V = \text{abs}(L \cdot Alpha')$, соответствующие декартовым координатам и модулю линейной скорости маятника (рис. 4.21).

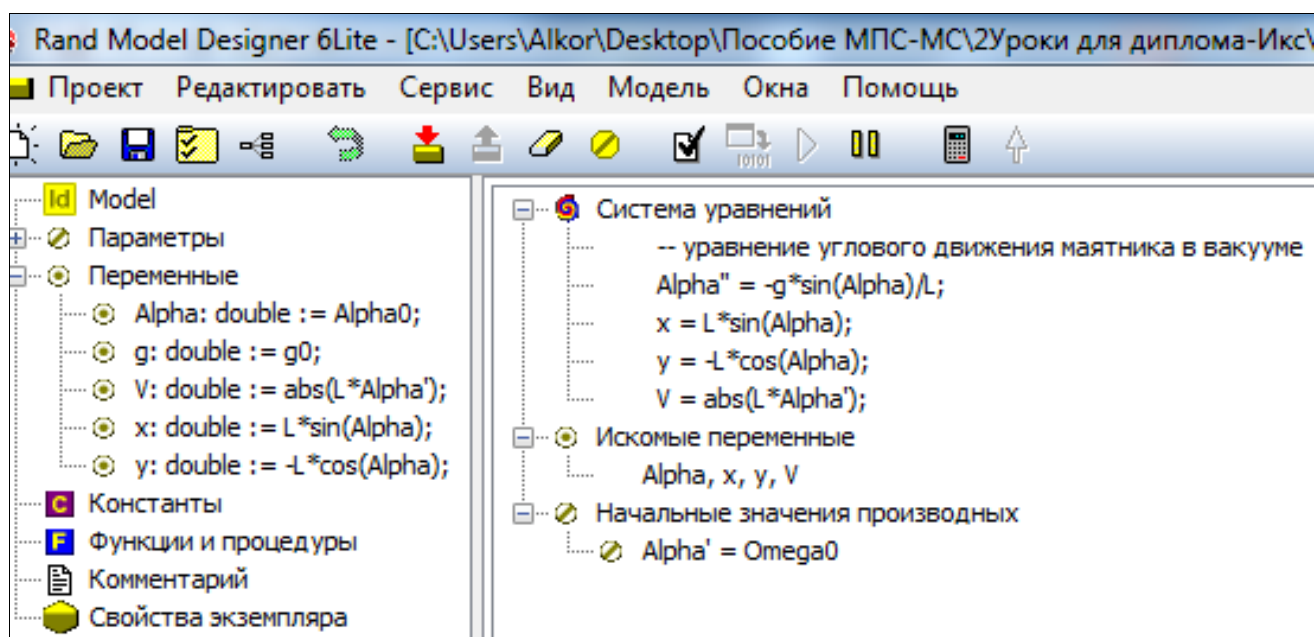



Рис. 4.21. Создание новых переменных и уравнений

Очень часто пользователь может получить больше всего информации при наблюдении за поведением трехмерного динамического изображения моделируемой системы в окне **3D-анимации** с помощью команды главного меню **«Окна/Новая 3D-анимация»** или при нажатии на кнопку  панели инструментов визуальной модели.

Динамические трехмерные изображения в окне 3D-анимации строятся из трехмерных примитивов (линий, шаров, цилиндров, конусов и т.д.), параметры которых (координаты, радиусу, цвет и т.п.) однозначно связываются со значениями соответствующих переменных модели.

Построим трехмерную анимацию движения маятника. Создадим новое 3D-окно и с помощью команды **«Параметры»** всплывающего меню вызовем диалог редактирования свойств 3D-анимации. В данной модели нам понадобится только два стандартных объекта – отрезок и сфера (рис. 4.22). Один конец линии должен всегда находиться в начале координат, а координаты второго конца (параметры x_2 , y_2) должны изменяться в соответствии со значением переменных x и y маятника. Для задания этого соответствия методом «drag-and-drop» перемещаем соответствующие переменные из окна переменных в колонку **«Переменная»** соответствующих параметров отрезка (рис. 4.22).

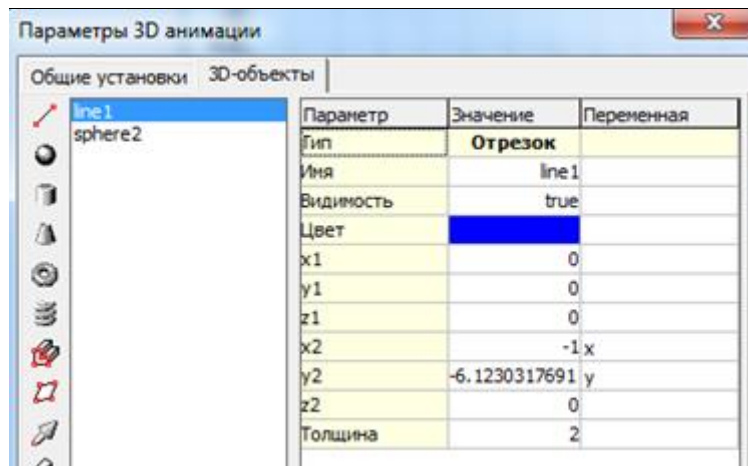


Рис. 4.22. Параметры 3D-анимации

Аналогичным образом эти же переменные x и y сопоставляем с координатами центра сферы. Трехмерная анимация маятника представлена на рис. 4.23.

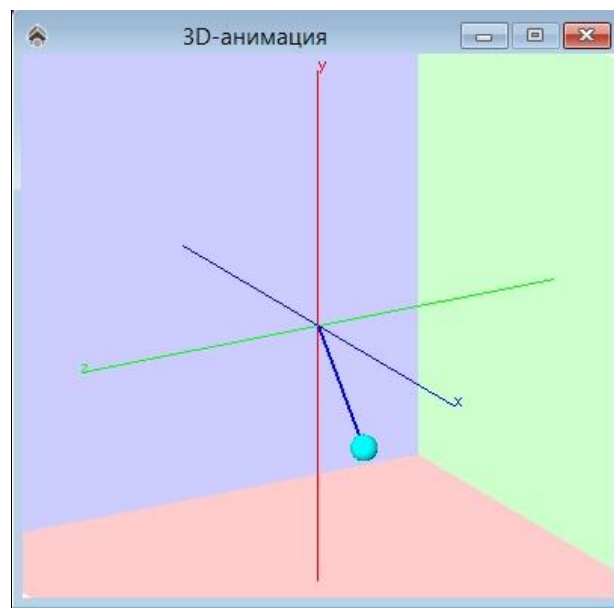




Рис. 4.23. 3D-анимация маятника

Двумерная анимация не теряет своего значения для пакетов визуального моделирования. Прежде всего, это касается разного рода панелей управления, которые по своей природе являются двумерными. Создадим 2D-анимационную панель управления маятником. Такая модель должна включать в себя ползунок для интерактивного изменения ускорения силы тяжести g и три индикатора, показывающие значения угла отклонения, угловой скорости и углового ускорения.

С помощью команды «**Окна/Новая 2D-анимация**» или нажатием кнопки  в окне визуальной модели создадим анимационное окно. Далее откроем с помощью кнопки  или команды «**Сервис/Стандартные 2D-компоненты**» панель стандартных 2D-компонентов (рис. 4.24).

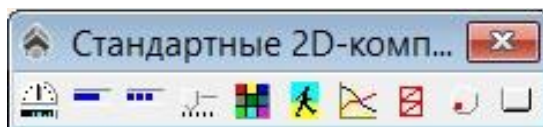


Рис. 4.24. Компоненты 2D анимации

Выберем на этой панели компоненту «**Ползунок**» и методом «drag-and-drop» переместим ее в левую часть окна анимации. Щелкнем на появившемся изображении ползунка правой клавишей мыши и с помощью команд всплывающего меню установим вертикальное расположение ползунка, режим показа числового значения и интервал значений 0 ... 99 (рис. 4.25).

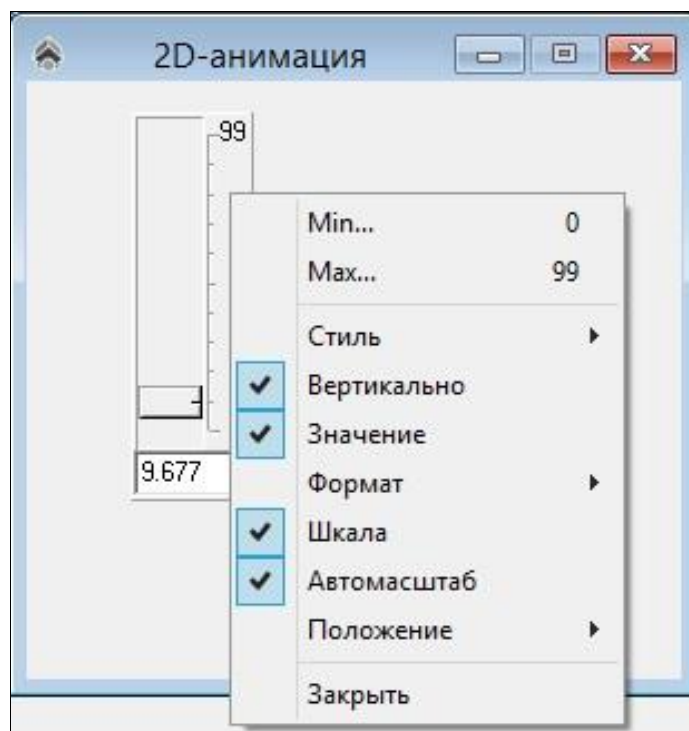


Рис. 4.25. Параметры компоненты «Ползунок»

С помощью мыши можно перемещать 2D-компоненту в пределах окна анимации, а также изменять размер компоненты. Щелкнем правой клавишей

мыши над компонентой и, выполнив команду «**Добавить надпись**» всплывающего меню, создадим пояснительную надпись для ползунка.

Надпись также можно перемещать с помощью мыши с прижатой клавишей. Теперь нужно сделать самое важное – связать ползунок с переменной модели. Делается это аналогично тому, как мы связывали переменные с параметрами примитивов в 3D-анимации: методом «drag-and-drop» перемещаем переменную g из окна переменных и помещаем ее на изображение ползунка (g нужно предварительно сделать переменной). Теперь ползунок будет отображать текущее значение переменной g (рис. 4.26).



Рис. 4.26. 2D-анимация для маятника

Используем три компонента «**Линейный индикатор сплошной**» для индикации значений переменных $Alpha$, $Alpha'$ и $Alpha''$ (рис. 4.26).

С помощью команды «**Заголовок окна**» всплывающего меню припишем панели заголовок «**Панель управления маятником**» и запретим режим редактирования, чтобы случайно не переместить анимационные компоненты.

Теперь мы можем, двигая ползунок, менять значение g и наблюдать реакцию маятника (лучше перед этим закрыть окна диаграмм, так как они сильно замедляют работу модели). Если резко уменьшить величину g в момент прохождения нижнего вертикального положения (максимум кинетической энергии), то маятник можно закрутить вокруг оси вращения (рис. 4.27).

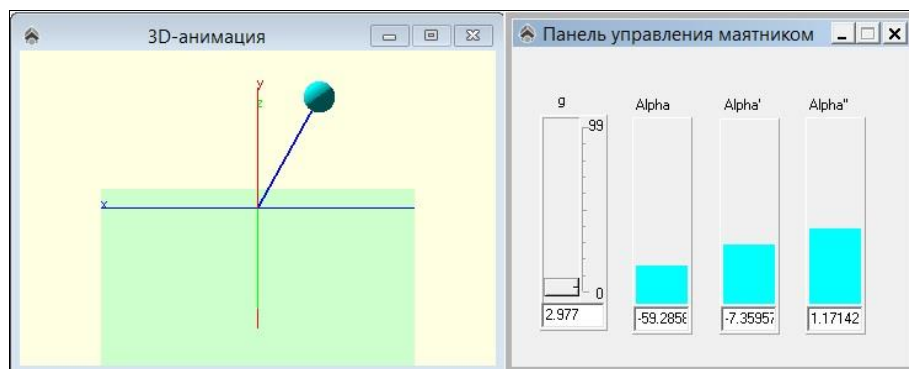



Рис. 4.27. 3D и 2D-анимации

4.3. ПРОЕКТ RMD «ГИБРИДНАЯ МОДЕЛЬ»

Новая модель – модель изолированной системы с несколькими качественными состояниями: **колебания, свободный полет, останов**. Каждое из состояний подчиняется своим законам. Такие модели называются **гибридными**.

В качестве примера гибридной системы рассмотрим модель обрывающегося маятника: это математический маятник, у которого в определенный момент связь между стержнем и материальной точкой исчезает или происходит отрыв материальной точки, после чего материальная точка начинает свободное движение в поле тяготения. Эту модель мы будем создавать на основе проекта «**Маятник.mvb**». С помощью команды «**Проект/Сохранить как**» сохраним проект «**Маятник.mvb**» как «**ОбрывМаятник.mvb**» в новой отдельной папке «**ОбрывМаятник**».

Поведение обрывающегося маятника включает в себя два качественно различных состояния, в которых он ведет себя как две различные непрерывные системы: математический маятник и свободный полет материальной точки. Математический маятник и его визуальная модель уже созданы. Оставим пока эту модель без изменений и займемся моделью полета материальной точки в поле тяготения. Для этого с помощью команды «**Модель/Преобразовать описание модели в класс**» главного меню или кнопки  на инструментальной панели преобразуем класс Model в класс «**МатемМаятник**», описание класса Model после этого инициализируется (рис. 4.28).

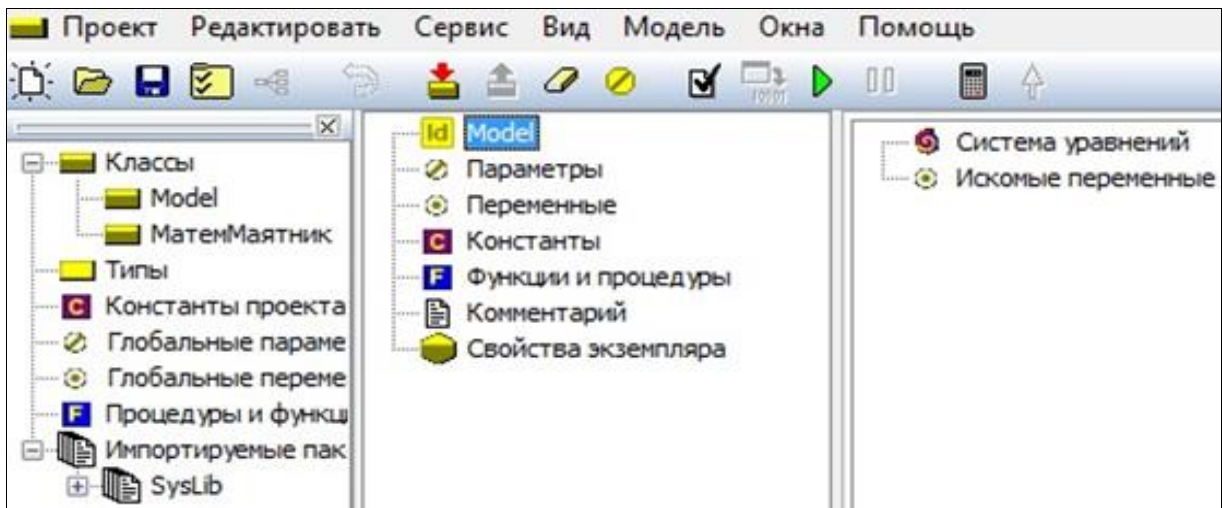


Рис. 4.28. Преобразование класса «Model»

На рис. 4.29 показана модель движения материальной точки в поле тяготения без сопротивления воздуха.

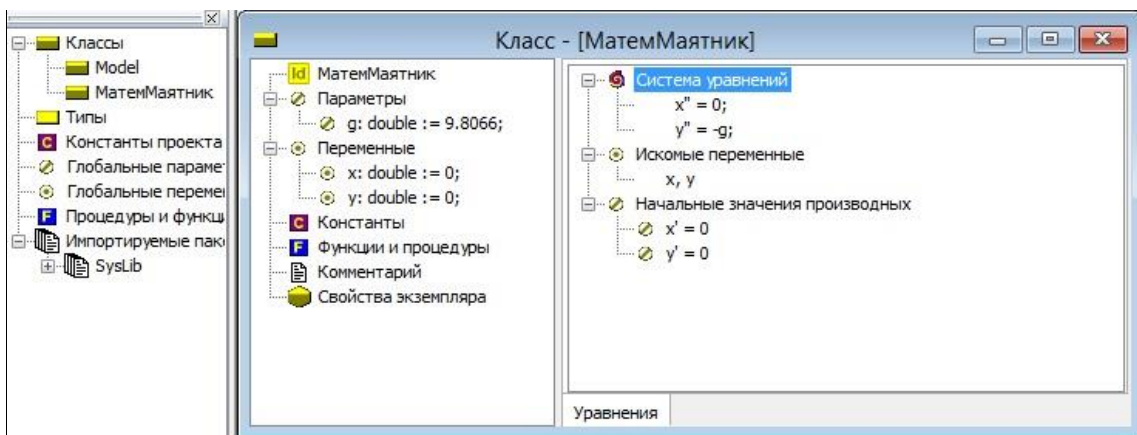


Рис. 4.29. Модель движения материальной точки

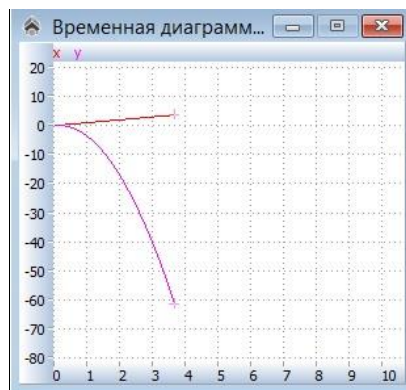


Рис. 4.30. Временная диаграмма

Создав и запустив визуальную модель, получим зависимость декартовых координат материальной точки от времени (рис. 4.30), свидетельствующую о правильности нашей модели.

Сохраним эту модель как непрерывный класс **«ДвижМатерТочки»**. Теперь у нас есть отлаженные модели поведения исследуемой системы в каждом из двух ее качественных состояний, и нужно определить в классе **«Model»** правила перехода из одного состояния в другое.

С помощью команды **«Преобразовать в/Гибридный объект»** всплывающего меню (рис. 4.31) превратим класс **«Model»** в гибридный класс.

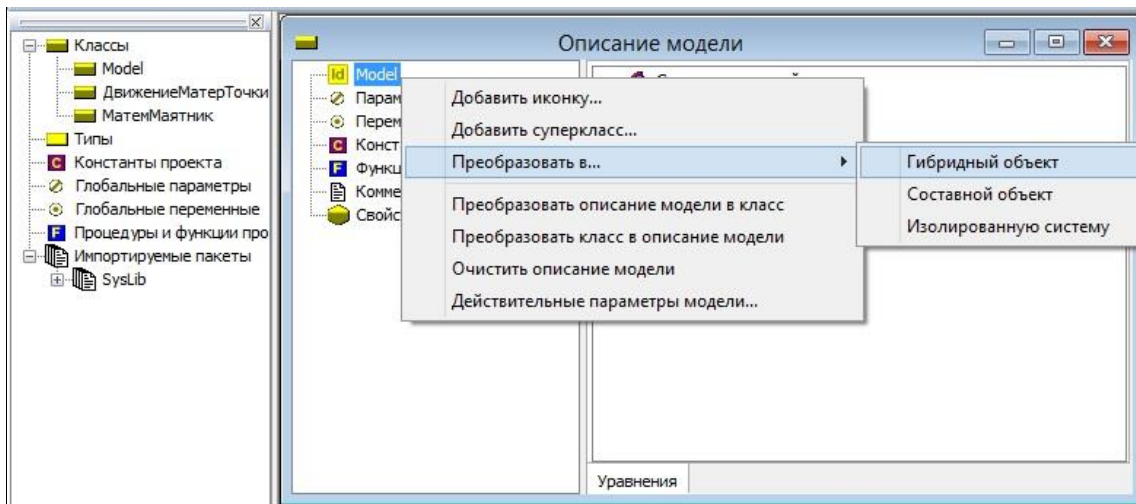


Рис. 4.31. Преобразование класса

В результате этого преобразования страничка **«Уравнения»** в редакторе поведения исчезнет и появится страничка **«Карта поведения»** (рис. 4.32).

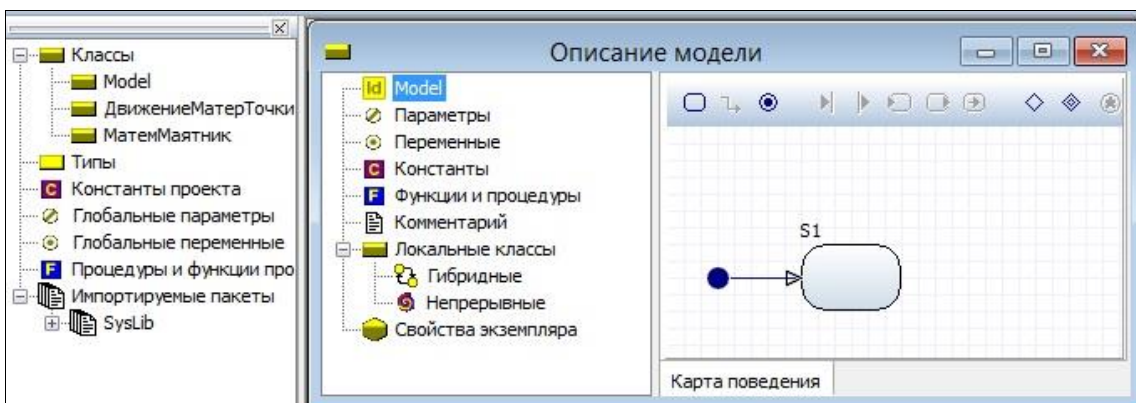


Рис. 4.32. Преобразованный класс

Карта поведения представляется в редакторе в графической форме (рис. 4.33).

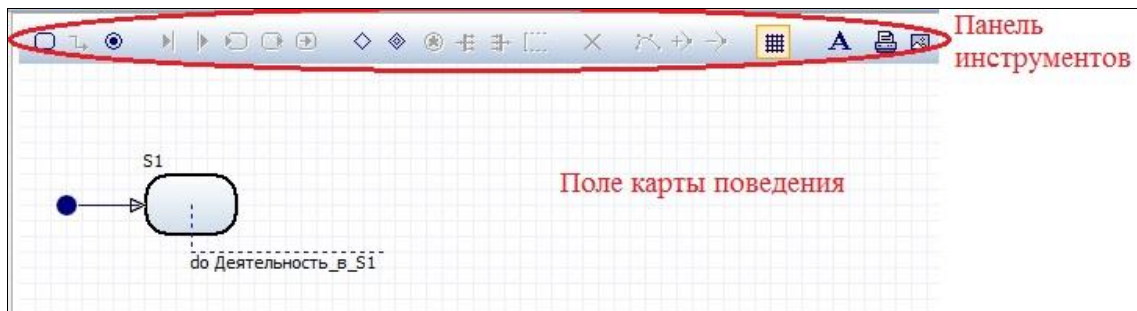


Рис. 4.33. Карта поведения

Основными элементами редактора карты поведения являются:

- панель инструментов с кнопками, обеспечивающими работу с элементами карты поведения;
- поле карты поведения, позволяющее создавать и редактировать карту поведения.

Панель инструментов содержит следующие кнопки:



– создает новое состояние.



– создает новый переход.



– создает конечное состояние.



– открывает окно «Условие срабатывания перехода».



– открывает окно текстового редактора для ввода мгновенных действий в переходе.



– открывает окно текстового редактора для ввода входных действий в состоянии.









– открывает окно текстового редактора для ввода выходных действий в состоянии.






– открывает окно «Внутренние переходы».



– создает новую точку ветвления.

-  – создает новую вероятностную точку ветвления.
-  – удаляет элемент карты поведения.
-  – отображает переход как сплайн – кривую.
-  – добавляет опорную точку в переходе.
-  – удаляет опорную точку в переходе.
-  – подключает/отключает изображение размерной сетки поля карты поведения.

поведения.

-  – вызывает окно «**Font**» для редактирования шрифта текста сносок.
-  – вызывает окно «**Print**» для установок печати.
-  – вызывает окно «**Save As**» для сохранения поля карты поведения.

Рассмотрим вариант модели, в которой разрушение стержня происходит при отклонении маятника на определенный угол при положительной угловой скорости. Для этого откройте созданный нами проект «**ОбрывМаятник**». Новая карта поведения по умолчанию включает в себя начальное состояние (обозначается закрашенным кружком), обычное состояние *S1* без деятельности и безусловный переход из начального состояния в *S1* (рис. 4.34).

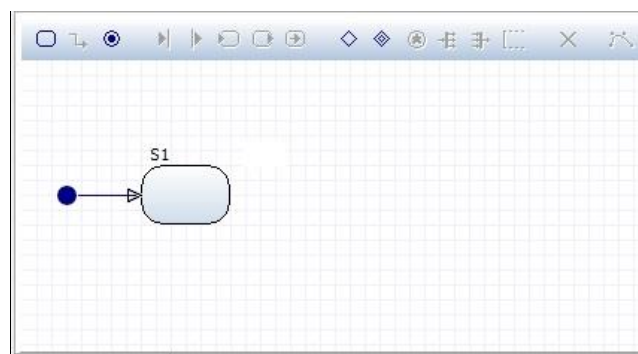


Рис. 4.34. Вид карты поведения по умолчанию

Карта поведения обрывающегося маятника должна содержать два обычных состояния – «**Колебания**» и «**Полет**» (рис. 4.35). Дважды щелкнув мышью

на имени «S1», перейдем в режим редактирования и заменим имя «S1» именем «Колебания» (рис. 4.35).

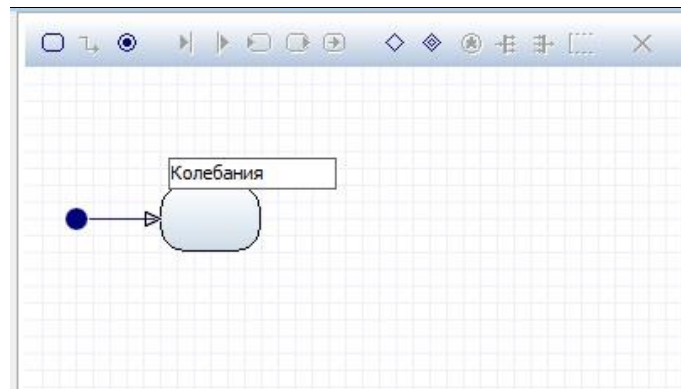




Рис. 4.35. Режим редактирования имени состояния

Добавим второе состояние. С помощью кнопки  на инструментальной панели редактора карты поведения создадим новое состояние и заменим его имя именем «Полет». Графический образ состояния можно перемещать с помощью мыши по карте поведения и изменять его размеры. С помощью кнопки  на инструментальной панели редактора карты поведения добавим конечное состояние. Далее добавим переходы из состояния «Колебания» в состояние «Полет» и из состояния «Полет» в конечное состояние (рис. 4.36).

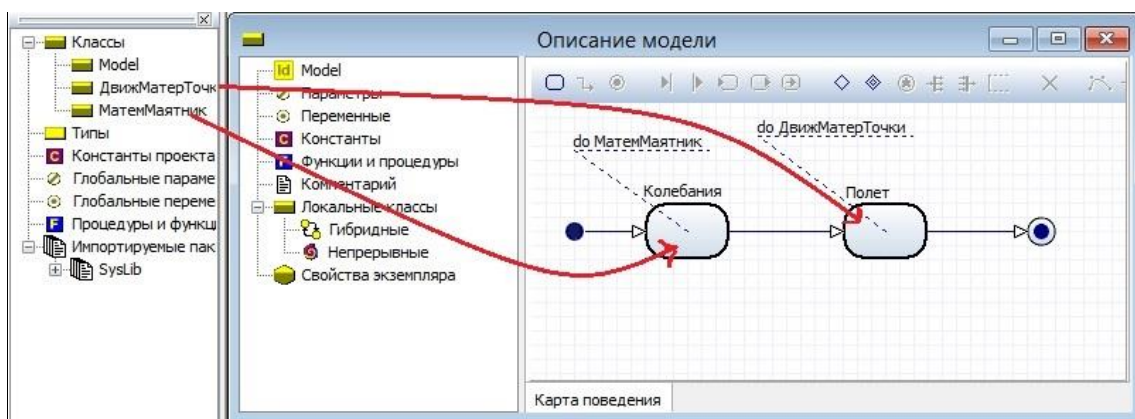


Рис. 4.36. Карта поведения обрывающегося маятника

Теперь припишем состояниям требуемое поведение: методом «drag-and-drop» переместим из менеджера проекта и поместим на изображения состояний «Колебания» и «Полет» классы «МатемМаятник» и «ДвижМатерТочки»

соответственно (на рис. 4.36 перемещение условно показано стрелками). Это означает, что экземпляры этих классов будут теперь являться деятельностями в соответствующих состояниях (рис. 4.36). Экземпляр деятельности создается при входе в состояние (когда состояние становится текущим) и уничтожается при выходе из него (после выполнения мгновенных действий на переходе, приведшем к смене поведения). Имя объекта – деятельности совпадает с именем состояния («Колебания», «Полет») которому он приписан.

Для завершения описания поведения маятника осталось указать условия срабатывания переходов и согласовать начальные условия деятельностей в состояниях карты поведения.

Введем в классе «**Model**» новые параметры и переменные (рис. 4.37–4.38):

– *Alpha_max* – угол отклонения маятника, при достижении которого с положительной угловой скоростью происходит обрыв.

– *Alpha_нач* – начальное отклонение маятника.

– *G* – ускорение силы тяжести.

– *L_маятн* – длина маятника.

– *Teta* – угол наклона вектора скорости материальной точки в особых точках траектории (обрыва и падения).

– *V* – линейная скорость материальной точки на траектории.

– *X* – горизонтальная координата материальной точки.

– *Y* – вертикальная координата материальной точки.

Нажмем правую кнопку мыши на изображении состояния «**Колебания**» и выполним команду «**Действительные значения параметров и переменных деятельности**» всплывающего меню. В появившемся диалоге с помощью двойного щелчка мыши изменим значения некоторых параметров объекта – деятельности в состоянии «**Колебания**» (рис. 4.37). Аналогично введем действительные значения параметров и действительные начальные значения переменных и производных в объекте «**Полет**» (рис. 4.38).

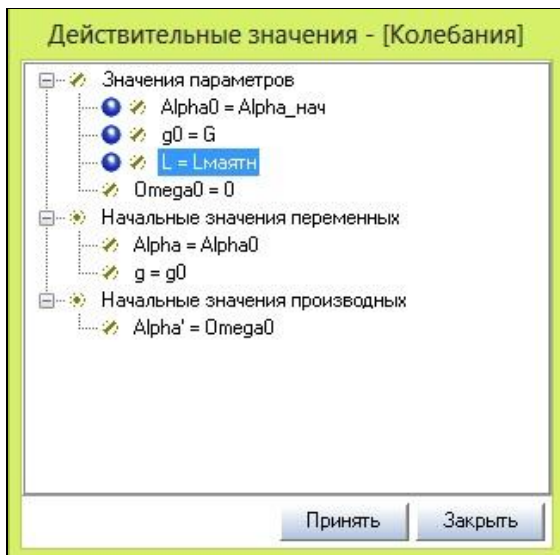


Рис. 4.37. Параметры объекта «Полет»

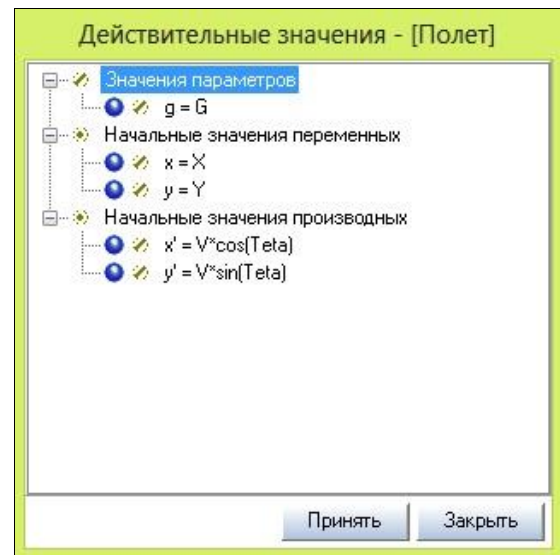


Рис. 4.38. Параметры объекта «Колебания»

Эти изменения отображаются и на карте поведения (рис. 4.39).

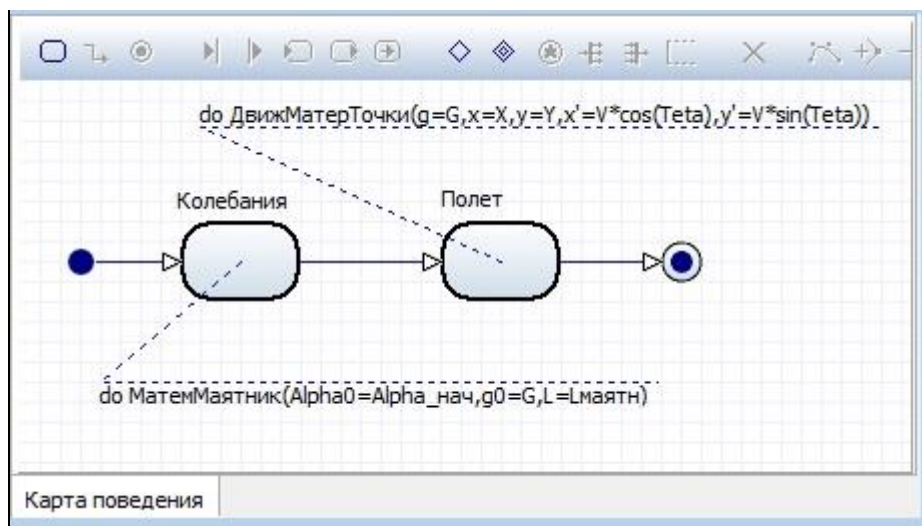



Рис. 4.39. Карта поведения с изменениями

Выделим (сделаем доступным для редактирования) переход из состояния «Колебания» в состояние «Полет». С помощью кнопки  на инструментальной панели редактора карты поведения или команды «Условие срабатывания перехода» всплывающего меню вызовем диалог редактирования условия срабатывания перехода (рис. 4.40).

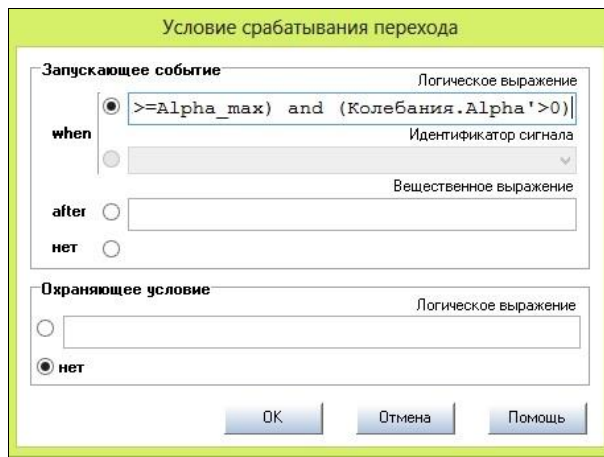



Рис. 4.40. Условия срабатывания перехода

Запускающим событием для этого перехода является изменение значения с FALSE на TRUE логического выражения (**Колебания.Alpha** \geq **Alpha_max**) and (**Колебания.Alpha**' $>$ **0**). Переменные объекта-деятельности видимы в условии и действиях исходящего перехода и указываются с именем объекта в качестве префикса.

Теперь нужно определить мгновенные действия этого перехода, которые вычисляли бы значения переменных *Teta*, *V*, *x*, *y*, необходимые для согласования начальных условий в состоянии «**Полет**».

С помощью кнопки  верхней инструментальной панели редактора карты поведения или команды «**Действия перехода**» всплывающего меню вызовем редактор мгновенных действий, в котором введем последовательность операторов присваивания (рис. 4.41).

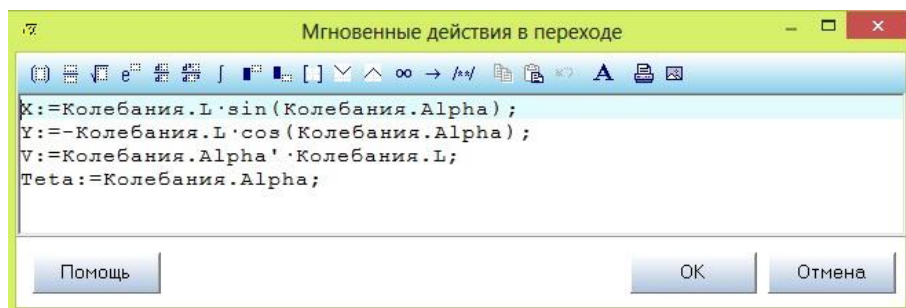


Рис. 4.41. Мгновенные действия в переходе

В мгновенных действиях могут использоваться также и другие алгоритмические операторы (условный оператор, оператор цикла и другие).

Аналогично введем условие срабатывания и мгновенные действия для перехода из состояния «**Полет**» в конечное состояние. Запускающее событие для этого перехода определяются выражением **Полет.y** <= 2*L, а мгновенные действия показаны на рис. 4.42.

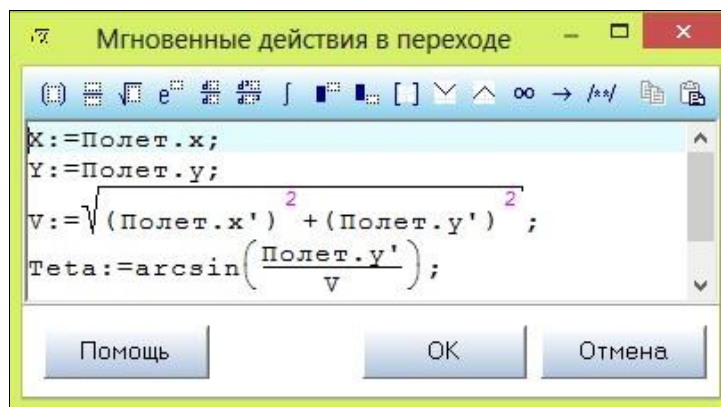


Рис. 4.42. Мгновенные действия в переходе 2

На рис. 4.43 показано начальное состояние главного окна визуальной модели обрывающегося маятника при первом запуске. Модель находится в точке останова. Для гибридной модели по умолчанию открываются окно переменных и окно карты поведения.

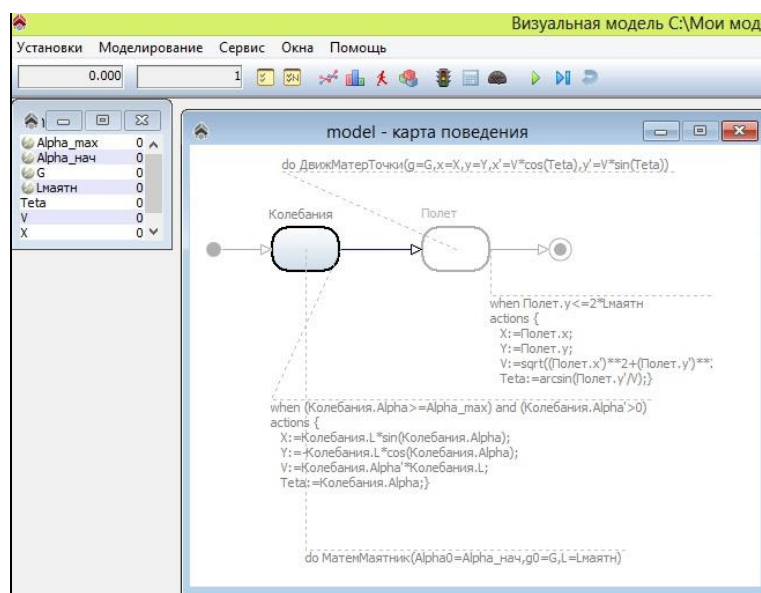


Рис. 4.43. Окно карты поведения

Теперь возникает вопрос. Как наблюдать изменение переменных в объектах, приписанных состояниям, если эти объекты существуют только в определенные интервалы времени? Например, в состоянии модели по рис. 4.43 объект «**Колебания**» существует, а объект «**Полет**» не существует.

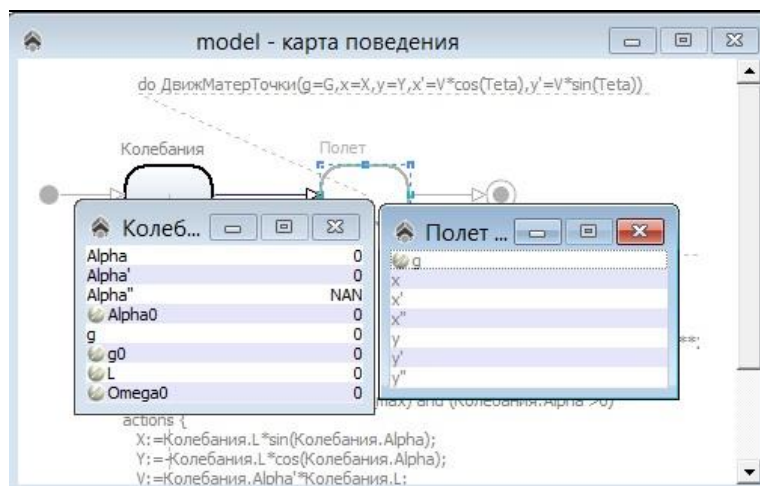


Рис. 4.44. Переменные в объектах-действиях

Нажмем правую кнопку мыши на изображении состояния «**Полет**» и в появившемся всплывающем меню выполним команду «**Переменные**». В появившемся окне переменных имена переменных изображены серым цветом, а значения отсутствуют (рис. 4.44), поскольку этого объекта пока еще не существует. Если же мы выполним те же самые действия для состояния «**Колебания**», то получим обычное окно переменных с их текущими значениями (рис. 4.44), так как состояние «**Колебания**» является текущим и его объект существует.

Для объектов-действий можно вводить виртуальные переменные. Добавим виртуальные переменные x и y для состояния колебания. Переменные объектов-действий можно отображать на диаграммах. Графики для этих переменных будут рисоваться только для тех временных интервалов, на которых соответствующий объект-деятельность существует. Создадим временную диаграмму, на которой отображаются переменные «**Колебания.x**», «**Колебания.y**», «**Полет.x**» и «**Полет.y**».

Создадим трехмерную анимацию обрывающегося маятника. Нам придется ввести 3D-объекты: «**line1**» и «**sphere2**» для изображения колебаний маятника, «**sphere3**» для изображения летящей материальной точки (рис. 4.45).

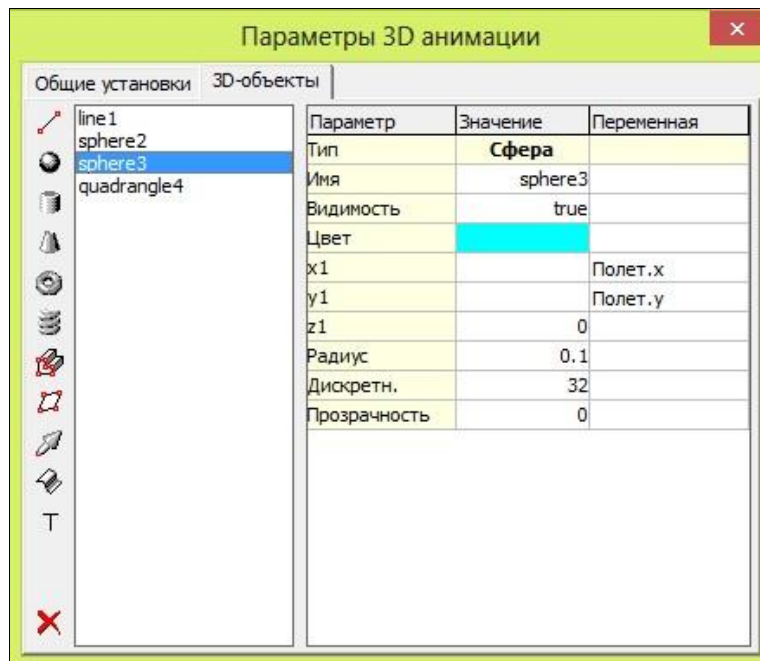


Рис. 4.45. Параметры 3D-анимации

Параметры 3D-объектов «**line1**» и «**sphere2**» связаны с переменными «**Колебания.x**» и «**Колебания.y**» и потому эти 3D-объекты видимы только в состоянии «**Колебания**». Параметры 3D-объекта «**sphere3**» задаются переменными «**Полет.x**» и «**Полет.y**», поэтому этот 3D-объект видим только в состоянии «**Полет**». Радиусы и цвет «**sphere2**» и «**sphere3**» устанавливаем одинаковыми.

В результате суперпозиции этих четырех изображений получаем динамическую трехмерную картинку обрывающегося маятника, шарик исчезает (рис. 4.46).

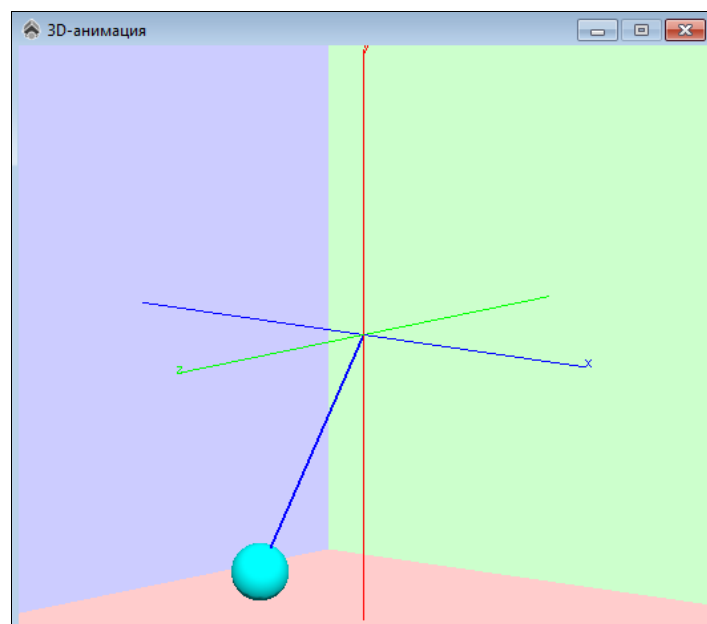


Рис. 4.46. 3D-анимация

4.4. ИНСТРУМЕНТАЛЬНАЯ СИСТЕМА МОДЕЛИРОВАНИЯ ANYLOGIC

Система AnyLogic, разработанная компанией XJ Technologies (Россия 1999г., сейчас это фирма AnyLogic), это среда компьютерного моделирования общего назначения. Это комплексный инструмент, охватывающий основные в настоящее время направления моделирования: дискретно-событийное моделирование, моделирование системной динамики и агентное моделирование. Использование AnyLogic дает возможность оценить эффект принимаемых решений в сложных системах реального мира.

В AnyLogic разработчик может гибко использовать различные уровни абстрагирования и различные стили и концепции и смешивать их при создании одной и той же модели (рис. 4.47).

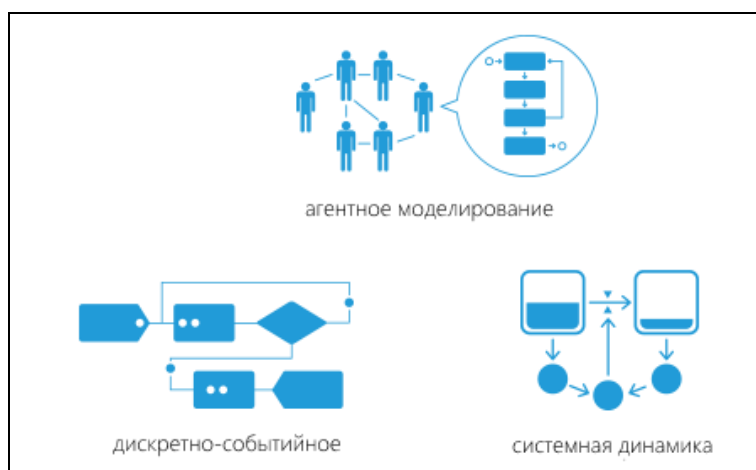


Рис. 4.47. Парадигмы моделирования в AnyLogic

AnyLogic, так же как и RMD, использует объектно-ориентированный подход к представлению сложных систем. Этот подход позволяет простым и естественным образом организовать и представить структуру сложной системы с помощью иерархии абстракций. Например, на некотором уровне абстракции автомобиль можно считать неким единым объектом. Но более детально его можно представить как совокупность взаимодействующих подсистем: двигателя, рулевого управления, тормозной системы и т.п. Каждая из этих подсистем может быть представлена, если это необходимо, своей структурой взаимодействующих подсистем.

Именно такую иерархию абстракций позволяет создать AnyLogic при разработке моделей. Всю модель можно рассматривать как единый объект **«root»**. При детальном рассмотрении видно, что этот объект может содержать, к примеру, два экземпляра класса MyClass с именами **myClass** и **myClass1**. Сам класс **«MyClass»** может иметь свое сложное строение, которое будет скрыто в классе **«Root»**. Такая иерархия структуры может быть произвольной глубины.

При построении модели в среде AnyLogic не предусмотрена возможность использования никаких других средств, кроме средств визуальной разработки (введения состояний и переходов стейтчарта, введения пиктограмм переменных и т.п.), задания численных значений параметров, аналитических записей соотношений переменных и аналитических записей условий наступления событий. Основной парадигмой, принятой в AnyLogic при разработке моделей, является визуальное проектирование – построение с помощью графических объектов и пиктограмм иерархий структуры и поведения активных объектов.

Решения, которые позволяет принять AnyLogic-моделирование:

1. Управление сложными современными производственными процессами и финансовыми операциями требует оперативности, планирования и оценки результата. AnyLogic помогает оценить последствия принятия нескольких альтернативных решений и выбрать лучшее; использовать прогностическую модель для оперативного, среднесрочного или стратегического планирования; оценить работу процесса «в виртуальном мире», проверить изменения до их реализации.

2. По построенной программной модели, имитирующей работу реальной системы, возможно исследовать производительность системы, ее «узкие места», оценить влияние параметров (например, оценить, куда направить ограниченный объем капиталовложений).

3. Проводить эксперименты «что, если...», оценивать альтернативные варианты и выбирать технологию, дающую наилучшие результаты.

4. Проводить моделирование и компьютерную имитацию функционирования сложных систем совместно с их визуализацией, что становится повседневным современным средством поддержки и обоснования принятия решений.

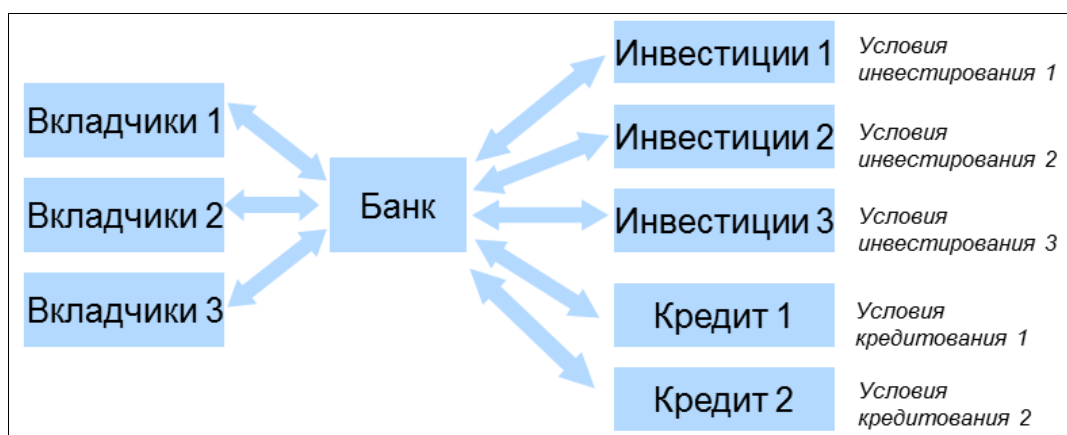


Рис. 4.48. Модель финансовых потоков банка

Например, представленная на рис. 4.48 схема, по которой будет построена модель банка, позволяет:

- оперативно оценить положение банка в будущем, учитывая сроки погашение кредитов и внесение вкладов, произвести оценку разрывов ликвидности;
- определить наилучшую схему развития банка, учитывая прибыльность и устойчивость к рискам;
- оценить последствия рисков непогашения кредитов или кризиса доверия к депозитам;
- моделировать финансовые потоки, следовательно, улучшить управляемость, получить точные и своевременные оценки, выявить перспективные направления и оперативно оценить разрывы ликвидности;
- моделировать фондовые рынки и поведение инвесторов на фондовых рынках по известным моделям составить инвестиционную стратегию, повысить отдачу от использования капитала и оценить общую ситуацию на рынке;
- оперативно оценить положение банка в будущем, учитывая сроки погашение кредитов и внесение вкладов, произвести оценку разрывов ликвидности;
- определить наилучшую схему развития банка, учитывая прибыльность и устойчивость к рискам;
- оценить последствия рисков непогашения кредитов или шока (кризиса доверия) депозитов.

Для моделирования систем с дискретными событиями (рис. 4.49) существует множество пакетов, облегчающих разработку дискретно-событийных

моделей и проведение экспериментов с моделями в этой традиционной области моделирования. В первую очередь, это GPSS (General Purpose Simulation System), который стал революцией более 50 лет назад, задав парадигму моделирования в этой области в виде блоков и транзактов. Транзакты отображают динамические объекты моделирования (заявки), а блоки – объекты, обрабатывающие эти заявки. Большинство других инструментов моделирования (Arena, Extend, ProModel, SimProcess и др.) также используют эту парадигму.

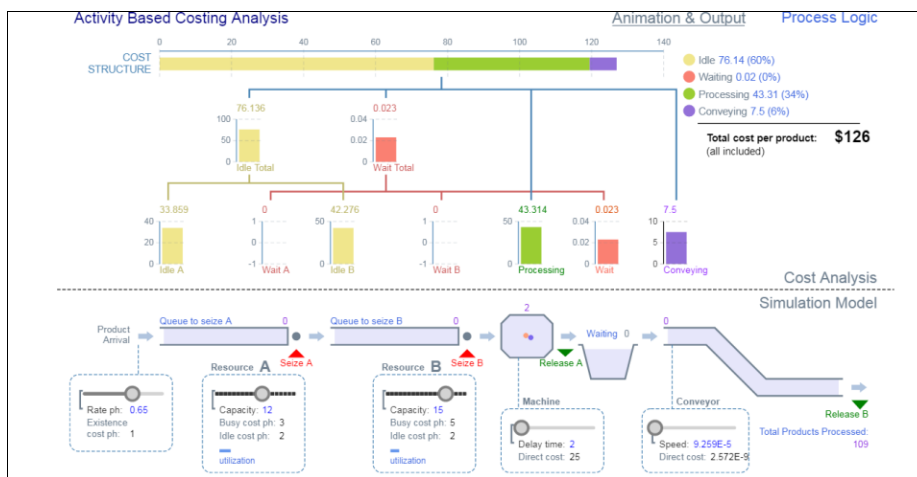


Рис. 4.49. AnyLogic модель дискретных событий (производство)

Моделирование системной динамики – это методология изучения и моделирования систем, характеризующихся циклами обратных связей в сложных взаимных причинных зависимостях их параметров (рис. 4.50).

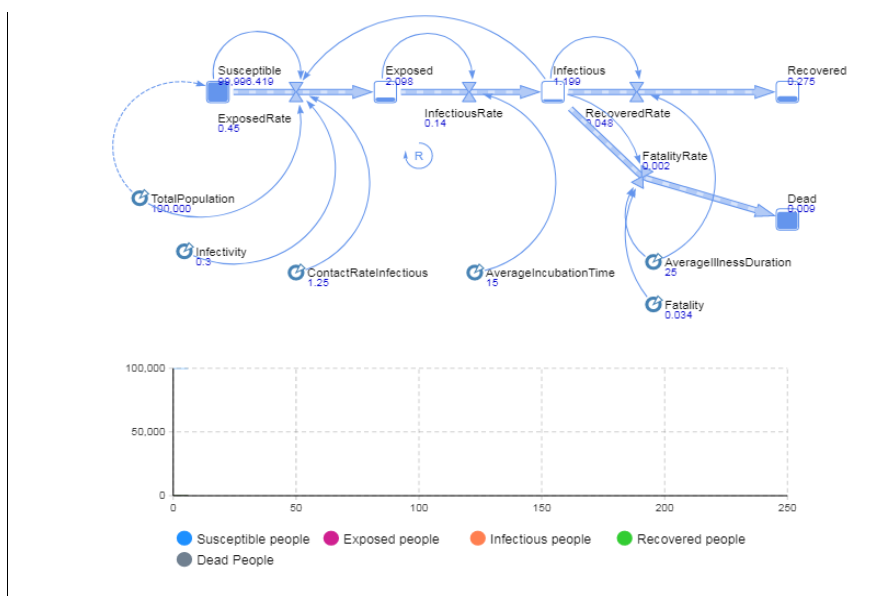


Рис. 4.50. AnyLogic модель системной динамики

Математически эти системы описываются системами дифференциальных уравнений, приведенных к форме Коши. Эти модели применяются для корпоративного планирования и анализа политики управления корпорацией, политик управления социальными и экономическими системами, в экологии и т.п. Джей Форрестер – американский инженер и системолог, разработчик теории системной динамики – в 1970-х гг. разработал модели «Мир-1» и «Мир-2», нацеленные на выработку сценариев развития всего человечества в его взаимоотношении с биосферой. Так родилось первое поколение компьютерных моделей, предназначенных для исследования долгосрочных тенденций мирового развития. Он опубликовал книгу «Мировая динамика», в которой обобщил свой вклад в создание первых компьютерных моделей, анализирующих глобальную систему. Он заложил методологические принципы системной динамики с использованием графического представления причинных зависимостей переменных, в виде так называемых «stock and flow diagrams», которые и сейчас являются основой инструментов моделирования, конкурирующих на рынке: VisSim, PowerSim, Stella, ModelMaker и др.

Моделирование динамических систем – это область моделирования систем управления, физических, механических систем, систем обработки сигналов и т.п. Широкое применение в этой области имеет пакет моделирования Simulink, являющийся составной частью пакета Matlab. Пакет имеет библиотеку предопределенных блоков, из которых можно методом «drag-and-drop» строить блочную структуру, аналогичную блочной структуре моделей, когда-то давно, лет 60 назад, набираемых для их решения на аналоговых вычислительных машинах из интеграторов, усилителей, сумматоров, источников сигналов и т.п.

Агентное моделирование. Под агентом понимается активный объект, обладающий поведением и имеющий возможность взаимодействия с другими агентами и со средой. Многоагентное моделирование позволяет вывести характеристики целого (множества агентов) из совокупности локальных поведений и характеристик отдельных активных элементов целого, распределенных в среде (рис. 4.51).

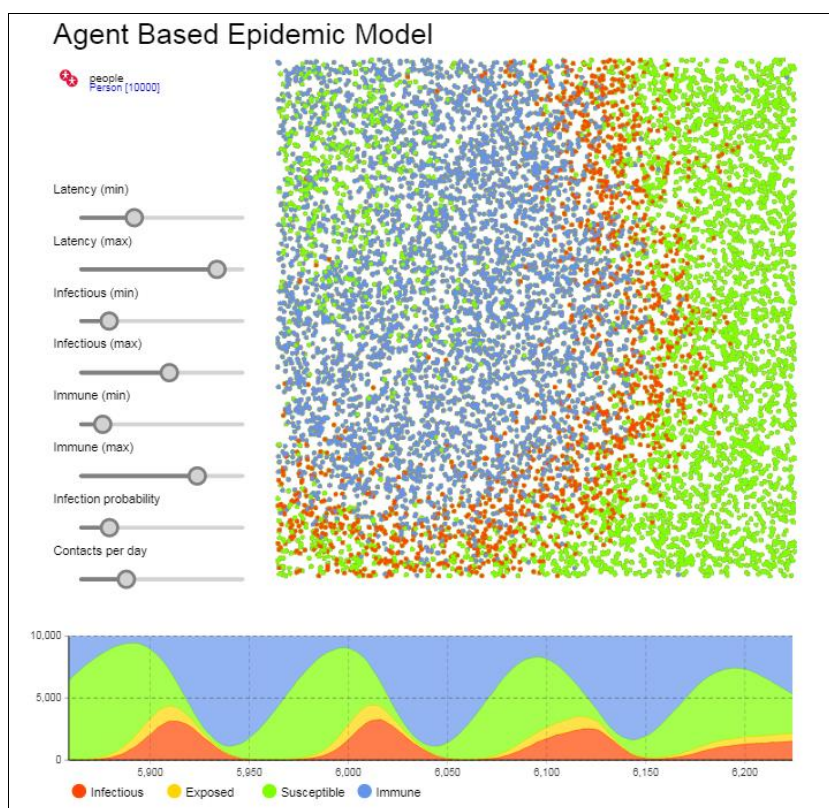


Рис. 4.51. AnyLogic-агентная модель развития эпидемии

Моделирование многоагентных систем используется при анализе социальных процессов, процессов урбанизации и даже при исследовании рынка в анализе предпочтений различных социальных групп или корпораций, выступающих как агенты со своим поведением.

- Если модель предполагает много индивидуальных объектов, используйте **агентное моделирование**.
- Если вы владеете информацией только о глобальных зависимостях, используйте **системную динамику**.
- Если система может быть описана как процесс, используйте **дискретно-событийное** моделирование.
- Если системе присущи все эти особенности, комбинируйте различные методы.

На рис. 4.52 показано, как производство, распределение и рынок могут быть совмещены в одной модели с применением трех подходов одновременно.

Дискретно-событийная модель описывает процессы на каждом складе. На уровне распределительной сети склады действуют как независимые агенты.

И, наконец, управляющий системой рынок моделируется с помощью системной динамики. Модель детально отражает все элементы и процессы этой системы.



Рис. 4.52. Применение различных методов моделирования

С AnyLogic разработчик не ограничен одним методом моделирования (рис. 4.53). Используя подходящий метод или их комбинацию, он может создать оптимальную для решения конкретной проблемы имитационную модель.



Рис. 4.53. Методы моделирования, реализуемые в AnyLogic

Таким образом, идеи и методы, направленные на управление сложностью, выработанные в последние десятилетия в области создания программных систем, позволяют разработчикам моделей в среде AnyLogic организовать

мышление, структурировать разработку и, в конечном счете, упростить и ускорить создание моделей.

Другой базовой концепцией AnyLogic является представление модели как набора взаимодействующих параллельно функционирующих активностей. Такой подход к моделированию интуитивно очень понятен и естественен во многих приложениях, поскольку системы реальной жизни состоят из совокупности активностей, взаимодействующих с другими объектами. Активный объект AnyLogic – это объект со своим собственным функционированием, взаимодействующий с окружением. Он может включать в себя любое количество экземпляров других активных объектов. Активные объекты могут динамически рождаться и исчезать в соответствии с законами функционирования системы. Так могут моделироваться социальные группы, холдинги компаний, транспортные системы и т.п.

Графическая среда моделирования AnyLogic поддерживает проектирование, разработку, документирование модели, выполнение компьютерных экспериментов с моделью, включая различные виды анализа: от анализа чувствительности до оптимизации параметров модели относительно некоторого критерия.

В результате AnyLogic не ограничивает пользователя одной-единственной парадигмой моделирования, что является характерным фактически для всех инструментов моделирования, существующих сегодня. В AnyLogic разработчик может гибко использовать различные уровни абстрагирования, различные стили и концепции, строить модели в рамках той или иной парадигмы и смешивать их при создании одной и той же модели (рис. 4.54), использовать ранее разработанные модули, собранные в библиотеки, дополнять и строить свои собственные библиотеки модулей. При разработке модели на AnyLogic можно использовать концепции и средства из нескольких «классических» областей моделирования. В агентной модели использовать методы системной динамики для представления изменений состояния среды или в непрерывной модели динамической системы учесть дискретные события, например, анализ IT-инфраструктуры компании (анализ производительности серверов, узких мест локальной сети и т.п.).

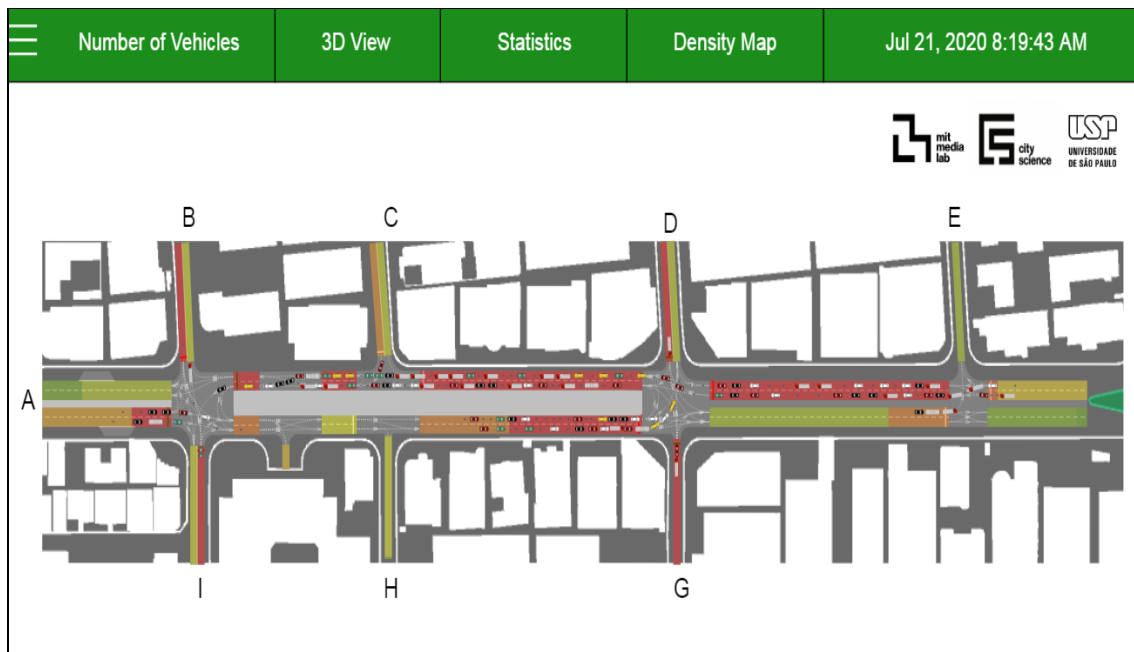


Рис. 4.54. AnyLogic модель дорожного движения

Он легко производится с помощью методов дискретного событийного моделирования, имеет немного пользы, если в модели не отражено влияние возможных изменений параметров этой инфраструктуры на бизнес-процессы и, в конечном счете, на прибыль компании, а такая связь в модели не может быть реализована только средствами дискретно-событийного моделирования. В AnyLogic легко строятся подобные модели с требуемым уровнем адекватности, позволяющие ответить на многие вопросы, интересующие исследователя. Богатые возможности анимации и визуального представления результатов в процессе работы модели позволяют понять суть процессов, происходящих в моделируемой системе, упростить отладку модели. Удобный интерфейс и многочисленные средства поддержки разработки моделей в AnyLogic делают не только использование, но и создание компьютерных имитационных моделей в этой среде моделирования доступным даже для начинающих.

В AnyLogic существуют две фазы имитационного моделирования: разработка модели и ее анализ. Разработка модели выполняется в среде редактора AnyLogic, анализ модели происходит в среде исполнения. В каждой фазе существуют свои средства управления. Переход из одной фазы в другую производится очень легко. Можно многократно использовать переход между фазами редактирования и исполнения модели при разработке модели.

В AnyLogic основным структурным блоком при создании моделей являются классы активных объектов. Использование активных объектов является

естественным средством структуризации модели сложных систем: мир состоит из множества параллельно функционирующих и взаимодействующих между собой сущностей. Различные типы этих сущностей и представляют разные активные объекты.

Чтобы создать модель AnyLogic, нужно создать классы активных объектов (или использовать объекты библиотек AnyLogic). Определение активного объекта задает шаблон, и отдельные объекты, построенные в соответствии с этим шаблоном (экземпляры активного объекта), могут использоваться затем как элементы других активных объектов. В любой класс могут быть включены несколько экземпляров других классов, которые могут различаться своими параметрами. Всегда один класс в модели является корневым. Для него в модели AnyLogic порождается один экземпляр с предопределенным именем «root», он и запускается исполнительной системой AnyLogic на выполнение. Имя класса корневого активного объекта можно менять в окне его свойств.

Каждый активный объект имеет структуру (совокупность включенных в него активных объектов и их связи), а также поведение, определяемое совокупностью переменных, параметров, стейтчартов и т.п. Каждый экземпляр активного объекта в работающей модели имеет свое собственное поведение. Он может иметь свои значения параметров, он функционирует независимо от других объектов, взаимодействуя с ними и с внешней средой.

При построении модели в среде AnyLogic не предусмотрена возможность использования никаких других средств, кроме средств визуальной разработки (введения состояний и переходов стейтчарта, введения пиктограмм переменных и т.п.), задания численных значений параметров, аналитических записей соотношений переменных и аналитических записей условий наступления событий. Основной парадигмой, принятой в AnyLogic при разработке моделей, является визуальное проектирование – построение с помощью графических объектов и пиктограмм иерархий структуры и поведения активных объектов.

Основным средством спецификации поведения объектов в AnyLogic являются переменные, таймеры и стейтчарты. Переменные отражают изменяющиеся характеристики объекта. Таймеры можно взводить на определенный интервал времени и по окончании этого интервала выполнять заданное действие. Стейтчарты позволяют визуально представить поведение объекта во времени под воздействием событий или условий. Стейтчарты состоят из графического изображения состояний и переходов между ними. Любая сложная логика поведения

объектов модели в AnyLogic может быть выражена с помощью комбинации стейтчартов, дифференциальных и алгебраических уравнений, переменных, таймеров и программного кода на Java. Алгебраические и дифференциальные уравнения, как и логические выражения, записываются в AnyLogic аналитически.

Интерпретация любого числа параллельно протекающих процессов в модели AnyLogic скрыта от пользователя. Никаких календарей событий разработчик модели на AnyLogic не ведет, отслеживание событий во всех процессах, определенных в модели, выполняется системой автоматически.

Понятие модельного времени является базовым в системах имитационного моделирования. **Модельное время** – это условное логическое время, в единицах которого определено поведение всех объектов модели.

В моделях AnyLogic модельное время может изменяться либо непрерывно, если поведение объектов описывается дифференциальными уравнениями, либо дискретно, переключаясь от момента наступления одного события к моменту наступления следующего события, если в модели присутствуют только дискретные события. Моменты наступления всех планируемых событий в дискретной модели исполнительная система хранит в так называемом календаре событий, выбирая оттуда наиболее раннее событие для выполнения связанных с ним действий. Значение текущего времени в моделях AnyLogic может быть получено обращением к функции «getTime ()».

Единицу модельного времени разработчик модели может интерпретировать как любой отрезок времени: секунду, минуту, час или год. Важно только, чтобы все процессы, зависящие от времени, были выражены в одних и тех же единицах. При моделировании физических процессов все параметры и уравнения должны быть выражены в одной и той же системе физических величин. Например, все физические величины выражаются в системе СИ, в которой единица времени – секунда, единица длины – метр, единица массы – килограмм.

Интерпретация модели выполняется на компьютере. **Физическое время**, затрачиваемое процессором на имитацию действий, которые должны выполняться в модели в течение одной единицы модельного времени, зависит от многих факторов. Поэтому единица физического и единица модельного времени не совпадают.

В AnyLogic приняты два режима выполнения моделей: режим виртуального времени и режим реального времени. В режиме виртуального времени

процессор работает с максимальной скоростью без привязки к физическому времени. Этот режим используется для факторного анализа модели, набора статистики, оптимизации параметров модели и т.п.

В режиме реального времени пользователь задает связь модельного времени с физическим временем, то есть устанавливает ограничение на скорость работы процессора при выполнении модели. В этом режиме задается количество единиц модельного времени, которые должны выполняться процессором в одну секунду. Обычно данный режим включается для того, чтобы визуально представить функционирование системы в реальном темпе наступления событий, проникнуть в суть процессов, происходящих в модели.

Удобные средства разработки анимационного представления модели в AnyLogic позволяют представить функционирование моделируемой системы в живой форме динамической анимации, что позволяет «увидеть» поведение сложной системы.

Средства анимации позволяют пользователю легко создать виртуальный мир (совокупность графических образов, ожившую мнемосхему и т.п.), управляемый динамическими параметрами модели по законам, определенным пользователем с помощью уравнений и логики моделируемых объектов. Визуальное представление поведения системы помогает пользователю проникнуть в суть процессов, происходящих в системе.

Система AnyLogic предоставляет пользователю возможность изменять параметры модели в ходе ее функционирования. Поэтому окно анимации можно назвать «стендом» для проведения компьютерного эксперимента (рис. 4.55).

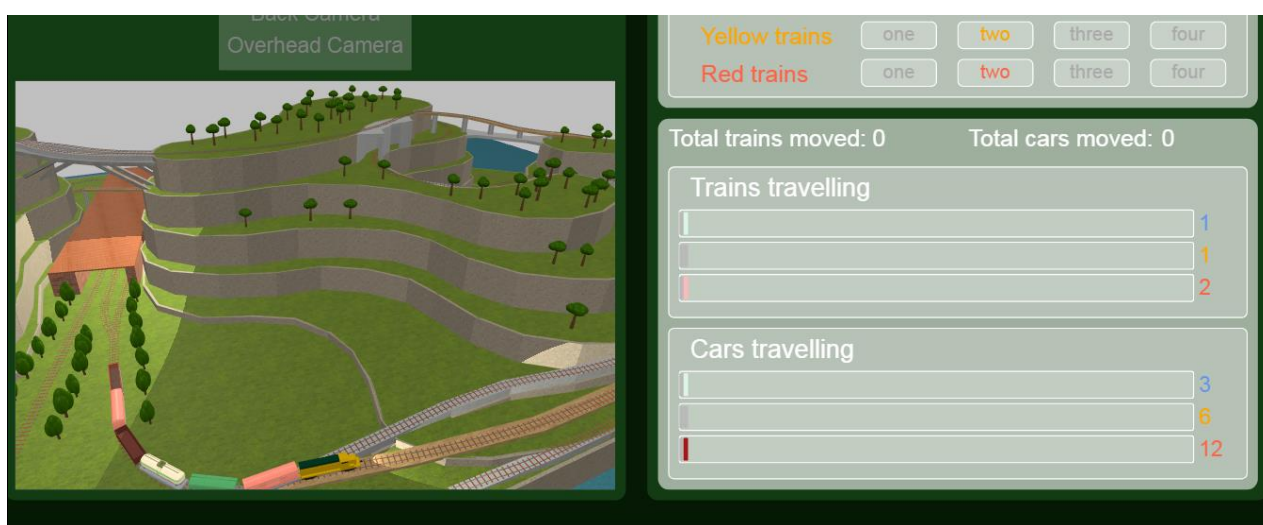


Рис. 4.55. Анимация в AnyLogic

4.5. ANYLOGIC-МОДЕЛИ СИСТЕМНОЙ ДИНАМИКИ

AnyLogic является инструментом имитационного моделирования, основанным на новых, нетрадиционных принципах. Вместо того чтобы долго рассказывать об особенностях этого инструмента, мы сразу начнем с изучения простейшей модели, уже разработанной в AnyLogic, чтобы составить первое представление о структуре инструмента и его функциях на двух фазах работы с моделью: разработка модели и ее анализ. Затем мы рассмотрим, как основные концепции, на которых базируется AnyLogic, используются на этих двух фазах.

Система AnyLogic проектировалась для работы в ОС Microsoft Windows, поэтому при работе в среде AnyLogic используются стандартные приемы и операции данной операционной системы.

Данные лабораторные работы написаны под версию **AnyLogic 8.5.2**.

При запуске AnyLogic отображается стартовая страница (рис. 4.56).

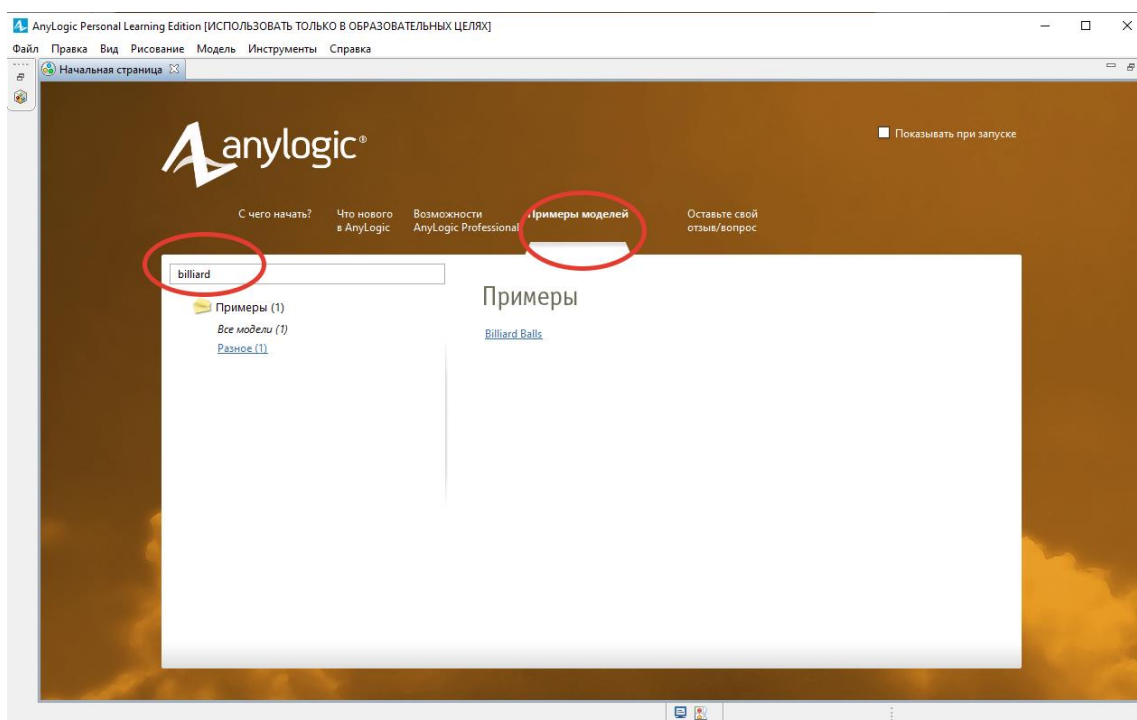


Рис. 4.56. Стартовая страница AnyLogic

Со стартовой страницы можно создать новый проект, открыть проект, с которым недавно работали, или открыть один из уже разработанных примеров моделей AnyLogic.

С помощью поиска на стартовой странице (вкладка «Примеры моделей») найдите модель-пример «Billiard Balls». Это имитационная модель движения бильярдных шаров. На экране появится следующее окно – рис. 4.57.

AnyLogic при открытии проекта всегда открывает среду разработки проекта – графический редактор модели. Рис. 4.57 показывает основные составляющие пользовательского интерфейса этого редактора. Рассмотрим их поочередно.

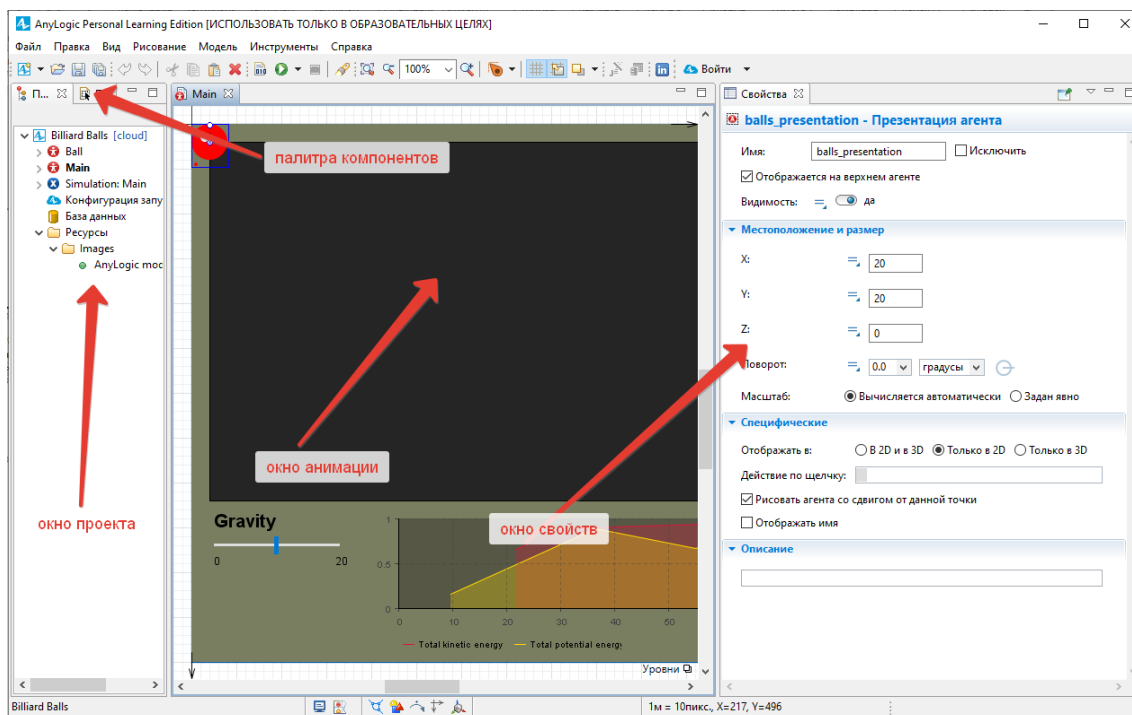


Рис. 4.57. Окна графического редактора модели

Окно «Проекты» обеспечивает навигацию по элементам проекта (рис. 4.58) для модели «Billiard Balls».

Проект всегда организуется иерархически, поэтому он отображается в виде дерева. Сам проект (**Billiard Balls**) образует «корень дерева» рабочего проекта, классы активных объектов и сообщений. На следующем уровне в классы активных объектов могут быть включены анимация, ресурсы, параметры симуляции, функции и т.д. Активный объект является основным структурным элементом модели в AnyLogic. Активным объектом называется сущность, которая **инкапсулирует** (включает в себя) данные (атрибуты объекта), функции (методы) и поведение как единое целое. Активный объект строится как класс,

который может включать в качестве составных элементов экземпляры других классов активных объектов.

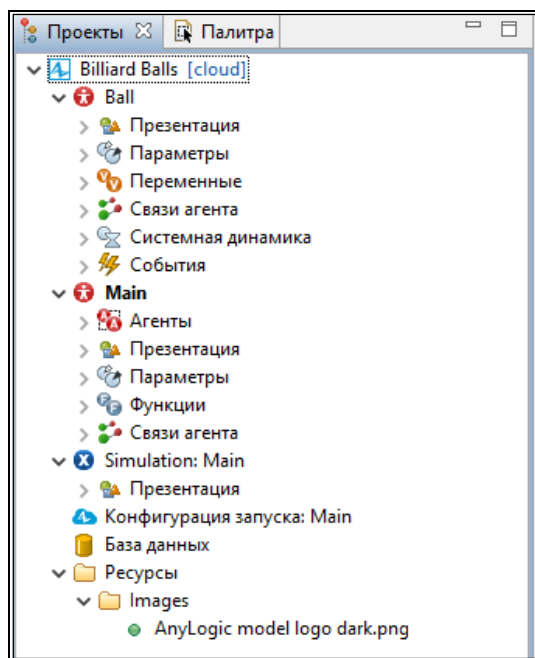

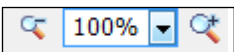


Рис. 4.58. Окно проекта

Наш проект «Billiard Balls» включает два класса активных объектов: класс Ball и класс Main. На дереве проекта (рис. 4.58) как составные элементы класса «Balls» показаны презентация, параметры, переменные, связи агента, системная динамика, события. Одна из ветвей в дереве проекта имеет название «Simulation» (Симуляция). В этой ветви происходит установка параметров для настройки симуляции эксперимента, таких как модельное время, параметры генерации случайных чисел, а также может быть введен фрагмент кода на Java для выполнения в процессе запуска модели.

Открыть окно проекта, если оно закрыто, можно с помощью щелчка мышью по кнопке «Проекты»  панели инструментов или командой «Вид | Проекты» из главного меню. Окно любого элемента дерева проекта (классов активных объектов, симуляции, ресурсов и т.п.) можно открыть двойным щелчком мышью по имени этого элемента в дереве проекта. Открытая нами среда создания проекта (рис. 4.57) содержит несколько окон. Структура активного объекта задается графически в специальном окне редактора – структурной диаграмме. В окне «Ball» представлен **стейтчарт** (или карта состояний), который определяет реакции активного объекта на внешние события – логику его дей-

ствий во времени. Стейтчарт является удобным расширением графического представления классического конечного автомата, он состоит из состояний и переходов между состояниями. В окне анимации можно построить анимированное поведение активного объекта.

Поле в любом окне можно двигать, нажав на нем правой кнопкой мыши. При этом также можно менять масштаб изображения в этом поле с помощью двух кнопок и окна масштаба , если окно активно.

Закройте и откройте окно проекта «balls», а также каждое из окон элементов, включенных в дерево проекта. Измените масштаб изображений в окнах.

При построении любой модели задается ее **структура** (т.е. ее **компоненты и связи этих компонентов**) и поведение отдельных компонентов. В AnyLogic активный объект имеет **структуру** и **поведение**. Структуру активного объекта составляют его **параметры, переменные, стейтчарты, а также экземпляры других активных объектов**, включенные как компоненты в данный активный объект (и, возможно, их связи).

На рис. 4.57 для нашего простого примера структурная диаграмма активного объекта – мяча – задается в окне с именем «Ball», где имеются **параметры, переменные, накопители, события** (рис. 4.59).

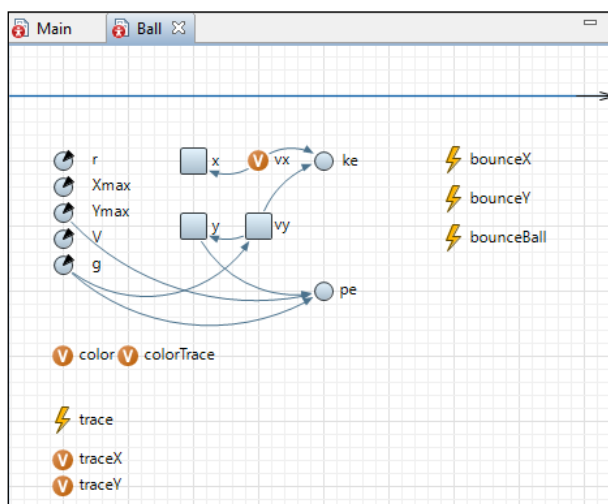
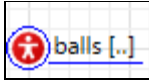


Рис. 4.59. Окно структуры активного объекта Ball

Связи между данными элементами показаны в виде стрелок.

Структура корневого активного объекта «Main» задана в окне с именем «Main». Именно корневой активный объект является верхним уровнем струк-

турной иерархии в нашем проекте. В модели (рис. 4.57) активный объект «**Main**» содержит агента с именем «**balls**»  – один экземпляр активного объекта Ball. В редакторе AnyLogic для каждого элемента модели существует свое **окно свойств**, в котором указываются свойства (параметры) этого элемента. При выделении какого-либо элемента в любом из окон редактора (в окне структуры, окне поведения, окне анимации или в окне проекта) справа появляется окно свойств именно этого выделенного элемента. Выделить элемент можно щелчком левой кнопки мыши на этом элементе. Окно «**Свойства**» (рис. 4.60) используется для просмотра и изменения свойств элементов. Открыть это окно, если оно закрыто, можно с помощью меню «**Вид | Свойства**».

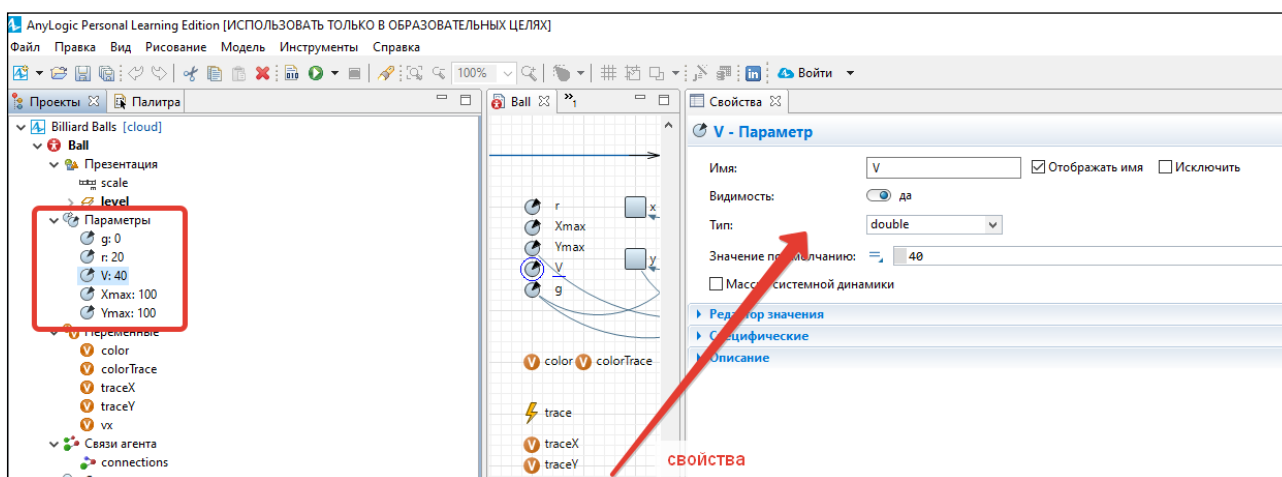


Рис. 4.60. Окно свойств класса «**Ball**»

Рассмотрим пример. При выделенном окне редактора структуры объекта «**Ball**» справа появится окно свойств, которое содержит три вкладки: «**Общие**», «**Редактор значения**», «**Специфические**» и «**Описание**». На вкладке «**Общие**» кроме имени объекта указаны его параметры. Смысл параметров очевиден: V – скорость, g – ускорение свободного падения, r – радиус мяча, X_{max} и Y_{max} – начальные значения координат, t – время.

Поведение мяча представлено в окне «**Ball**» (окно поведения активного объекта), содержащем простейший стейтчарт (карту состояний), который можно считать расширенным графом переходов **конечного автомата**.

Модель представляет собой несколько шаров, движущихся в пределах двумерной прямоугольной области. Шары упруго отскакивают друг от друга и от границ области. Изначально шары имеют случайные местоположения и скорости. Общая кинетическая и потенциальная энергии показаны на диаграмме. Вы можете добавить / удалить гравитационную силу, направленную вниз, и наблюдать за изменениями.

Шары реализуются как **агенты**, внутри которых находятся дифференциальные уравнения. Эта модель также демонстрирует захватывающие визуальные эффекты любой логики динамической графики.

Стейтчарт модели бильярдных шаров (рис. 4.61) состоит из трех **накопителей** x , y , V , которые определяют изменение положения шара при взаимодействии с границами или другими шарами. Накопитель является основным элементом системно-динамических диаграмм потоков и накопителей.

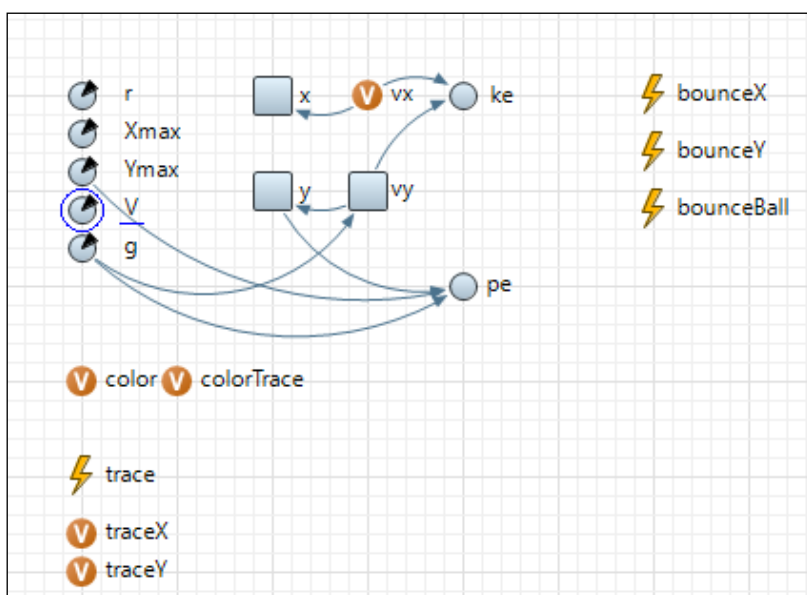


Рис. 4.61. Окно стейтчарта

В **системной динамике** накопители используются для представления таких объектов реального мира, в которых сосредотачиваются некоторые ресурсы. Это могут быть деньги, вещества, численности людей, какие-то материальные объекты и т.п. Накопители задают статическое состояние моделируемой системы. Их значения изменяются с течением времени согласно существую-

щим в системе потокам. Таким образом, потоки задают **динамику системы**. Входящие в накопитель потоки увеличивают значение накопителя, а исходящие из него потоки, соответственно, его уменьшают.

Для накопителей x , y , V_y используются дифференциальные уравнения:

$$\frac{dx}{dt} = V_x, \quad \frac{dy}{dt} = V_y, \quad \frac{dV_y}{dt} = g.$$

При установке правой части дифференциального уравнения на стейтчарте образуются связи накопителей с параметрами, переменными, динамическими переменными, и другими накопителями. Таким образом, накопители содержат изменяемую координату шара с течением времени.

Стейтчарт содержит также события (рис. 4.61). Событие выполняет какое-то действие в зависимости от условия. Событие **bounceX** (подпрыгивание по x координате) выполняется при условии

$$x \leq r \ \&\& \ v_x < 0 \ || \ x \geq X_{\max} - r \ \&\& \ v_x > 0$$

(координата x – центр шара по x). это выражение означает, что шар либо коснулся левой стенки площадки, либо правой площадки и пытается продолжить свой путь за пределы границ). В этом случае нужно реализовать событие отскока

$$v_x = -v_x;$$

`bounceX.restart();`

Меняем вектор движения на вектор с противоположным знаком и перезапускаем событие.

Аналогичным образом не даем мячу двигаться за пределы площадки по оси y . Событие **bounceY**, условие срабатывания

$$x \leq r \ \&\& \ v_x < 0 \ || \ x \geq X_{\max} - r \ \&\& \ v_x > 0,$$

действие

$$v_x = -v_x;$$

`bounceX.restart();`

Также на стейтчарте имеется событие **bounceBall** – отскок мяча. Оно зависит от функции **checkHit** (проверка, произошло ли касание мяча с другим мячом). Эту функцию можно найти в разделе «**Функции**» объекта **Main** (рис. 4.62).

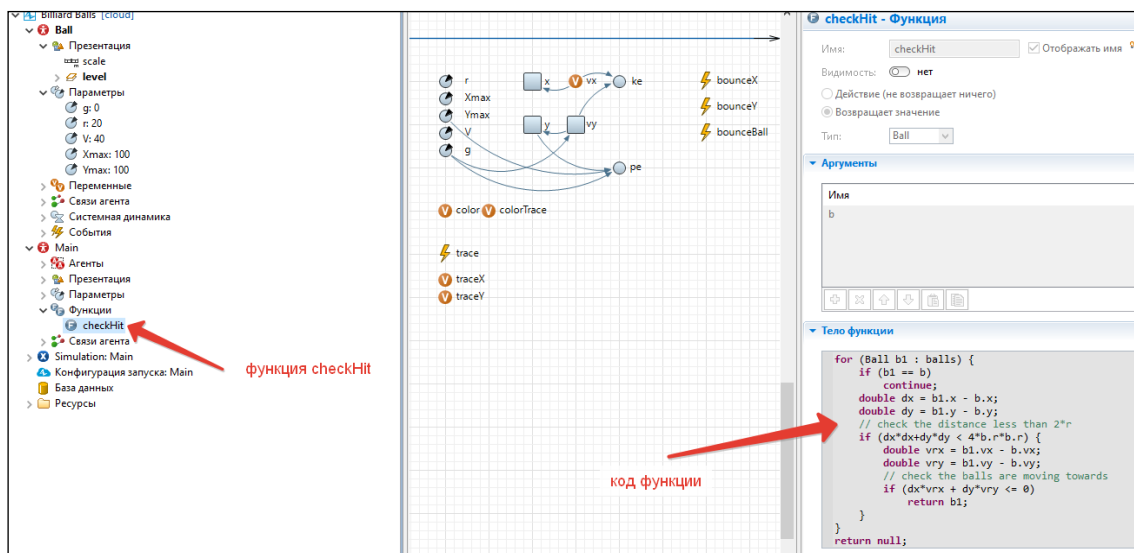


Рис. 4.62. Просмотр свойств функции **checkHit**

Таким образом, стейтchart в этой модели следит за событиями. При наступлении нужного события выполняется необходимое действие. И условие наступления события, и само действие, меняющее переменные модели, записываются здесь обычными алгебраическими соотношениями.

В настоящее время стейтchart широко используются как удобное средство визуального описания поведения сложных систем. В частности, стейтchart являются основным средством представления поведения в языке UML (Unified Modeling Language).

В окне редактора анимации активного объекта для модели строится двумерное анимационное представление, которое показывает, что происходит с моделью с течением времени. Именно здесь визуально представляется имитация поведения моделируемой системы. Для модели **Billiard Balls** в окне анимации построено изображение мяча, представленного закрашенным кругом. В другом окне можно построить трехмерное анимационное представление.

Анимация в AnyLogic создается в виде динамических графических объектов, которые дают возможность наглядно представить динамику изменения состояния моделируемой системы, т.е. поведение ее во времени. Основная идея здесь состоит в том, что параметры элементов анимационной картинки (для круга это его **координаты центра, радиус, цвет** и т.п.) связываются с переменными и параметрами модели. Изменение переменных модели во времени

заставляет изменяться во времени графический образ, что позволяет наглядно представить динамику моделируемой системы с помощью динамически меняющейся графики. На рис. 4.63 показано, что когда в нашей модели в окне анимации выделен красный круг, то в окне свойств представлены свойства этого круга. А именно координаты x и y центра круга анимационной картинке имеют статические значения (значения этих параметров в редакторе) 20 и 20, а площадка, по которой движутся шарики, связана с объектом «balls», параметры и стейтchart которого мы рассмотрели ранее. Таким образом, изменение данных переменных будет вызывать перемещение центра графического образа мяча на анимационной картинке при работе модели.

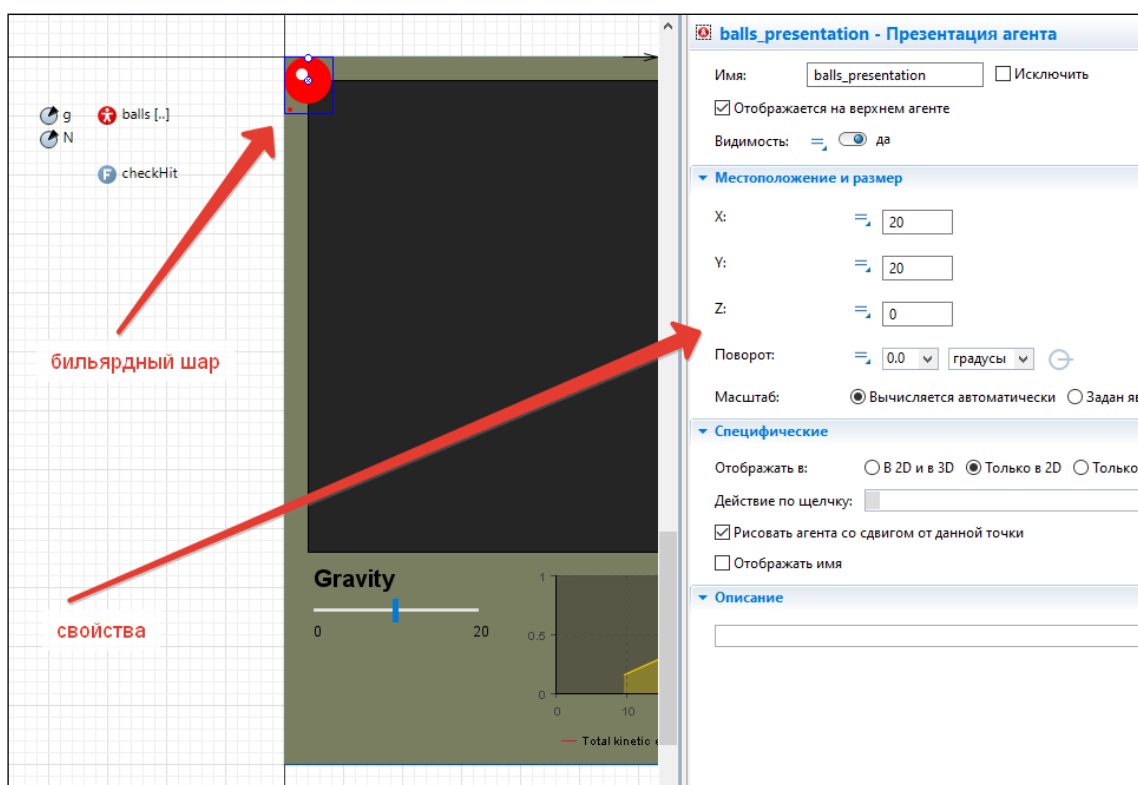



Рис. 4.63. Окно анимации и окно свойств

Щелкните мышью на нескольких элементах окон редактора (на переменных и на поле окна редактора структуры, на переходе и состоянии окна стейтчарта, на графических элементах окна редактора анимации, на объекте «simulation» дерева проекта и т.п.). Вы увидите, что для каждого элемента модели окно свойств имеет свою структуру и содержит специфическую информа-

цию и параметры, характеризующие именно данный элемент. Например, для параметра N (количество шаров) уже задано значение 10, которое вы можете сменить. Если выделено несколько элементов, окно свойств не показывает ничего. Модель «**Billiard Balls**» уже построена и готова к запуску.

В **режиме выполнения модели** пользователь имеет возможность выполнять различные эксперименты с моделью. Рассмотрим здесь простейшие средства управления экспериментом.

Для **запуска** модели щелкните кнопку  на панели инструментов. Этим действием запустится компилятор, который построит исполняемый код проекта на языке **Java**, оттранслирует его и затем запустит модель на исполнение. При этом откроется окно наблюдения (рис. 4.64).

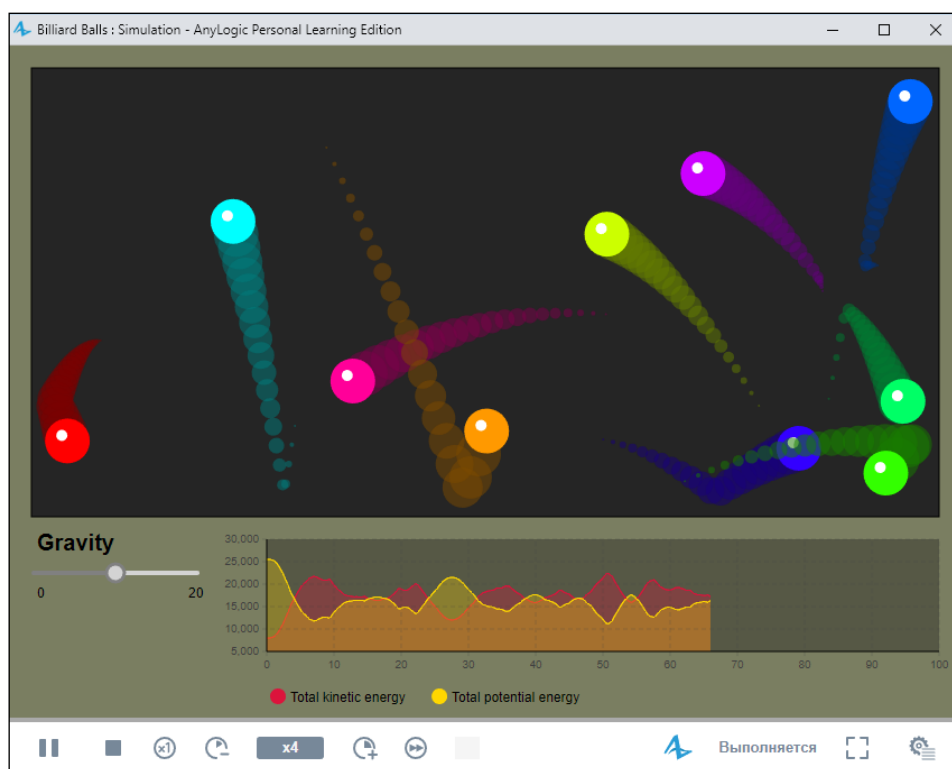



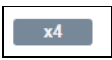


Рис. 4.64. Окно наблюдения проекта «Billiard Balls»

Снизу окна находятся кнопки управления запуском модели. Ее можно приостановить , запустить снова , остановить , ускорить в несколько раз .


В данном примере можно провести **такие эксперименты с моделью**. Для этого в окне анимации кроме движущегося изображения мяча можно видеть **слайдер** или **бегунок** – подвижные указатели для изменения параметров модели во время ее выполнения. **Бегунок** отвечает за силу тяжести (**Gravity**). Двигая слайдеры, можно менять в этой модели ускорение свободного падения g . Изменение параметра позволяет исследовать поведение модели в различных условиях. Это и есть компьютерный эксперимент. Изменение параметров и переменных возможно также и без слайдеров.

При запуске модели также подсчитывается общая кинетическая энергия (**Total kinetic energy**) и общая потенциальная энергия (**Total potential energy**). В экспериментах можно наблюдать, как меняется потенциальная и кинетическая энергия в зависимости от гравитации.

Проведите несколько экспериментов с моделью, изменяя параметр и переменные модели либо слайдерами, либо перед запуском.

В AnyLogic скорость выполнения модели может быть установлена максимальная (и модель будет выполняться в режиме виртуального времени с максимально возможной скоростью выполнения соответствующего программного кода процессором), а может соответствовать (при возможности) реальному физическому времени с некоторым коэффициентом. Единица модельного времени при этом равна одной секунде физического времени. Переключение между виртуальным и реальным временем исполнения модели осуществляется кнопкой



панели инструментов (при запущенной модели), а уменьшение масштаба времени (масштаба модельного времени относительно физического только в режиме реального времени) выполняется с помощью двух кнопок и расположенного между ними поля . Это поле указывает коэффициент ускорения модельного времени относительно физического (здесь $x4$ означает четырехкратный коэффициент ускорения).

Выполните несколько экспериментов с различными скоростями выполнения модели, используя кнопки останова, рестарта, запуска.

Автоматическая отрисовка изображения (**автоматическое обновление окна анимации**) производится каждый раз, как только произошло событие в

активном окне. При отсутствии этого режима отрисовка изображения анимации производится с некоторым шагом по времени. Чем чаще отрисовывается изображение, тем медленнее работает программа моделирования.

Запуск модели на выполнение производится в AnyLogic в соответствии с определенным набором значений параметров модели, а также с некоторыми дополнительными установками (например, точность, шаг численных методов и т.п.). Совокупность всех установок для проведения компьютерного эксперимента с моделью называется в AnyLogic **экспериментом**. Все эксперименты, возможные для выполнения в данном проекте, представлены как элементы группы (корня) с именем **«Эксперименты»** в окне проекта. Один такой эксперимент с названием **«Simulation»** уже построен при создании нового проекта с установками по умолчанию, он и выбран в качестве текущего (название текущего эксперимента показано жирным шрифтом).

Значения параметров, режим реального либо виртуального времени при выполнении эксперимента, условие прекращения выполнения эксперимента и многое другое, относящееся к проведению эксперимента, можно до запуска модели установить в окне свойств объекта **«Simulation»**, являющегося в данном проекте единственным элементом группы эксперименты в дереве классов модели. В окне **«Свойства»** объекта **«Simulation»** вы можете увидеть эти параметры и поменять установки, прежде чем запустить модель. Например, на вкладке **«Модельное время»** указано **«Реальное время со скоростью 4»**, т.е. модель запустится уже с четырехкратным ускорением. Начальное значение параметра *gravity* равно 10, что примерно соответствует ускорению свободного падения на Земле, ближе к 20 – это условия как на Юпитере, на Луне – около 1,6. Параметр **«Number of balls»** отвечает за количество бильярдных шаров на поле. Дополнительно можно определить условие остановки в заданное время или заданную дату и запустить модель. Полный набор свойств экспериментов описан в **руководстве пользователя AnyLogic (раздел пакета AnyLogic)**, а для данной модели показан на рис. 4.65.

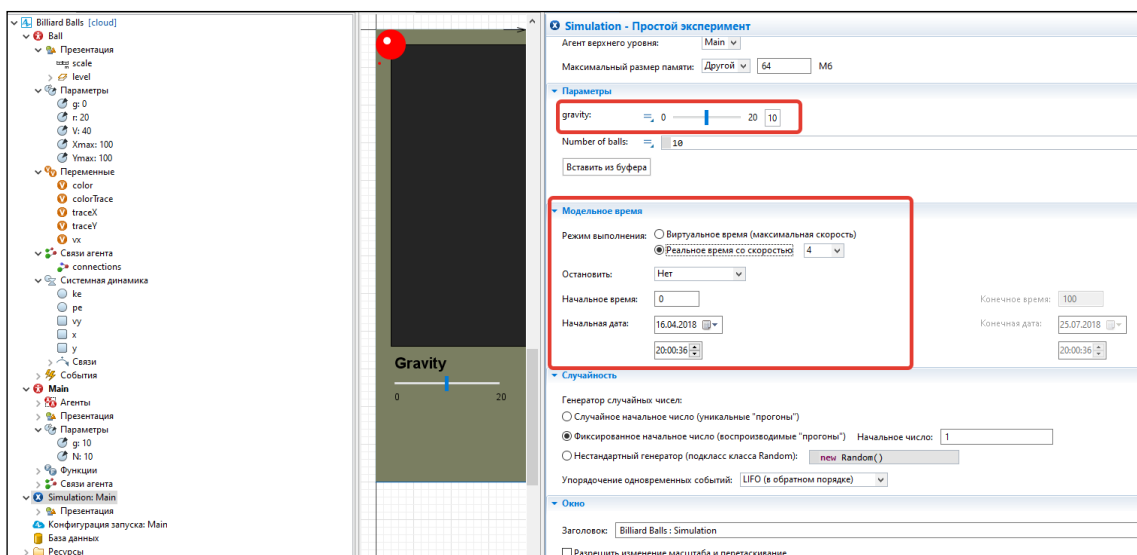



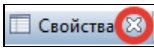

Рис. 4.65. Работа со свойствами эксперимента «Simulation»

Для одной и той же модели в AnyLogic можно определить несколько различных экспериментов на этапе построения модели.


Работать с окнами при разработке и исследовании модели требуется постоянно. Поэтому необходимо уметь открывать нужные окна, изменять их размеры, закрывать, сворачивать и разворачивать.

Вновь откройте редактор проекта «Balls». В редакторе и при работе модели открытие и закрытие окна проекта и окна свойств выполняется соответ-

ственно кнопками  панели инструментов. Кроме того, открыть эти окна можно, выбрав в главном меню команды «Вид | Проекты, Вид | Свойства, Вид | Палитра» соответственно. Откройте и закройте окна несколько раз в редакторе и в окне наблюдения при выполнении модели.

Окна структуры, поведения или анимации можно открыть двойным щелчком мыши на именах соответствующих объектов дерева проекта, если они закрыты. Попробуйте закрыть () и открыть несколько раз каждое из окон редактора. Для удобства редактирования любое окно может быть максимально увеличено () так, чтобы оно заняло все поле редактора. В этом случае кнопки управления таким окном размещаются в правом верхнем углу.

Каждое окно редактора сделайте активным, щелкнув на нем мышью, измените его размеры, измените масштаб изображения в ту и другую сторону.

Запустите модель на выполнение (). В появившемся окне наблюдения можно открыть «Панель разработчика» (рис. 4.66).

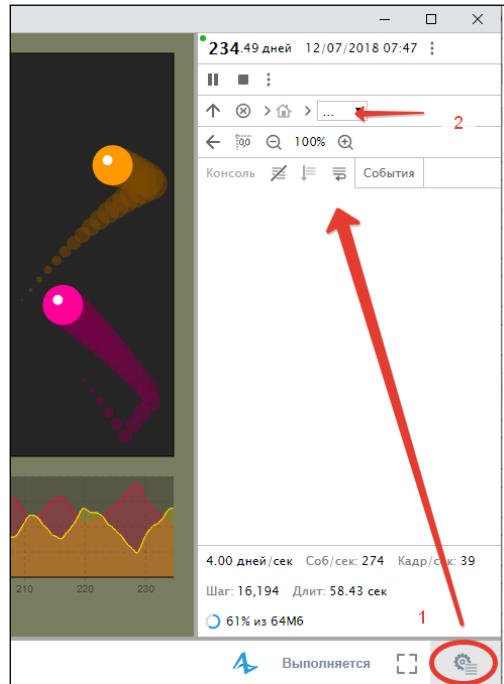


Рис. 4.66. Открытие панели разработчика при запуске модели

Можно отобразить окно стейтчарта «balls» (2), и мы увидим логику и анимацию движения одного шара, при этом можно наблюдать в динамике изменение параметров и накопителей.

Состояние окна стейтчарта «balls» показано на рис. 4.67.

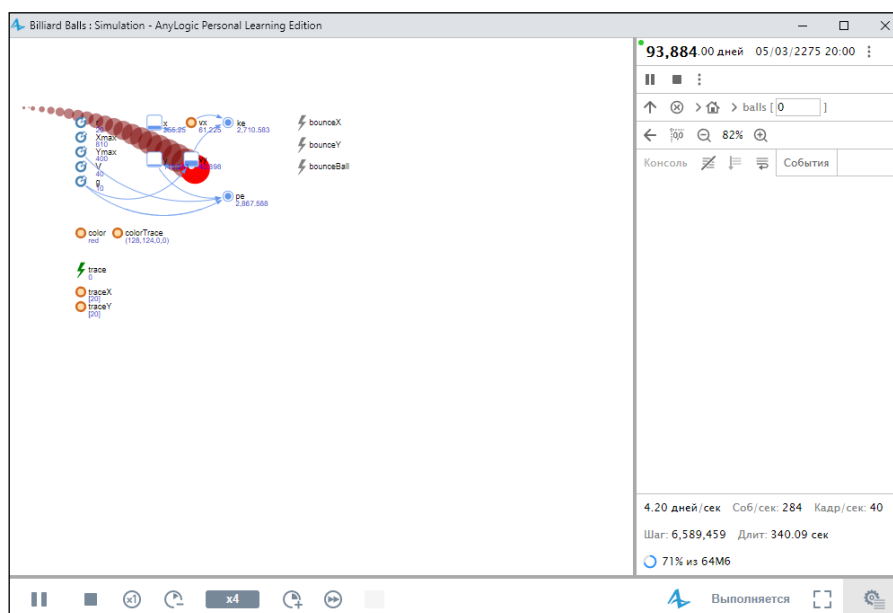


Рис. 4.67. Окно стейтчарта Balls в момент имитации модели

Выполним некоторые упражнения с моделью **Balls**, которые дадут некоторое представление о средствах разработки моделей в AnyLogic.

Для наглядности дополним анимационное представление поведения мяча так, чтобы при отскоке мяча от левой и правой стенок менялся на синий цвет, а при отскоке от верхней и нижней стенок менялся на белый. Для этого есть события **bounceX** и **bounceY**, которые меняют направление движения. Событие **bounceX** уже имеет действие. Дополним его строкой `color = Color.white`, которая изменит цвет шарика.

Получится код:

```
vx = -vx;
```

```
color = Color.white;
```

```
bounceX.restart().
```

Аналогичным образом дополним код для **bounceY**:

```
vy = -vy;
```

```
color = Color.blue;
```

```
bounceY.restart().
```

Запустите модель и посмотрите, что получилось.

Самостоятельно добавьте изменение цвета шарика при столкновении с другим шариком на изначальный цвет.

Рассмотрим еще одну модель системной динамики – **«Модель сердечных сокращений»**. Как уже говорилось, имитационное моделирование зародилось и стало актуальным вследствие необходимости исследования сложных систем, имеющих важное практическое значение, которые проанализировать обычными способами в заданные сроки, кроме компьютерного эксперимента, в принципе, не представляется возможным.


В отличие от других видов и способов моделирования с применением ЭВМ имитационное моделирование имеет свою специфику: запуск в компьютере взаимодействующих вычислительных процессов, которые являются по своим временным параметрам – с точностью до масштабов времени и пространства – аналогами исследуемых процессов.

AnyLogic – система имитационного моделирования нового поколения, основанная на результатах, полученных в теории моделирования и в информационных технологиях за последнее десятилетие.

Язык моделирования AnyLogic доказал свою эффективность в моделировании больших систем повышенного уровня сложности. Основными строительными блоками модели AnyLogic являются активные объекты, которые позволяют моделировать любые объекты реального мира.

Активный объект является экземпляром класса активного объекта. Чтобы получить модель AnyLogic, необходимо создать классы активных объектов (или использовать объекты библиотек AnyLogic) и задать их взаимосвязи. AnyLogic интерпретирует создаваемые вами графически классы активных объектов в классы Java. Поэтому вы можете пользоваться всеми преимуществами объектно-ориентированного моделирования: **наследованием, полиморфизмом** и т.д.

В данных лабораторных работах на примерах разработки простых моделей динамических систем описываются основные средства создания моделей в AnyLogic и представление результатов моделирования. Модели эти интересны не сами по себе. Простые, понятные всем явления здесь выбраны для моделирования специально, чтобы трудности понимания постановки проблемы не затмевали вопросов использования технических приемов при построении модели.

Замечание. При разработке проектов не забывайте периодически сохранять с помощью кнопки  «Панели инструментов» вносимые изменения. Кроме того, набрав «Инструменты | Настройки», можно открыть вкладку «Основные» и установить флажок «Включить автосохранение». Также можно установить интервал автосохранения, например, в 1 минуту.


Математики давно задумывались над тем, чтобы построить математическую модель сердца для описания режимов его работы. Это нужно, в частности, для того, чтобы найти значения факторов, при которых наступает сбой сердечного ритма. Волновой процесс, протекающий в сердце, вполне возможно описать с помощью формул, такие уравнения известны. Вопрос в другом. Что это за волны, насколько адекватно уравнения отражают происходящие в сердечной мышце процессы? Это требует длительной серьезной работы математиков и физиологов.

Здесь мы не будем заниматься данным вопросом. Мы рассмотрим простейшую математическую модель, описывающую процессы, которые похожи на биение сердца. Модель эта задается парой дифференциальных уравнений первого порядка:

$$\frac{dx}{dt} = \frac{(x - x^3 - b)}{eps}, \quad \frac{db}{dt} = x - x_0,$$

где x – радиус сердца, x_0 – его значение в начальный момент, b – переменная, а eps – параметр. Эта модель – одна из простейших, описывающих динамику работы сердца.

Целью построения данной модели является исследование характера зависимостей переменных состояния x и b от времени при разных значениях параметра eps , а также анализ фазовой диаграммы изменения радиуса x от значения b . Очевидно, что все эти величины вещественные. Для построения модели необходимо задать начальные значения переменных x и b , а также значение параметра eps .

Для хранения проектов необходимо создать новую папку. Создайте, например, на «Рабочем столе» папку с именем «My Models». Запустите AnyLogic (поиск Windows 10 | AnyLogic) или щелкните по файлу проекта AnyLogic (расширение alp, значок ). Новый проект можно создать и сочетание клавиш **Ctrl + N** (рис. 4.68).

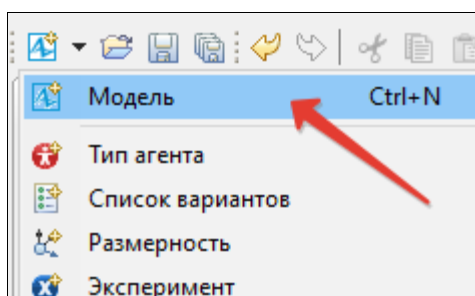


Рис. 4.68. Создание новой модели

В появившемся диалоговом окне установите нужную рабочую папку (вы создавали папку на «Рабочем столе»), в ней наберите «Heart» как имя проекта (это имя нового файла, в котором будет храниться ваш новый проект) и щелкните «ОК». Новый проект под названием «Heart» будет создан, и на экране вы увидите изображение, представленное на рис. 4.69.

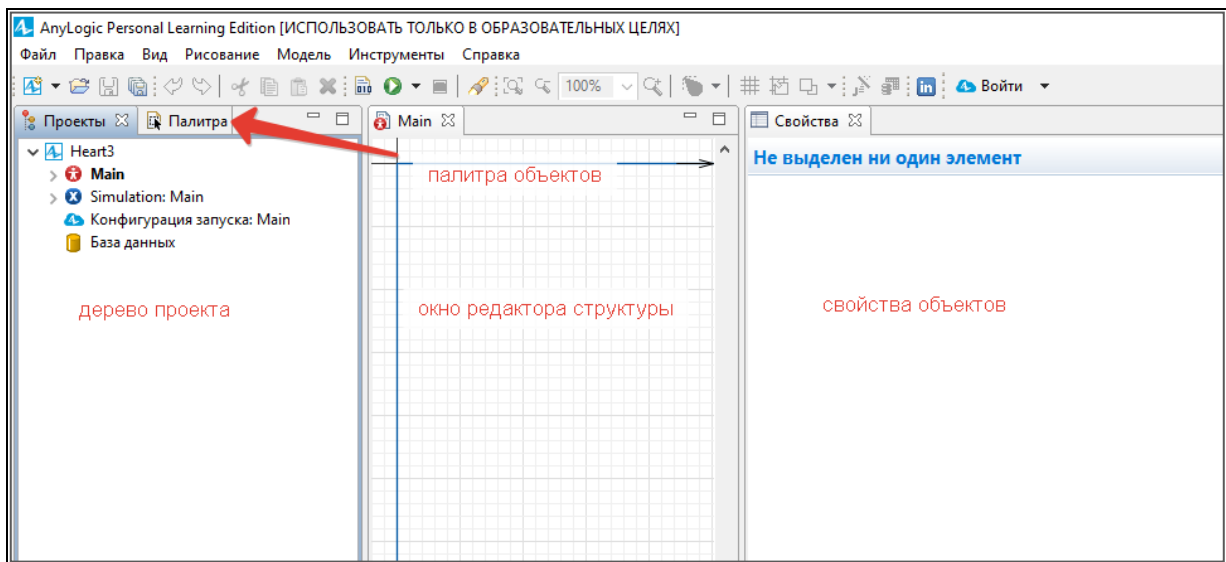


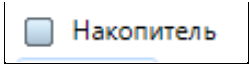
Рис. 4.69. Наименования окон AnyLogic


Открытое окно редактора нового проекта содержит три части. Слева в окне классов автоматически будет строиться дерево проекта. Оно обеспечивает удобную навигацию по элементам проекта, которые будут создаваться при построении модели. Для нового проекта в нем уже создан корневой класс активного объекта с именем «**Main**», а для проведения экспериментов с будущей моделью уже создан один эксперимент с именем «**Simulation**» у узла дерева «**Эксперименты**».


Центральное окно – окно графического редактора структуры для создания структуры активного объекта, представляющего модель. С помощью кнопок управления этим окном, которые находятся вверху справа, можно уменьшить, свернуть или закрыть это окно. Уменьшите размер окна редактора структуры объекта «**Main**». Двойным щелчком мыши на имени класса активного объекта в окне классов (в данном случае «**Main**») можно открыть окно редактора структуры этого объекта, если оно закрыто. Окно справа – это окно свойств выделенного элемента модели. В данном случае это окно показывает свойства класса корневого объекта с именем «**Main**», если окно его структуры активно. В окне свойств можно редактировать (изменять, устанавливать) свойства любых создаваемых объектов модели. Измените имя корневого объекта нашей модели, назвав его «**Heart**» (вместо установленного по умолчанию имени «**Main**»).

Для этого в поле «Имя класса» вкладки «Общие» окна «Свойства» корневого объекта введите «Heart» вместо «Main». В окне классов имя корневого объекта сразу изменится.

Наша задача – построение модели, в которой присутствуют две переменные состояния, x и b , и два параметра – x_0 и eps , где x_0 – начальное значение x . Начальное значение переменной b зададим константой.

Перейдите на вкладку «Палитра», откройте раздел «Системная динамика» вкладки «Палитра». Для введения первой переменной x щелкните мышью на кнопке накопитель , после чего щелкните мышью в каком-либо месте поля окна редактора структуры объекта «Heart».

Пиктограмма  появится в поле редактора с именем «Stock». Одновременно справа вместо окна свойств объекта «Heart» появится окно свойств переменной (именно эта переменная сейчас выделена). В это окно в поле имени (Имя) вместо предопределенного имени *stock* введите x и нажмите клавишу <Enter>.

При выделенной пиктограмме переменной () ее имя можно перемещать по полю окна структуры и изменять. Саму пиктограмму переменной также можно перемещать по полю (при нажатой на ней левой кнопке мыши).

В модели переменная x определяется дифференциальным уравнением

$$\frac{dx}{dt} = \frac{(x - x^3 - b)}{eps}$$

с начальным значением x (равным x_0) и с параметром eps . В AnyLogic можно подобные зависимости задавать именно в таком аналитическом виде. Для того чтобы определить переменную x , в окне свойств выберите режим задания уравнения «Произвольный», у вас станет активным поле дифференциального уравнения, в которое и нужно ввести наше уравнение

$d(x)/dt =$

 $(x - x * x * x - b) / eps$, которое представлено на рис. 4.70.

В поле начального значения запишите x_0 . Вторая переменная – накопитель b определяется дифференциальным уравнением

$$\frac{db}{dt} = x - x_0$$

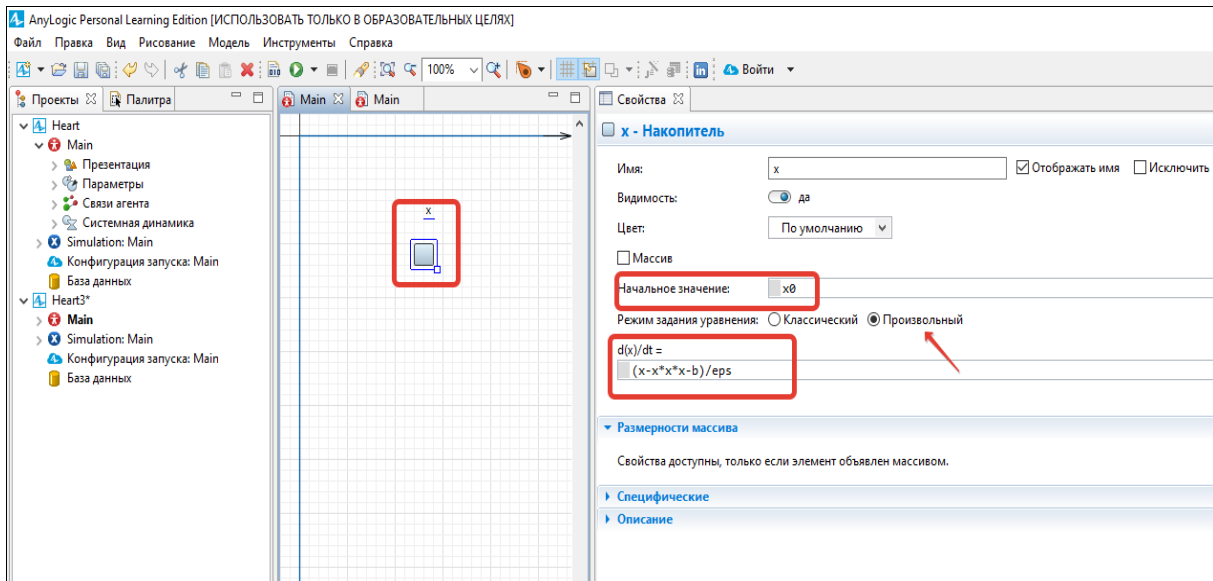
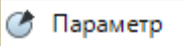



Рис. 4.70. Установка свойств накопителя x

Действия по ее введению в модель аналогичны тем, которые были сделаны ранее для переменной x . Пусть начальное значение b равно 0 . В поле «Начальное значение» окна свойств переменной b величину 0 можно не записывать: если это поле пусто, по умолчанию значение переменной считается нулевым.

В поле окон редактора можно помещать текстовые комментарии. Для того чтобы поместить такой комментарий на вкладке «Палитра» перейдите в раздел «Презентация», щелкните на кнопке «Текст» Аа Текст панели инструментов и затем, щелкнув в какое-нибудь место окна редактора, нарисуйте прямоугольник, в котором можно записать произвольный текст-комментарий, никак не влияющий на работу модели. Размеры и место расположения этого комментария можно менять с помощью мыши. Введите, например в поле редактора комментарий «Структура активного объекта». Его можно редактировать в поле «Текст» окна свойств этого комментария.

Пусть $x_0=0,5$, $eps=0,01$. Это параметры, которые, в отличие от накопителей, не будут меняться в ходе выполнения модели. Перейдите на вкладку «Па-

литра», раздел «Системная динамика», перенесите в окно структуры «Параметр» . В свойствах этого объекта задайте имя x_0 , тип **double**, Значение по умолчанию 0,5. Остальные поля также оставьте без изменения. Переменная eps со значением 0,01 задается так же.

Вы можете заметить, что появились сигналы, информирующие о возможных ошибках построения модели . Это происходит потому, что мы еще не построили связи между накопителями и параметрами. Чтобы сделать это, достаточно выбрать любой накопитель (например, x), и, щелкнув по нему правой кнопкой мыши, выбрать пункт меню «Исправить ошибку в связях | Создать недостающие связи». То же самое проделать и с накопителем b .

В результате на экране вы получите изображение, представленное на рис. 4.71.

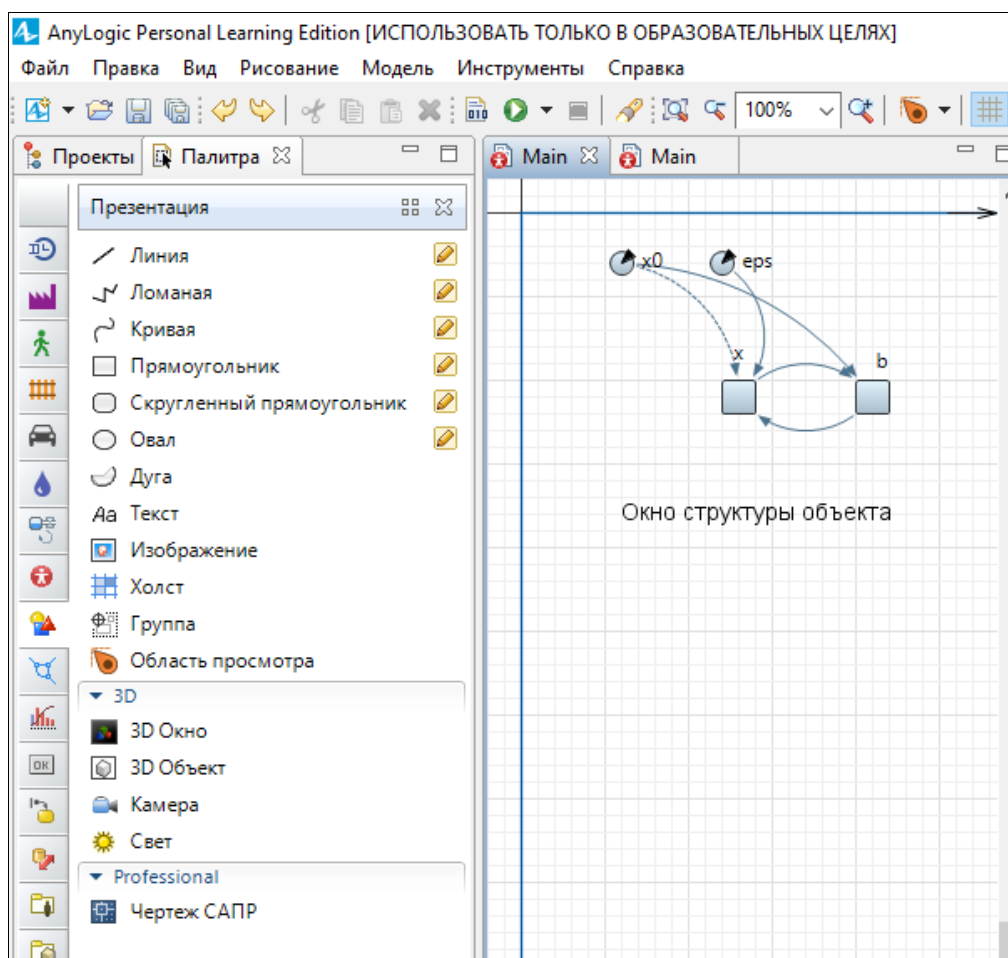


Рис. 4.71. Стейтchart для модели



Щелкнув на кнопке запуска выполнения модели , после компиляции в окне наблюдения увидим открытым только окно «**Root**» переменных и параметров с их начальными значениями (рис. 4.72).




Рис. 4.72. Запуск модели

Запуск модели () приведет к тому, что переменные b и x в этом окне начнут изменяться в соответствии с определенными для них уравнениями.

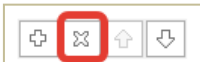
По умолчанию выполнение модели будет очень долго продолжаться, в окне свойств объекта «**Simulations**» установлено «**Нет**» в поле «**Остановить**». Но это можно и изменить.

Одним из преимуществ AnyLogic является возможность наглядного представления поведения модели, в частности, представления изменения во времени всех ее переменных. Введем графики изменения переменных x и b .

1. Запустите модель на выполнение 

2. Выберите на вкладке «**Палитра**» раздел «**Статистика**» и поместите в окно структуры проекта «**Временной график**». Так он называется потому, что горизонтальная ось (абсцисса) отражает временные отрезки.

3. В окне свойств временного графика в разделе «**Данные**» в поле «**Заголовков**» введите x , в поле «**Значение**» тоже введите x . Второе значение для

графика удалите с помощью кнопки 

4. Для отображения значения накопителя b также постройте временной график.

5. На «Палитре» в разделе «Статистика» найдите элемент «График» и перенесите его в окно структуры проекта. В поле «Заголовок» введите «Фазовая диаграмма», в поле «Значение по оси x » введите x , «Значение по оси y » введите b .

6. Запустите модель на выполнение. В окнах будут рисоваться графики значений от времени соответствующих переменных.

Изменять параметр, как уже говорилось, можно в окне этого параметра, появляющемся в результате двойного клика на нем в окне «Root» или при выборе команды «Изменить» контекстного меню параметра (рис. 4.73).

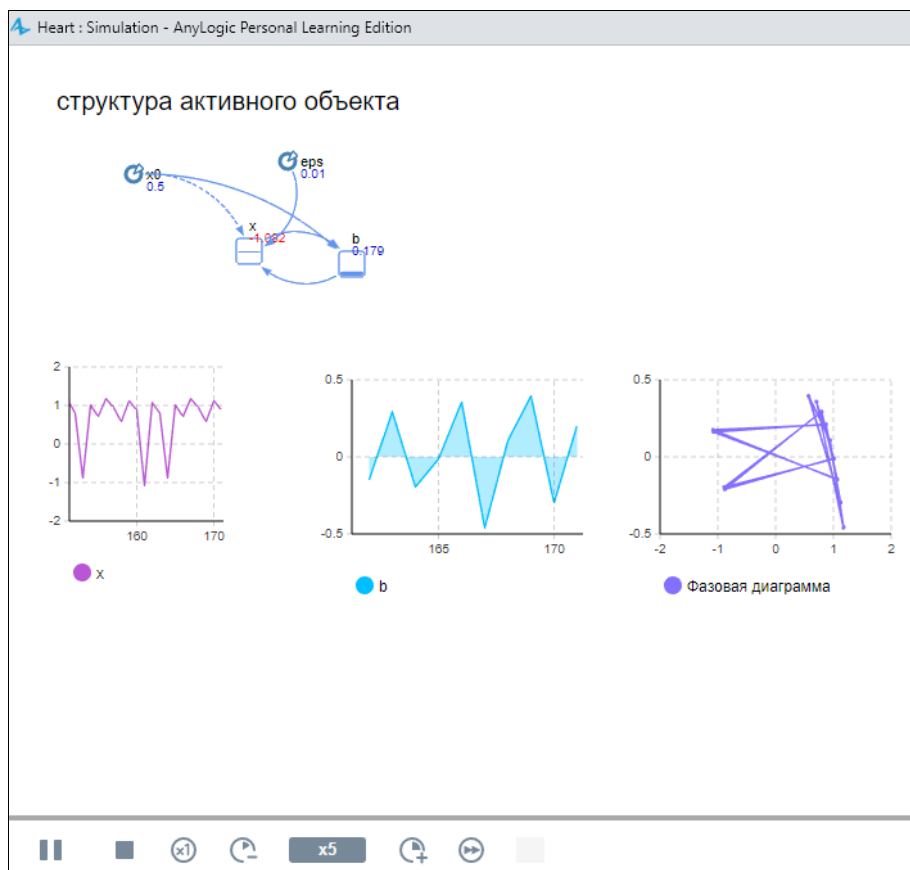
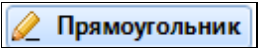


Рис. 4.73. Демонстрация модели с графиками

Для лучшего понимания динамики модели и развивающихся во времени процессов в AnyLogic предусмотрена возможность построения анимированного изображения, состоящего из динамических графических элементов. Графиче-

ские элементы здесь называются динамическими, поскольку все их параметры: координаты, размер, цвет и даже их видимость – в процессе выполнения модели можно сделать зависимыми от переменных и параметров, которые меняются со временем при выполнении модели.

Закройте скомпилированную модель и вернитесь в редактор. Перейдите на вкладку «Палитра» и найдите раздел «Презентация».

Для анимации мы будем использовать специальный прямоугольник, в который поместим все элементы. В разделе «Презентация» найдите элемент «Прямоугольник» и, нажав на него дважды мышью, переведите его в состояние редактирования . В окне редактора структуры проекта нарисуйте прямоугольник и в свойствах задайте какой-либо нейтральный цвет этого прямоугольника, например, светло-желтый (рис. 4.74). В дальнейшем будем называть этот прямоугольник «Поле анимации».

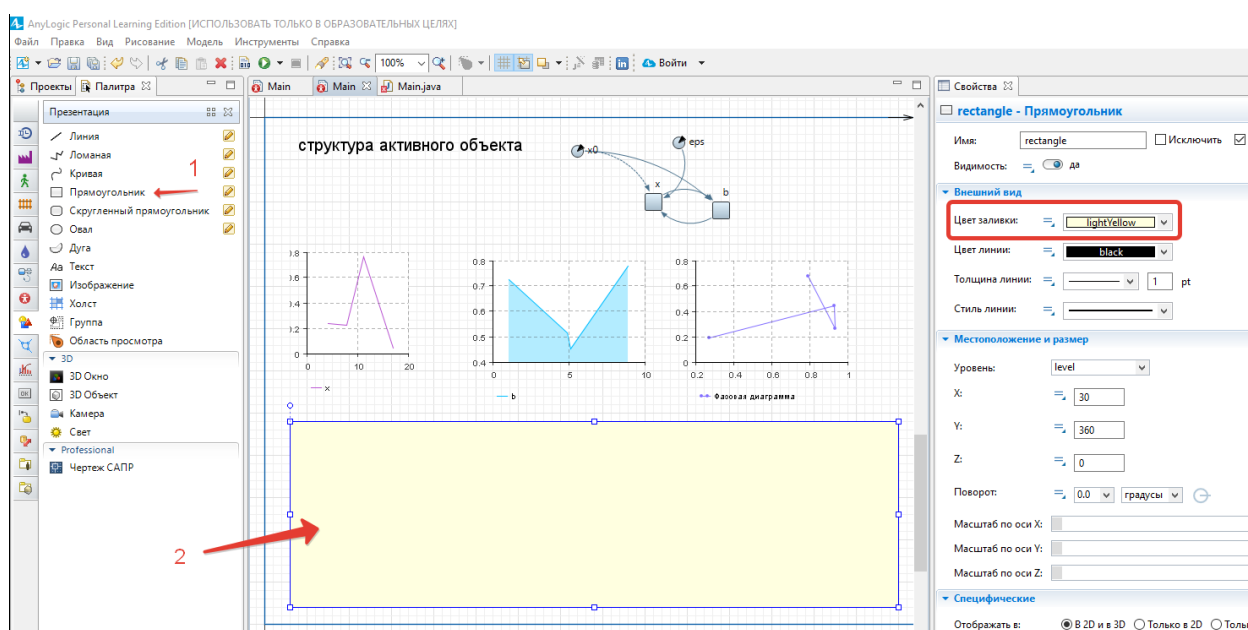
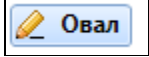



Рис. 4.74. Создание поля анимации

Будем строить анимацию динамики работы сердечной мышцы как изображения круга (овала), радиус которого будет меняться. Этот радиус является функцией от значения переменной x модели. Во-первых, построим изображение овала. Для этого на вкладке «Палитра» в разделе «Презентация» найдите объект «Овал» и двойным щелчком мыши переведите его в состояние редак-

тирования . Нарисуйте в любом месте поля анимации какой-нибудь овал. Справа появится окно «Свойства» этого овала. Свойства овала уже установлены по умолчанию, и мы можем их редактировать. По умолчанию имя этого объекта будет «Oval», координаты x и y соответствуют месту, куда мы поместили овал в поле анимации, а ширина и высота соответствуют тому, что мы нарисовали.

Если мы запустим модель на выполнение, мы увидим, что в окне анимации этот овал находится неподвижно именно в том месте, которое определено его заданными координатами, и с установленными нами размерами. Как уже говорилось, в AnyLogic принята следующая концепция: каждая характеристика графического элемента, помещенного в окно анимации, имеет два значения – **статическое и динамическое**.

Статическое значение определяет характеристику (координату, угол поворота, цвет и т.п.) объекта в окне анимации в статике как константу в процессе редактирования. Динамическое определяет значение этой характеристики в процессе выполнения модели и может быть определено как значение любой переменной модели и даже как любая функция от переменных модели. Поэтому обычно характеристика графического объекта имеет два поля, которые можно менять нажатием на пиктограмму . Радиус X:

Если динамическое значение не определено, в динамике характеристика этого графического объекта сохраняет свое статическое значение.

Выделите овал, который будет представлять динамику бьющегося сердца. Статические значения его характеристик задайте так: в окне свойств овала в левых полях установите координаты центра x и y равными 150 и 450 соответственно, цвет заливки определите бордовым, цвет линии границы – красным; ширину линии границы установите равным 2.

Динамические значения характеристик овала должны отражать «жизнь» этого объекта в процессе выполнения модели. Изменение объема сердца будем представлять меняющимся радиусом овала. Радиус овала задается в окне

его свойств на вкладке «Овал». Определите радиусы овала как меняющиеся в зависимости от переменной x так, как показано на рис. 4.75.

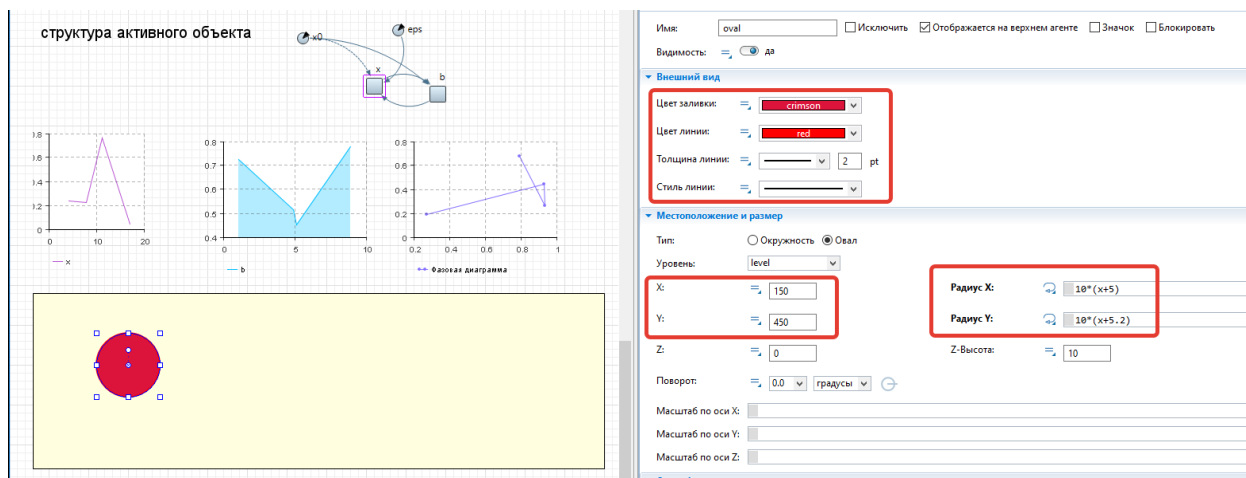



Рис. 4.75. Добавление к анимации объекта «Сердце»

Обратите внимание, что радиус x и радиус y – это динамические характеристики, и их нужно задавать в поле  «Динамическое значение».

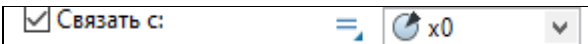
Радиус x меняется по уравнению $10*(x+5)$, радиус y меняется по уравнению $10*(x+5.2)$.

Запустите модель. Проведите эксперименты с установкой различных коэффициентов между реальным и модельным временем. Кроме того, можно изменять параметры модели ($x0$ и eps) и наблюдать, как изменяется характер сердцебиения.

Целью разработки этой модели является исследование влияния ее параметров на поведение системы. Введем график непосредственно в поле анимации. Выделите накопитель x и щелкните по нему правой кнопкой мыши. Появится контекстное меню, в котором выберите «Создать диаграмму» | «Создать временной график». Появившийся временной график перенесите в поле анимации. Потянув за «узелки» временного графика установите его в правой части поля анимации.

В окне свойств временного графика в разделе «Данные» проверьте (или установите сами, если не установлено) «Заголовок x », «Значение y ». В разделе «Обновление данных» в поле «Отображать до 100» последних значений. В

разделе «Масштаб» в поле «Временной диапазон» поставьте значение 20; установите переключатель «Вертикальная шкала» в состояние «Фиксированный», от -1,5 до 1,5.

Для более гибкого управления параметрами в процессе запуска модели удобно использовать специальные элементы управления – бегунки. Установим их для параметров x_0 и eps . Щелкните правой кнопкой мыши по параметру x_0 в стейтчарте. В появившемся контекстном меню выберите «Создать элемент управления» | «Создать бегунок». Для него в окне свойств установите минимальное значение 0, максимальное 1 и проверьте, что в поле «Связать с» установлен параметр x_0 .

Щелкните мышью на кнопке слайдера (бегунка) и поместите его в поле анимации под овалом. С помощью объекта «Текст» (вкладка «Палитра», раздел «Презентация») подпишите этот бегунок как $x_0=$. Если при перемещении по экрану бегунок оказался под полем анимации, то можно его переместить на передний план с помощью кнопок верхнего меню (рис. 4.76).

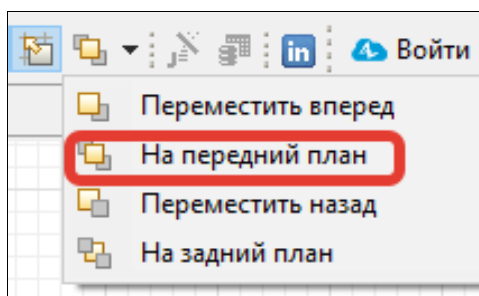


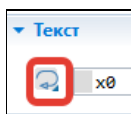
Рис. 4.76. Вид панели перемещения объекта

Аналогичным образом создайте бегунок для второго параметра eps , для бегунка возьмите минимальное значение 0,01, максимальное 0,5. Запустите модель.

Двигая слайдер, можно наблюдать изменение характера пульса на графике. В окне запуска модели, содержащем переменные и параметры модели, можно видеть, как значение параметра eps изменяется при движении слайдера. Конечно, имя параметра и его текущее значение удобно отобразить рядом со слайдером. У нас уже имеется подпись параметров рядом со слайдерами. Для параметра x_0 подпись выглядит, как « $x_0=$ », а для параметра eps подпись

«eps=». Поставьте после знака «=» для параметра x_0 еще один текстовый объект (вкладка «Палитра», раздел «Презентация»). В окне свойств задайте цвет этого нового текста синим. В окне свойств в разделе «Текст» выберите «Дина-

мическое значение x_0 »



Аналогичным образом поместите новый объект «Текст» для отображения значения параметра eps (рис. 4.77).

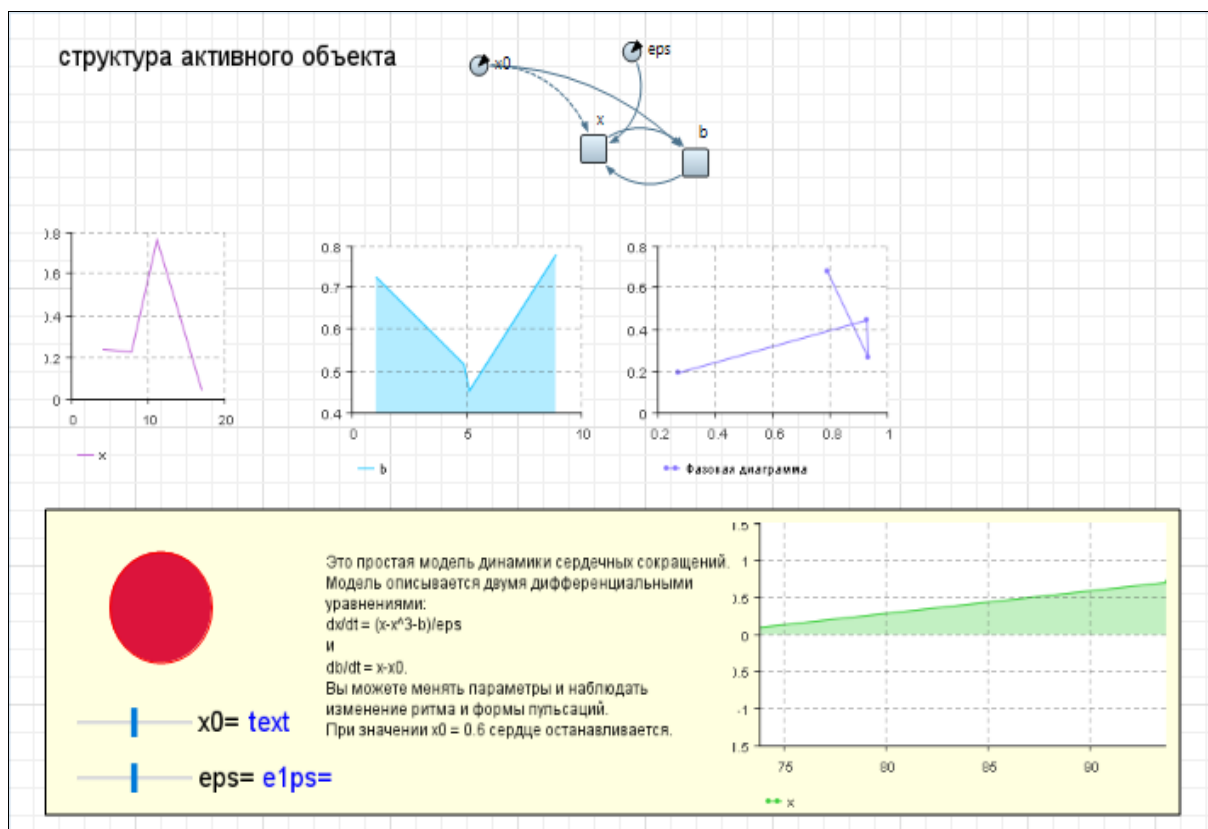


Рис. 4.77. Результат построения модели в редакторе

Запустите модель и убедитесь, что значение параметров x_0 и eps при изменении его слайдера можно видеть непосредственно у слайдера.

В поле анимации введите поясняющий текст. На вкладке «Презентация» в разделе «Презентация» снова выберите объект «Текст». Двойным кликом мыши переведите его в состояние редактирования и перенесите объект в поле анимации на свободное место. В поле «Текст» данного окна зададим статический текст модель сердца, который будет отображаться в этом месте поля ани-

мации. Нажав кнопку «**Выбрать**» окна свойств этого объекта, можно выбрать параметры (стиль, размер) текста.

Выберите шрифт **Century Gothic**, стиль **Bold**, размер 22. Следующий текст, содержащий пояснение к модели, введите в поле статического текста:

Это простая модель динамики сердечных сокращений. Модель описывается двумя дифференциальными уравнениями: $dx/dt = (x-x^3-b)/\epsilon$ и $db/dt = x-x_0$. Вы можете менять параметры и наблюдать изменение ритма и формы пульсаций. При значении $x_0 = 0,6$ сердце останавливается.

Все другие параметры текста оставьте установленными по умолчанию. Для того чтобы текст уместился в окне анимации по ширине, в нужных местах следует поставить перевод строки.

Результат создания модели в редакторе показан на рис. 4.77, а в режиме запуска модели показан на рис. 4.78.

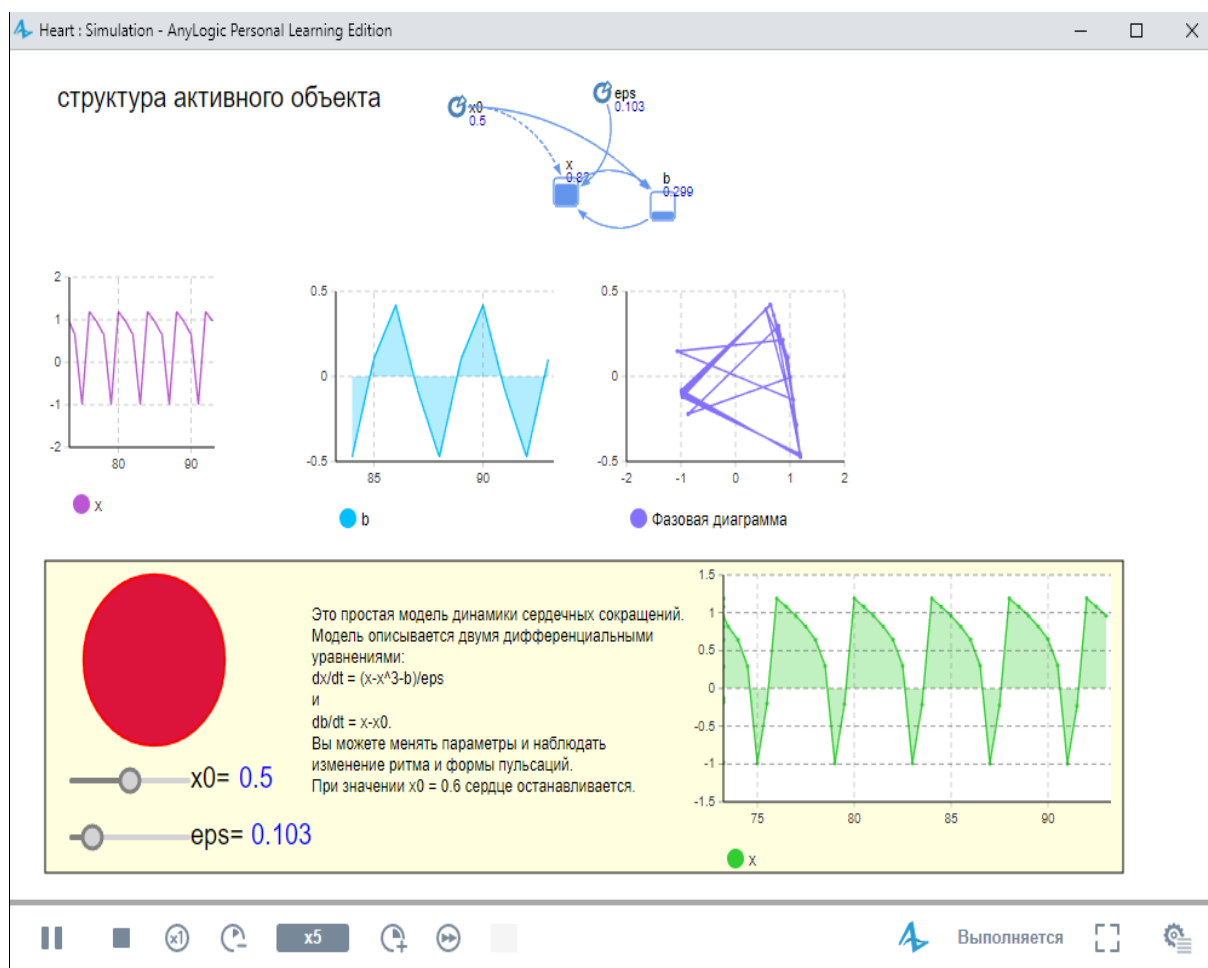



Рис. 4.78. Результат моделирования

4.6. ANYLOGIC ДИСКРЕТНО-СОБЫТИЙНАЯ МОДЕЛЬ

С помощью инструментов AnyLogic можно быстро создавать сложные дискретно-событийные модели:

- модели производственных процессов;
- модели систем обслуживания (банки, аэропорты и т.д.);
- модели бизнес-процессов с оценкой затрат операций;
- модели логистики и цепочек доставки.

Библиотека моделирования процессов позволяет создавать гибкие модели с наглядной визуализацией моделируемого процесса и возможностью сбора необходимой статистики.

Итак, создаем новую модель. Щелкните по кнопке «Создать» , при этом появится диалоговое окно, в котором вы должны будете дать имя файлу вашей модели и выбрать директорию, где он будет храниться (рис. 4.79).

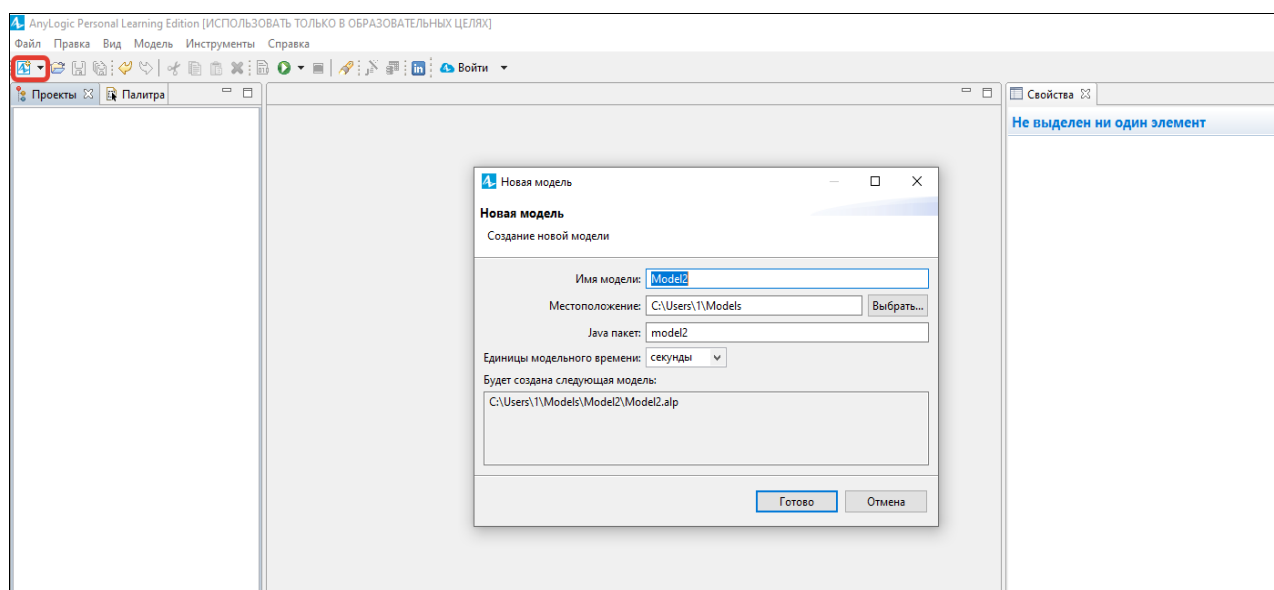


Рис. 4.79. Создание нового проекта

Вы можете создать различные наборы конфигурационных настроек модели, называемые экспериментами. В дереве модели эксперименты отображаются под элементом «Эксперименты».

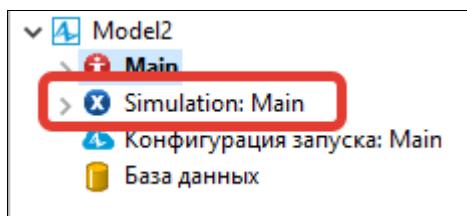


Рис. 4.80. Настройки симуляции

Один эксперимент, названный «**Simulation**», создается автоматически. Выберите его щелчком мыши по элементу дерева и посмотрите настройки модели в окне «**Свойства**» (рис. 4.81).

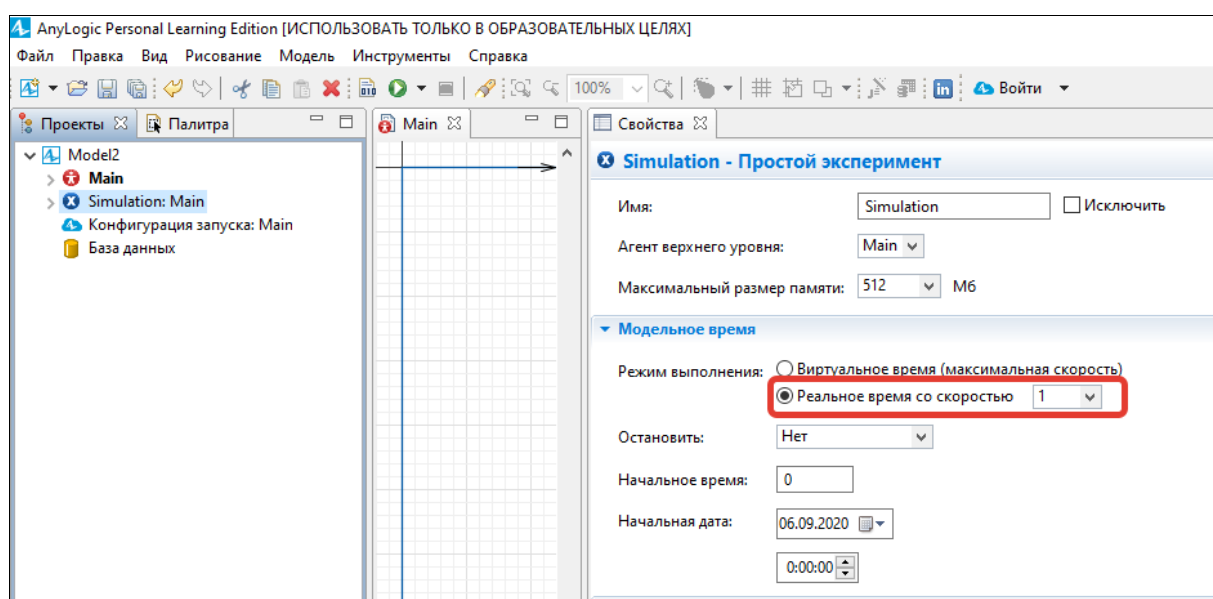


Рис. 4.81. Настройка свойств эксперимента

В режиме реального времени задается связь модельного времени с физическим временем, то есть задается количество единиц модельного времени, выполняемых в одну секунду. Режим реального времени лучше всего подходит для показа анимации. В режиме виртуального времени модель выполняется без привязки к физическому времени: она просто выполняется так быстро, насколько это возможно. Этот режим лучше всего подходит, когда требуется моделировать работу системы в течение достаточно длительного периода времени, например при показе модели заказчиком проекта.

Мы в основном будем пользоваться режимом реального времени, чтобы иметь возможность отображать анимацию модели с фиксированной скоро-

стью. Дополнительные свойства эксперимента позволяют управлять выполнением модели. Вы можете запустить модель так, чтобы она работала бесконечно, но можете и остановить ее в заданный момент времени. Вы можете остановить модель по достижении переменной заданного значения или по выполнении какого-нибудь определенного условия.

Если в модели используются случайные числа, то Вы по желанию можете сделать выбор – генерировать уникальные случайные числа (это нужно при сборе сравнительной статистики по нескольким запускам) или генерировать одинаковые числа и добиваться многократного запуска одной и той же модели (это нужно для изучения поведения модели при нескольких запусках).

Для построения модели с помощью библиотеки моделирования процессов нужно перетащить объекты с вкладки «**Палитра**» в окно редактора структуры, а затем задать требуемые свойства объектов и соединить их друг с другом.

Объекты легко соединяются друг с другом с помощью специальных точек и линий, которые называются **портами**. Достаточно подвести объект к другому объекту и эта линия будет построена. Но даже если этого не случилось, можно двойным кликом мыши щелкнуть по точке (порту) и провести линию к другой точке у объекта.

После создания и соединения объектов можно, кликнув по одному из объектов, задавать его свойства в правом окне.

В этом разделе мы создадим модель простой системы массового обслуживания, а именно модель банковского отделения. В банковском отделении находятся банкомат и стойки банковских кассиров, что позволяет быстро и эффективно обслуживать посетителей банка. Операции с наличностью клиенты банка производят с помощью банкомата, а более сложные операции, такие как оплата счетов, – с помощью кассиров.

Мы произведем оценку затрат операций и увидим, сколько денег тратится на обслуживание одного клиента и какую часть этой суммы составляют накладные расходы на оплату работы персонала банка, а какую – на обслуживание посетителей.

Создайте новую модель. В свойствах эксперимента «**Simulation**» задайте выполнение модели в режиме реального времени с выполнением одной единицы модельного времени в одну секунду. В этой модели под единицей модельного времени мы будем понимать одну минуту работы банковского отделения.

Создадим блок-схему модели, которая пока будет состоять только из банкомата. Для этого создайте и соедините объекты как показано на рис. 4.82.

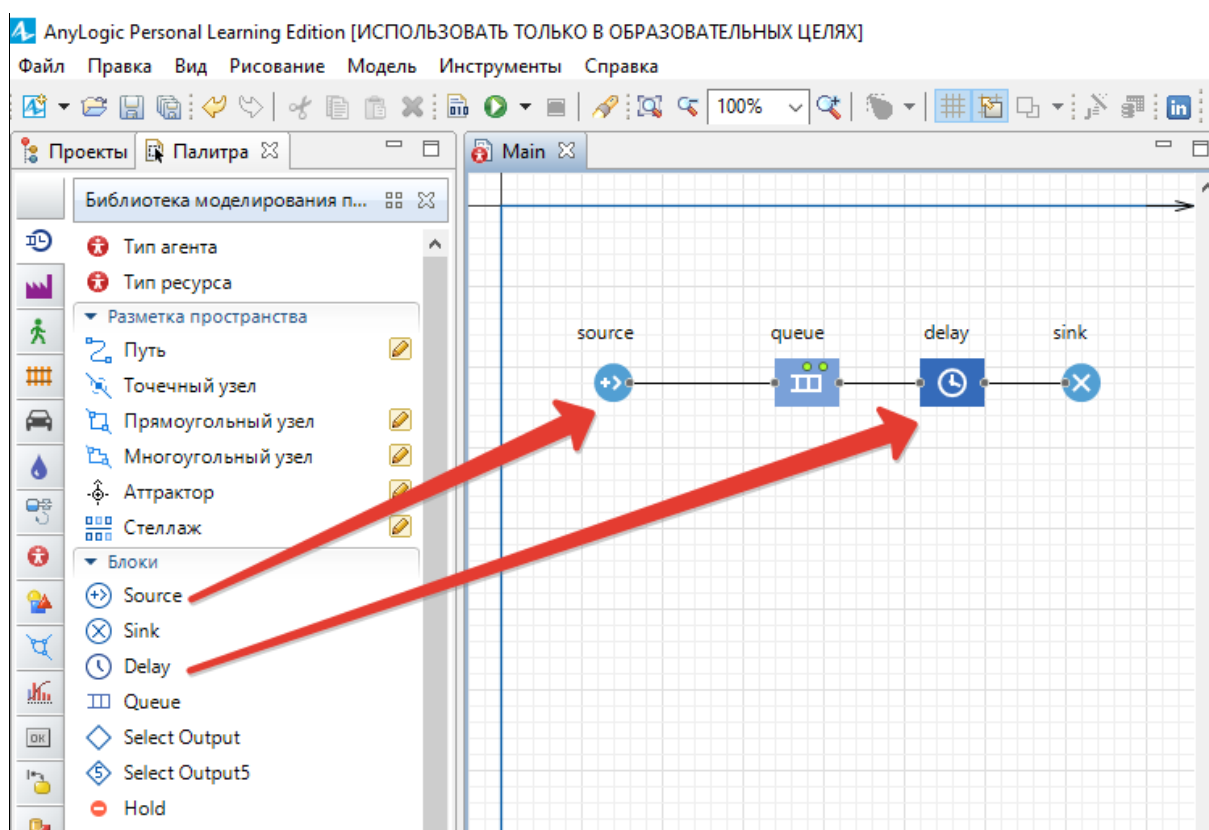


Рис. 4.82. Перенос объектов из палитры на область редактора структуры

Создайте блок-схему: добавьте объект «**Source**», добавьте объект «**Queue**», добавьте объект «**Delay**», добавьте объект «**Sink**».


Объект «**Source**» генерирует заявки (**entities**) определенного типа через заданный временной интервал. Заявки представляют собой объекты, которые производятся, обрабатываются, обслуживаются или еще каким-нибудь образом подвергаются действию моделируемого процесса: это могут быть клиенты в системе обслуживания, детали в модели производства, документы в модели


документооборота и т.д. В нашем примере заявками будут посетители банка, а объект **«Source»** будет моделировать их приход в банковское отделение.

Чтобы узнать детальное описание объектов библиотеки моделирования процессов, достаточно навести на них мышью в палитре и немного «зависнуть» на нем. Появится всплывающее окно, в котором представлено подробное описание объекта. Также описание можно получить с помощью главного меню **«Справка»** или по нажатию на клавишу **F1**.

Объект **«Queue»** моделирует очередь клиентов, ожидающих обслуживания. Объект **«Delay»** моделирует задержку, которая вызвана обслуживанием. В нашем примере он будет моделировать банкомат, тратящий определенное время на обслуживание клиента. Объект **«Sink»** обозначает конец блок-схемы и выполняет удаление обработанных заявок из системы.

Теперь мы можем запустить созданную модель. Для каждой модели автоматически создается блок-схема с наглядной визуализацией процесса, с помощью которой вы можете изучить текущее состояние модели, например: длину очереди, количество обслуженных человек и т.д.

Запустите модель и откройте окно визуализированной блок-схемы. Щелкните мышью по кнопке **«Запустить»** . Появится окно блок-схемы.

Щелкните мышью по другой кнопке **«Запустить»** . AnyLogic переключится в режим работы модели (рис. 4.83).

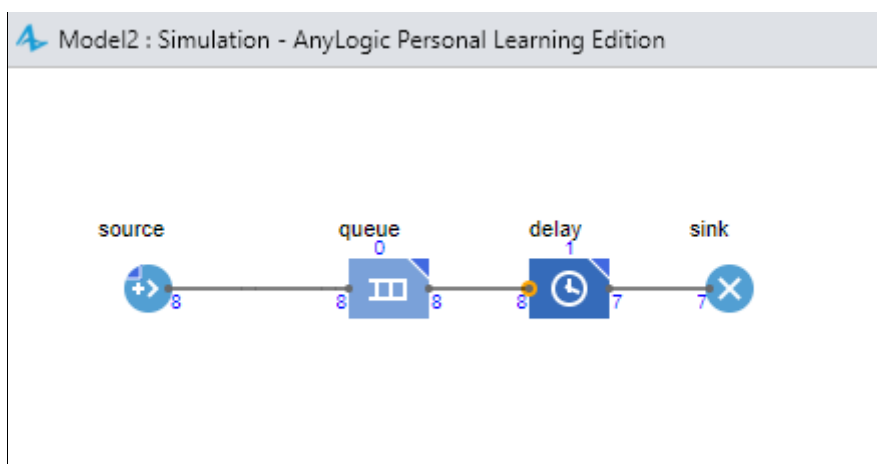


Рис. 4.83. Модель в запущенном состоянии

С помощью визуализированной блок-схемы можете проследить, сколько человек находится в очереди, сколько человек в данный момент обслуживается и т.д. Этому соответствуют числа, расположенные рядом с портами. 8 человек стоят в очереди, а 7 человек покинули очередь

Теперь, изменяя свойства созданных объектов, мы зададим данные нашей модели. Измените свойства объектов блок-схемы «**Source**», «**Queue**», «**Delay**».

Выделите объект «**Source**» и справа в окне «**Свойства**» установите параметр «**Прибывают согласно**» в значении «**Интенсивности**», параметр «**Интенсивность прибытия**» равным **exponential (0,67)** (рис. 4.84).

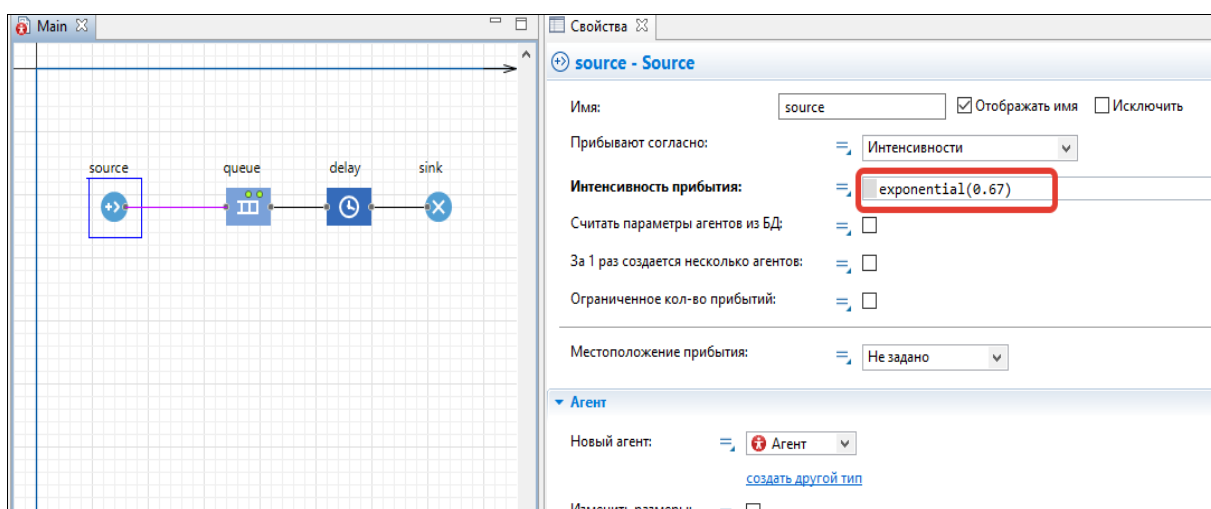


Рис. 4.84. Параметры для объекта «**Source**»

Интервал между приходом клиентов экспоненциально распределен со средним значением равным 1,5 единицам модельного времени. Заметьте, что аргумент функции **exponential ()** равен 0,67, потому что в качестве аргумента задается интенсивность прихода клиентов.

Функция **exponential ()** является стандартной функцией генератора случайных чисел AnyLogic. AnyLogic предоставляет функции и других случайных распределений, таких как нормальное распределение, равномерное распределение, треугольное распределение, и т.д. За детальным описанием функций

и их параметров можно обратиться к «Руководству пользователя» или «Справочнику классов» (смотрите методы класса «Func»).

Выделите объект «Queue» и в окне свойств задайте следующие значения (рис. 4.85).

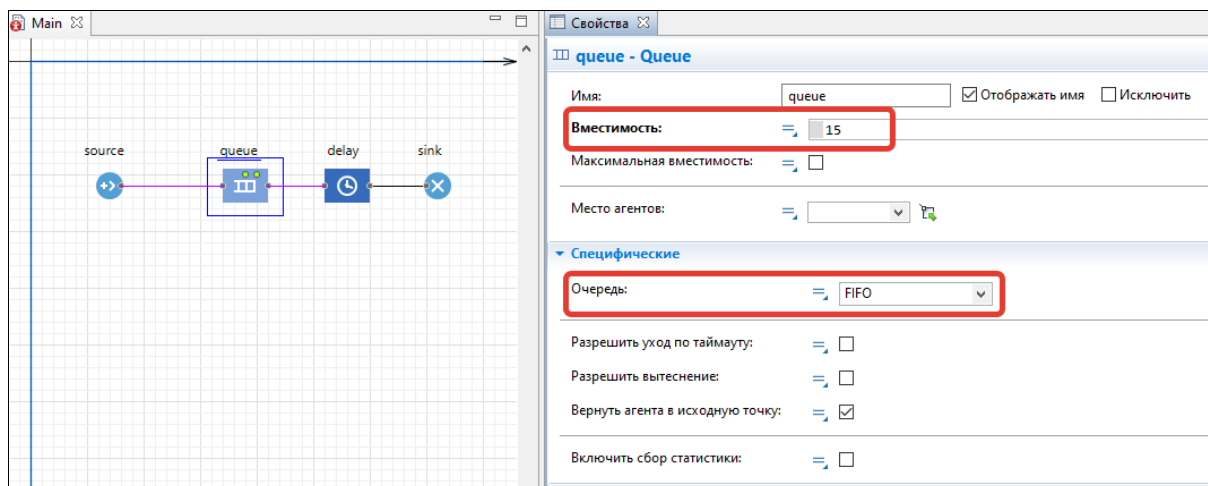


Рис. 4.85. Параметры для «Queue»

Задайте вместимость очереди – в ней будут находиться не более 15 человек. Также в окне свойств установите флажок «Включить сбор статистики».

Выберите следующий объект «Delay» и задайте для него свойства, как показано на рис. 4.86.

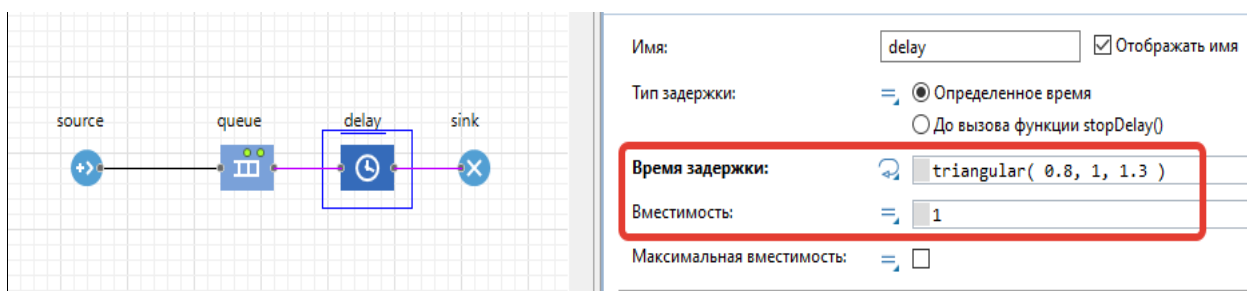



Рис. 4.86. Параметры объекта «Delay»

Введите время задержки **triangular (0.8, 1, 1.3)** и установите флажок «Включить сбор статистики».

Теперь можно запустить модель щелчком по кнопке «Запустить»  и провести эксперименты с моделью.

Визуализация процесса с помощью блок-схемы очень удобна, поскольку визуализированная блок-схема создается для каждой модели автоматически. В этом примере мы хотим создать визуализированный план банковского отделения. Для этого нам нужно создать анимационную диаграмму, а затем нарисовать на ней банкомат и очередь (рис. 4.87).

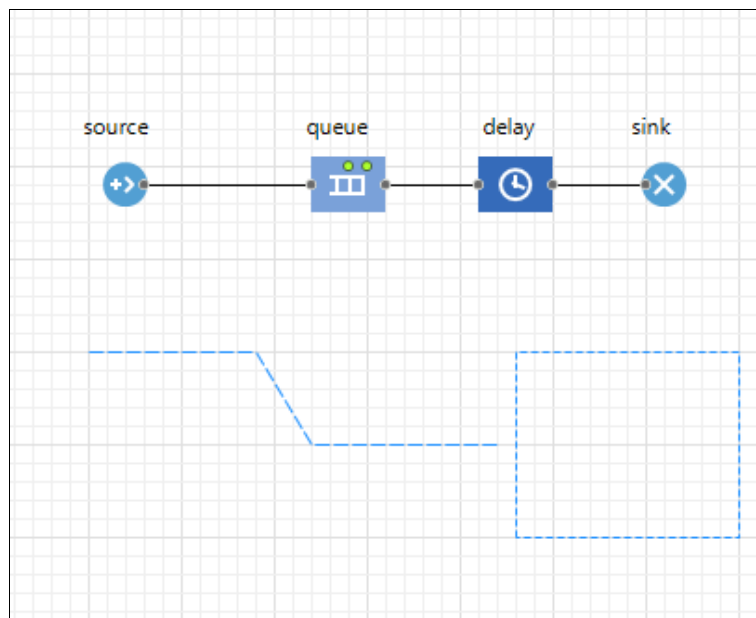


Рис. 4.87. Создание объектов для анимации банкомата и очереди

Нарисуйте банкомат и очередь. Нарисуйте очередь с помощью инструмента «Путь» (рис. 4.88).

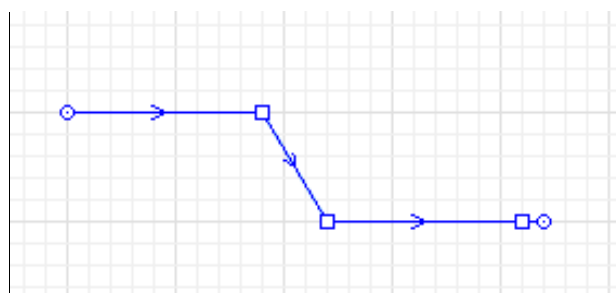
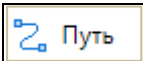


Рис. 4.88. Путь движения очереди

Для этого перейдите на вкладку «Палитра» и выберите раздел «Библиотека моделирования процессов». Щелкните двойным щелчком мыши по кнопке «Путь»  и отметьте первую точку щелчком мыши по диаграм-

ме. Щелкните в других местах диаграммы, чтобы добавить промежуточные точки. Последнюю точку добавьте двойным щелчком. Важно помнить, что рисовать нужно слева направо, при этом стрелками будет показано направление движения посетителей. После того, как нарисовали линию, задайте ей имя в окне свойств «GoToService».

Нарисуйте прямоугольник с помощью инструмента «Прямоугольный узел» «Палитра» | «Библиотека моделирования процессов» (рис. 4.89).

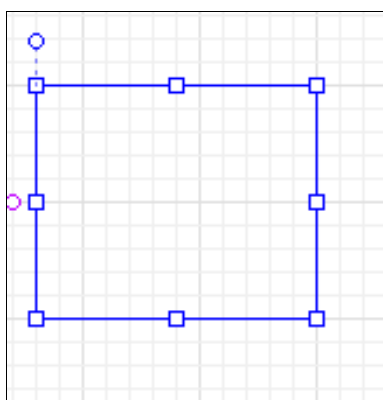


Рис. 4.89. Зона обслуживания

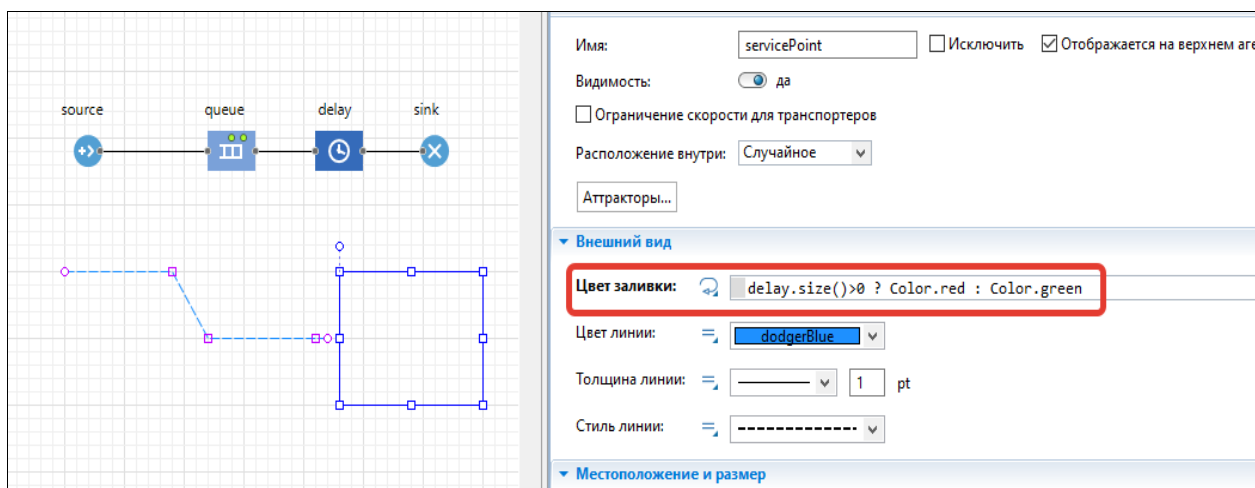



Рис. 4.90. Свойства для прямоугольника точки обслуживания

Назовите прямоугольник «**servicePoint**», задайте цвет заливки. Цвет будет меняться динамически, поэтому в поле «Цвет» с помощью кнопки  нужно выбрать «Динамическое значение» (рис. 4.90). В поле для изменения цвета нужно ввести строчку кода:

```
delay.size ()>0 ? Color.red : Color.green
```


Здесь функция **size ()** объекта **«Delay»** возвращает число человек, обслуживаемых в данный момент времени. Если банкомат занят, то цвет прямоугольника будет красным, в противном случае – зеленым. **«Color»** – это класс **«Java»**, позволяющий использовать стандартные цвета (черный, синий, красный, голубой, желтый и т.д.), и создавать любые другие.

Теперь мы должны задать созданные анимационные объекты в качестве анимационных фигур объектов блок-схемы нашей системы.

Снова выделите объект **«Delay»** и откройте его окно свойств. В поле **«Место агентов»** укажите из списка **«servicePoint»**. Проверьте (или укажите), что в разделе **«Специфические»** стоит флажок у параметра **«Одиночный агент»**.

Выделите объект **«Queue»** на диаграмме процесса. В окне свойств укажите в поле **«Место агентов»** значение **«GoToService»**.

Установим некоторые параметры эксперимента для данной модели. В окне **«Проекты»** выделите кликом объект **«Simulation»**, в окне свойств в разделе **«Модельное время»** установите **Реальное время со скоростью 5**. Тем самым мы ускорим модель в 5 раз. Запустите модель (рис. 4.91).

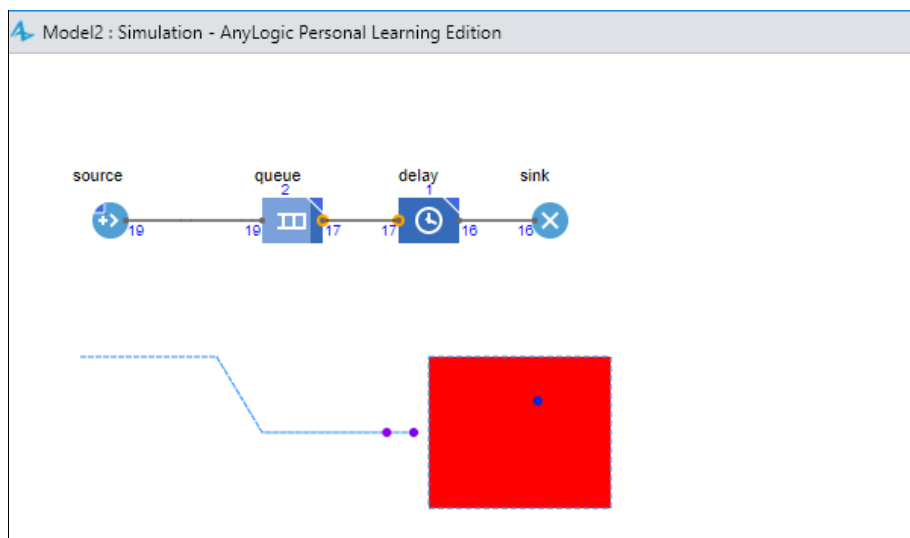




Рис. 4.91. Модель в состоянии выполнения

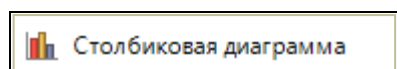
Щелкните по кнопке **«Запустить»** . Чтобы остановить выполнение, щелкните по кнопке **«Остановить»** . Модель начнет выполнение, и появится окно анимации. Запустив модель, вы увидите окно анимации. Цвет прямоугольника будет меняться в зависимости от того, обслуживается ли клиент

в данный момент времени AnyLogic позволяет производить сбор сложной статистики. Для этого нужно лишь включить у объекта режим сбора статистики, поскольку по умолчанию он отключен для повышения скорости выполнения модели.

Включите сбор статистики: для элементов **«Queue»** и **«Delay»** в окне свойств должен стоять флажок **«Включить сбор статистики»**.

Вы можете просмотреть собранную статистику с помощью диаграмм и графиков. Покажем статистику занятости банкомата с помощью индикатора.

Добавьте **индикатор** на анимацию. Для этого перейдите на вкладку **«Палитра»** в раздел **«Статистика»**. Найдите элемент **«Столбиковая диаграмма»**



и перенесите этот объект на свободное поле в окне структуры проекта. С помощью окна свойств установите **«Направление столбцов»** горизонтальное (рис. 4.92).

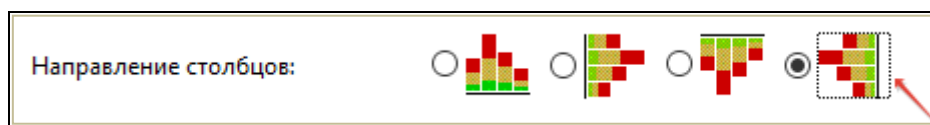


Рис. 4.92. Установка направления столбцов диаграммы

На вкладке **«Данные»** оставьте только один элемент с данными. В поле **«Заголовок»** установите значение **«Длина Очереди»**, а в поле **«Значение»** задайте **«queue.statsSize.mean ()»**. Здесь метод **«mean ()»** – возвращает среднюю длину очереди.

Вторую диаграмму расположите рядом с изображением банкомата. Назовите диаграмму **«Обслуживание»**, а в качестве значения задайте выражение: **«delay.statsUtilization.mean ()»**, задающее среднее время обслуживания заявки в процессоре. Направление столбцов вертикальное.

Запустите модель на выполнение. Вид получившейся модели показан на рис. 4.93.

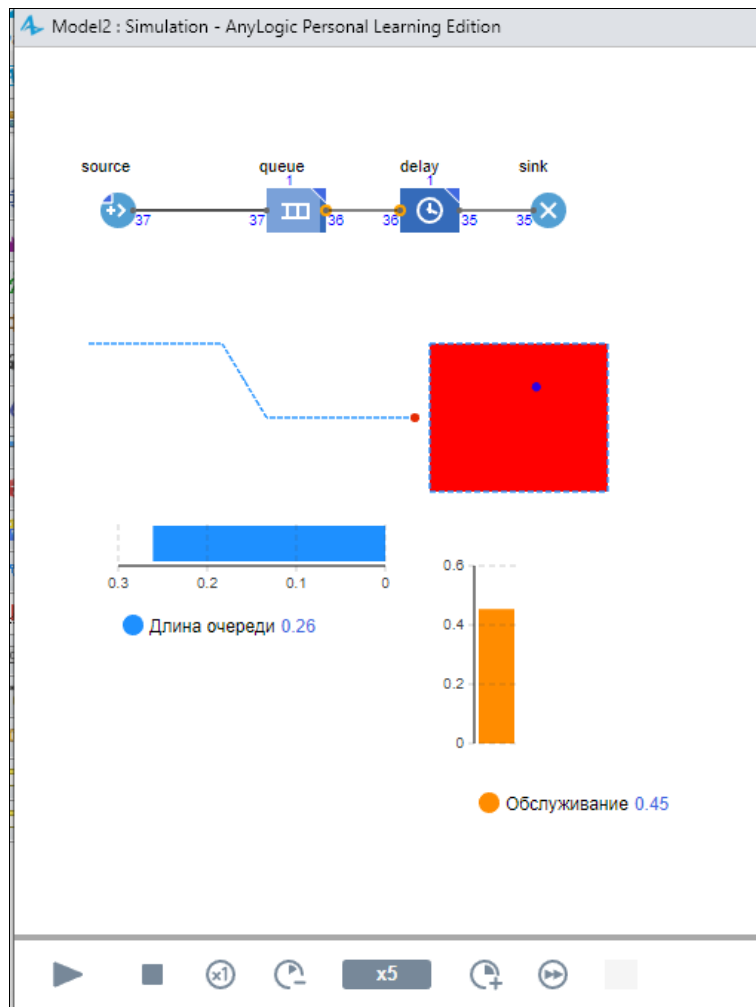



Рис. 4.93. Модель банка в режиме запуска с диаграммами

Для ускорения работы модели, переключитесь в режим виртуального времени . В режиме виртуального времени модель будет выполняться с максимально возможной скоростью.

Вы научились основам создания дискретно-событийных моделей с применением библиотеки моделирования процессов. Теперь мы усложним нашу модель, добавив в нее служащих – банковских кассиров. Мы могли бы смоделировать кассиров, как и банкомат, с помощью объектов «Delay». Но более удобным представляется моделирование кассиров с помощью «Ресурсов».

«Ресурс» – это специальный объект библиотеки моделирования процессов, который может потребоваться заявке для выполнения какой-то задачи. В нашем примере посетителям банковского отделения (заявкам) необходимо получить помощь у банковских служащих (которые моделируются ресурсами).

Для моделирования второго канала нужно использовать следующие элементы палитры библиотеки моделирования процессов:

- **SelectOutput** – моделирование процесса выбора клиентом канала обслуживания.
- **Service** – узел обслуживания клиентов, состоящий из очереди и прибора обслуживания «Ddelay».
- **ResourcePool** – источник ресурсов.

Используем ранее созданную модель офиса с анимацией очереди к банкомату и построим систему обслуживания в соответствии с рис. 4.94.

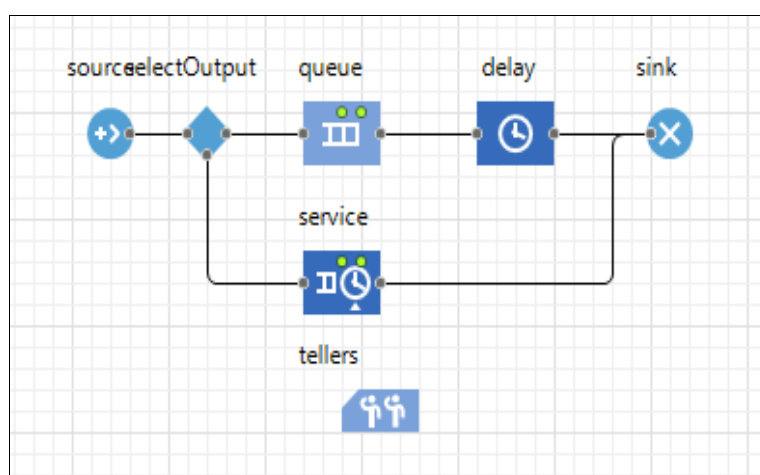


Рис. 4.94. Двухканальная модель банковского офиса

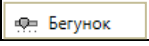
Напомним, что связи, соединяющие объекты, можно удалять и создавать новые. Чтобы создать новую связь, нужно выделить двойным щелчком мыши зеленую точку у объекта и провести линию до другой такой зеленой точки. Чтобы создать изгиб соединительной линии, достаточно кликнуть по полю. Чтобы завершить редактирование соединительной линии, нужно сделать двойной щелчек мышкой.

Объект «**SelectOutput**» является блоком принятия решения. В зависимости от заданного условия, заявка, поступившая в объект, будет поступать на один из двух выходов объекта. У данного объекта в окне свойств установите в поле «**Вероятность**» значение равное 0,5.

Для объекта «**Service**» задайте время задержки, которое соответствует треугольному закону распределения «**triangular (2.5, 6, 11)**».

что будет означать предпочтительное значение 6; «Вместимость очереди» – 20 клиентов. Для элемента включите сбор статистики.

Для объекта «ResourcePool» задайте имя «Tellers». «Количество ресурсов» равно 4. Это количество консультантов в системе. Включите сбор статистики.

Рядом с объектом «Tellers» поместите элемент бегунок  (найти его можно на палитре объектов в разделе «Элементы управления») для текущего выбора количества консультантов.

Свойства элемента (рис. 4.95):

- минимальное значение равно нулю;
- максимальное значение равно 4;
- элемент должен быть связан со свойством «capacity» (число ресурсов)

элемента «Tellers».

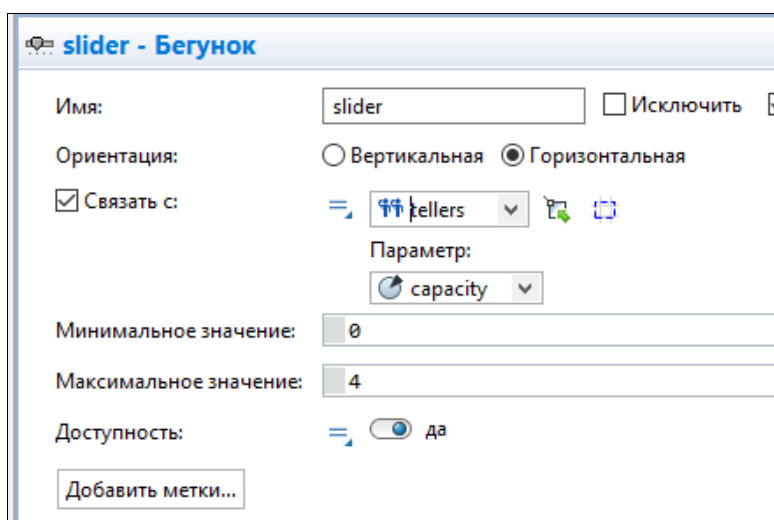


Рис. 4.95. Установка свойств для бегунка

Запустите модель и изучите ее поведение. Изменяйте положение бегунка и посмотрите, как это влияет на длину очереди посетителей. Измените значение задержки очереди обслуживания (параметр «Время задержки» у объекта «Delay») на значение «triangular (0,8, 1,5, 3,5)». Как это повлияло на модель?

4.7. ANYLOGIC-АГЕНТНАЯ МОДЕЛЬ

Агентное моделирование позволяет решать задачи, которые трудно описать математическими методами. Они состоят из набора взаимодействующих отдельных объектов (агентов), каждый из которых имеет какие-то заданные правила поведения и на основе поступающих сведений принимает решение об их применении. Например, такой агент может описывать поведение особи в стаде или человека в толпе. Поведение всей системы будет складываться как результат взаимодействия агентов, но спрогнозировать его математическим соотношением трудно. Например, поведение толпы на выходе из здания.

В AnyLogic содержится специальная пешеходная библиотека, с помощью которой можно создавать модели, имитирующие движение людских потоков. Это может быть актуально при изучении узких мест при проходах в здания, загруженность транспортных потоков, изучение планов эвакуации и другое.

Рассмотрим задачу, описанную в книге И.А. Калинина «Информатика. Углубленный уровень: задачник-практикум для 10–11 классов».

В некоторой школе планируется повысить уровень обеспечения безопасности. В первую очередь, специалисты рекомендовали ограничить проход в школу посторонних, для чего на входе установят два турникета и начнут проверять всех входящих с помощью специальной карты. По мнению директора школы, реальное применение такого подхода приведет к существенным затруднениям перед началом занятий. Как подтвердить или проверить эти предположения?

Создадим модель, показывающую первый этаж школы с турникетами и агентами пешеходами, единственной задачей которых будет пройти через турникет. При этом мы учтем, что проход через турникет (а точнее, проверка) занимает некоторое время. Время от времени будет попадаться человек без документов (например, без карточки), который будет возвращаться назад. Время прохода может меняться (человек проверяет документы медленнее, чем автомат – карточки), вероятность появления «неправильного» документа тоже. Чтобы модель выглядела реалистично, найдем и загрузим типовой план первого этажа школы. Авторы использовали план из типового проекта школы 65-426/1. Найти его можно в Интернете и сохранить на своем компьютере.

Запустим AnyLogic, создадим новый проект, как мы это делали на предыдущих лабораторных работах. На втором шаге создания проекта укажем, что модель создаем с нуля, не используя заготовки.

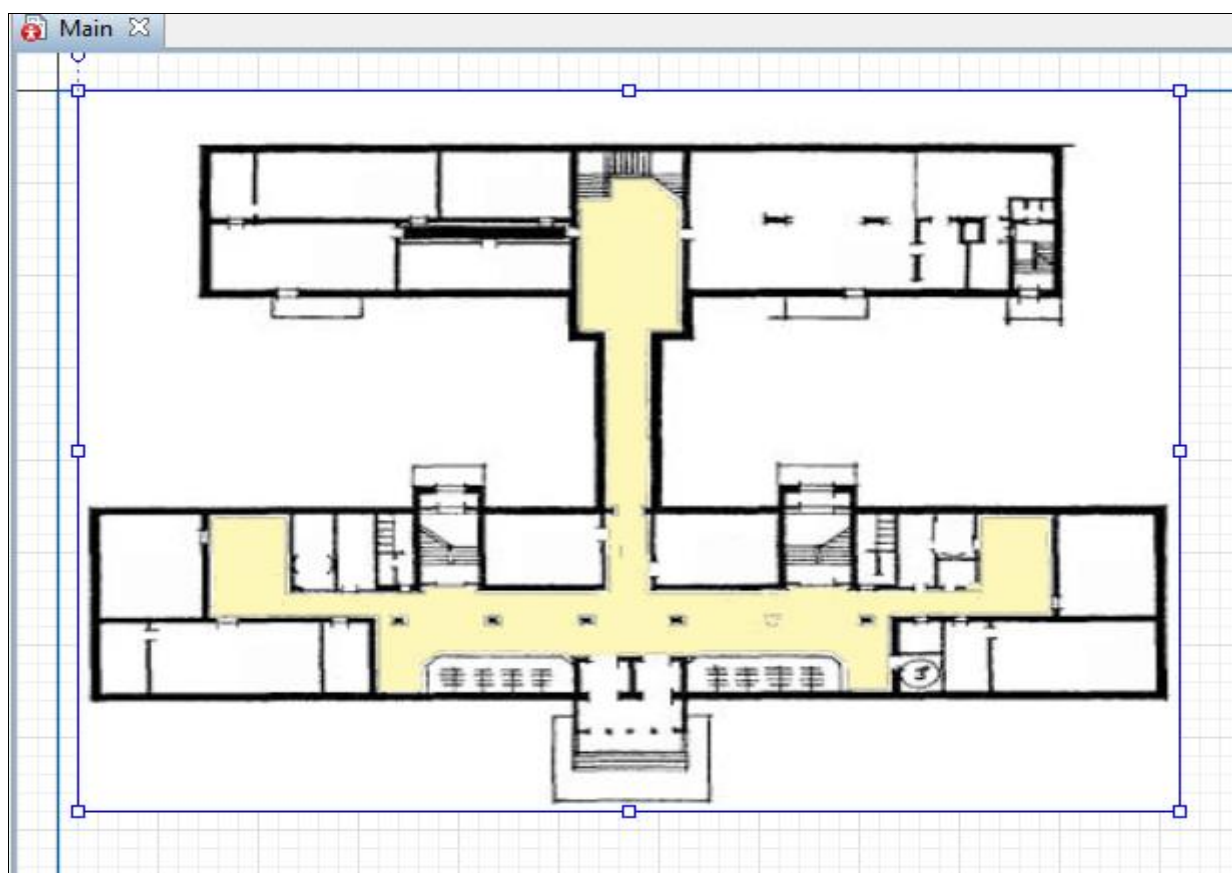

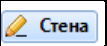


Рис. 4.96. Создание плана школы

Возьмем за основу типовой проект школы 65-426/1 (рис. 4.96). Найдем в сети план первого этажа, очистим его от лишних артефактов и разместим на рабочем листе. Для этого перейдите на «Палитру», раздел «Презентация», перенесите на рабочее поле объект «Изображение». После переноса откроется диалоговое окно, в котором нужно выбрать файл с планом этажа. Если это диалоговое окно не открылось, то можно в свойствах объекта «Изображение» нажать на кнопку  **Добавить**.

Заблокируйте изображение, установив флажок «Блокировать» в свойствах изображения. Вы не сможете выбрать заблокированную фигуру в графическом редакторе до тех пор, пока вы не снимете с нее блокировку. Мы делаем так потому, что мы будем рисовать другие фигуры поверх этого изображения, и по-

этому мы хотим исключить возможность случайного редактирования изображения при рисовании этих фигур. Примерное расположение показано на рис. 4.96.

Теперь прямо поверх плана мы отметим стены (поскольку на картинке система выделить стены автоматически не может). Для этого перейдите на «**Палитре**» в раздел «**Пешеходная библиотека**» и выберите объект «**Стена**». Двойным кликом перейдите режим редактирования стен . Последовательно щелкайте мышью в тех точках диаграммы, куда вы хотите поместить углы стены. Каждый щелчок добавляет часть линии той стены, которую вы рисуете.

Чтобы добавить кривую линию, щелкните левой кнопкой мыши в точку конца кривой линии и двигайте указатель мыши, удерживая кнопку. Пока вы ведете указатель мыши, вы заметите, как изменяется радиус кривизны. Чтобы нарисовать окружность, двигайте указателем ровно вдоль сетки координат. Отпустите левую кнопку мыши, когда рисунок готов.

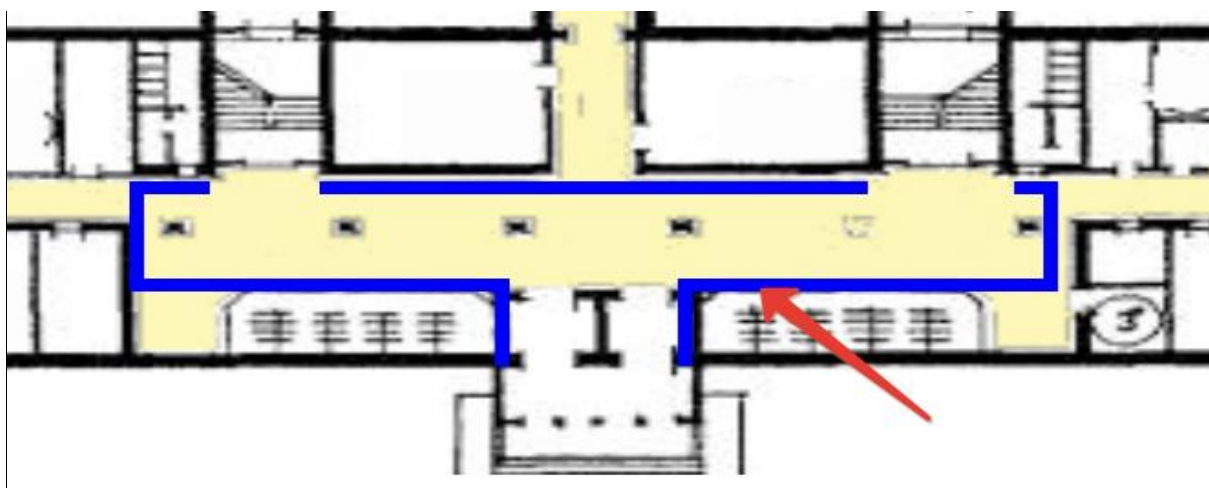
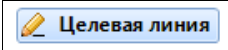


Рис. 4.97. Стены в плане школы

Чтобы завершить рисование, добавьте последнюю точку стены двойным щелчком мыши (рис. 4.97). Нарисуйте стены, как показано на рис. 4.97. Там, где в школе предполагаются выходы на другие этажи, сделайте разрывы стен. В этом случае двойным кликом нужно завершить редактирование предыдущей линии и затем нарисовать новую стену. Выделите все стены с помощью зажатой клавиши «**Shift**» и кликая на каждую стену. В свойствах линий в разделе «**Внешний вид**» укажите цвет линий синий, толщина 4 pt.

Ученики будут приходить в школу со стороны крыльца и за это будет отвечать специальный объект «**Целевая линия**». Найдите ее на «**Палитре**» в раз-

деле «Пешеходная библиотека». Двойным кликом по объекту «Целевая линия» переведите его в режим редактирования . Поместите перед входом, как показано на рис. 4.98 и в свойствах задайте имя «Enter».

С помощью объекта «Целевая линия» разместите две линии в районе переходов на верхние этажи. Эти линии назовем: левый выход – «Exit1», правый – «Exit2» (рис. 4.98).

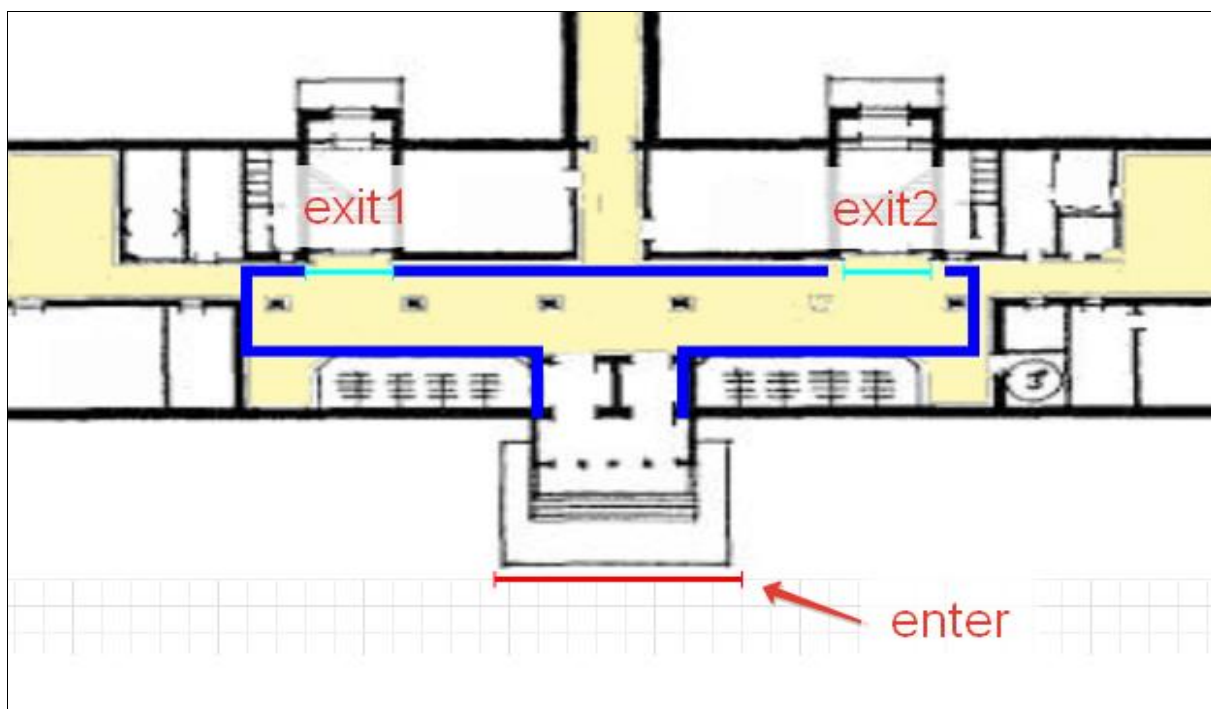


Рис. 4.98. Создание входа и выходов на плане школы

Построим логику движения пешеходов. Перенесите в редактор следующие блоки: «pedSource», «pedSelectOutput», «pedGoTo», «pedSink» – из «Палитры», раздела «Пешеходная библиотека» и с помощью портов соедините, как показано на рис. 4.99. Если автоматически соединительные линии не получилось создать или линия соединилась не с тем портом, то ее можно выделить кликом мыши и, нажав на кнопку «Delete», удалить. Новый порт рисуем от одной точки (двойной клик по точке) до другой точки (нужно попасть прямо в зеленую точку порта). В свойствах данных объектов укажите те названия, как предложено на рисунке, чтобы в дальнейшем не запутаться.

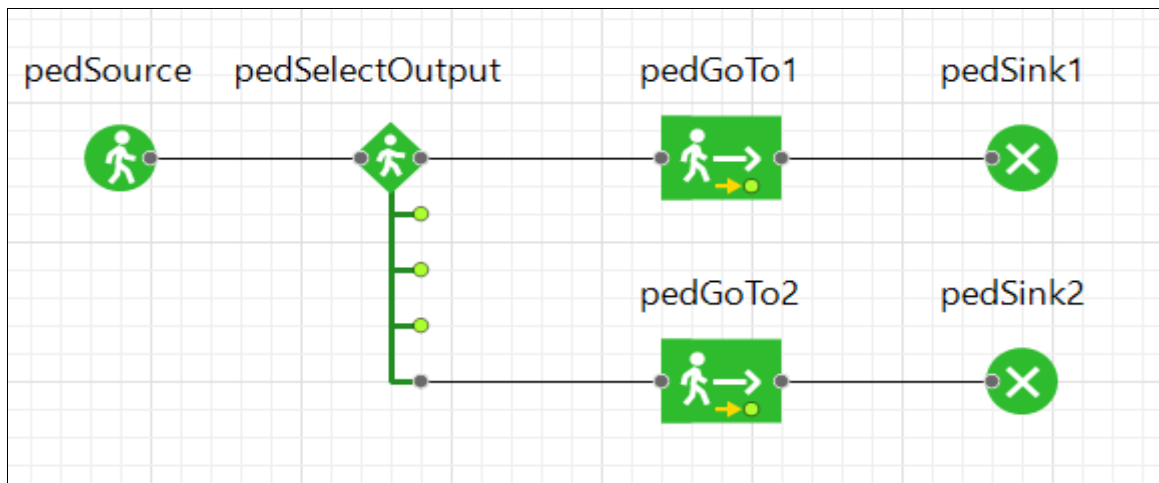


Рис. 4.99. Логическая схема движения пешеходов

Рассмотрим указанные блоки подробнее.

- Объект **«pedSource»** создает пешеходов. Обычно он используется в качестве начальной точки диаграммы процесса, формализующей поток пешеходов.
- Объект **«pedSelectOutput»** направляет входящих в объект пешеходов на один из пяти выходных портов. Выходной порт будет выбираться: либо в соответствии с заданными коэффициентами предпочтения, либо в зависимости от того, для какого из этих портов будет выполнено заданное условие.
- Объект **«pedGoTo»** моделирует перемещение пешеходов из текущего местоположения в другое (заданное параметром этого объекта). С помощью этого объекта мы будем моделировать то, как пешеходы перемещаются от входа в школу далее к турникету.
- Объект **«pedSink»** удаляет поступивших в объект пешеходов из моделируемой среды. Обычно объект используется в качестве конечной точки диаграммы процесса.

Изменим свойства блоков данной диаграммы. Для объекта **«pedSource»** в свойствах укажите:

Место появления – линия.

Целевую линию выберите из списка **«Enter»**,

Прибывают пешеходы согласно интенсивности,

Интенсивность прибытия пешеходов 1 200 в час,

Количество прибывших пешеходов ограничено (галочка),

Максимальное количество прибывших пешеходов 600.

Объект **«pedSelectOutput»** использует вероятности переходов школьников на тот или иной выход с этажа. Параметр **«Коэффициент предпочтения»** у соединенных портов объекта **«pedSelectOutput»** задайте равным 0.5, у свободных портов – 0.

Объект **«pedGoto1»**

Режим: достичь цели.

Цель: линия.

Целевая линия: выбрать из списка **«Exit1»**.

Объект **«pedGoto2»**

Режим: достичь цели.

Цель: линия.

Целевая линия: выбрать из списка **«Exit2»**.

Свойства для **«pedSink1»** и **«pedSink2»** менять не будем.

Установите на вкладке **«Проекты для объекта «Simulation»** в окне свойств параметр **«Реальное время»** со скоростью 10 (рис. 4.100).

Запустите проект и посмотрите, что получилось.

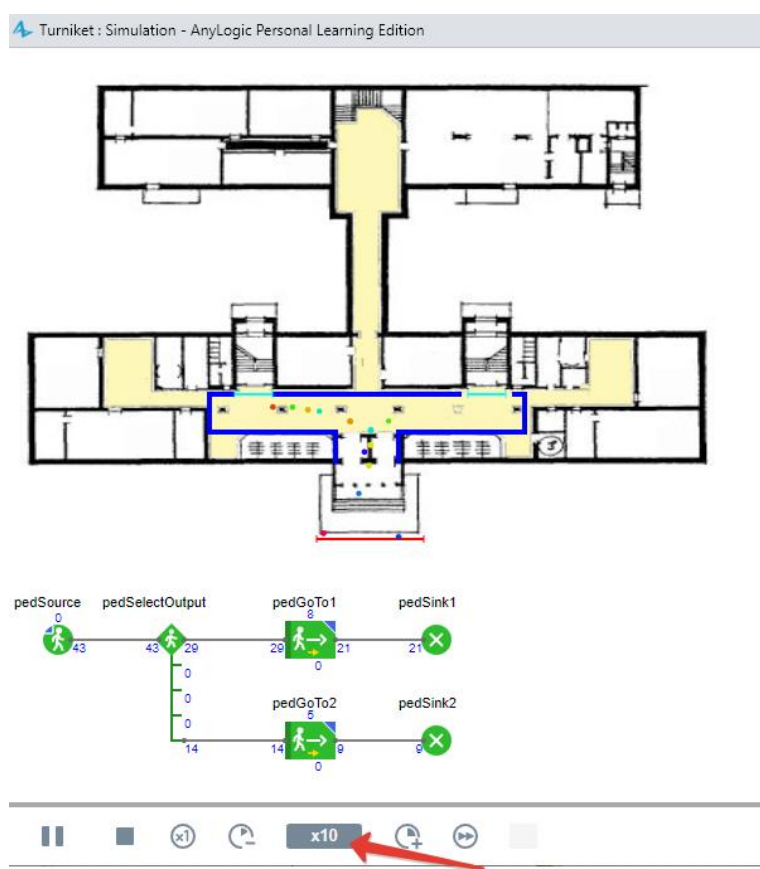


Рис. 4.100. Проект с турникетами в режиме запуска

Результат будет вполне предсказуемым – никаких препятствий нет. 600 человек попадают внутрь примерно за 30 минут. По дороге нет затруднений, время зависит только от частоты появления и скорости движения.

Дополнительные теоретические материалы по ООМ можно найти в монографиях: [11; 13; 24–27; 47], перечисленных в библиографическом списке.

4.8. КОНТРОЛЬНЫЕ ВОПРОСЫ К ГЛАВЕ 4

1. Что такое RMD?
2. В чем суть ООМ?
3. Что такое гибридная модель?
4. Что такое визуальная модель?
5. Каково назначение AnyLogic?
6. Опишите парадигмы моделирования в AnyLogic.
7. Что такое модель системной динамики?
8. Что такое агентная модель?
9. Что такое событийная модель?
10. Какую роль играет стрейтчарт в моделях AnyLogic?
11. Перечислите особенности систем массового обслуживания (СМО).
12. Как строится анимация процесса обслуживания заявок в AnyLogic?
13. Как определить среднее число заявок в очереди в модели AnyLogic?
14. Как определить среднее число заявок обслуженных процессором в модели AnyLogic?
15. Как строится модель AnyLogic в которой выполняется выбор канала обслуживания?
16. Перечислите особенности построения анимации канала обслуживания с ресурсами.
17. Перечислите основные этапы создания гистограмм в модели СМО.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

НАУЧНАЯ ЛИТЕРАТУРА

1. Арнольд, В.И. «Жесткие» и «мягкие» математические модели / В.И. Арнольд. – Москва: МЦНМО, 2000. – 32 с. – ISBN 5-94057-4.
2. Введение в математическое моделирование / под редакцией П.В. Трусова. – Москва: Логос, 2007. – 440 с.
3. Голованов, Н.Н. Геометрическое моделирование / Н.Н. Голованов. – Москва: Физматлит, 2002. – 472 с. – ISBN 5-94052-048-0.
4. Мышкис, А.Д. Элементы теории математических моделей / А.Д. Мышкис. – Москва: КомКнига, 2007. – 192 с. – ISBN 978-5-484-00953-4.
5. Неуймин, Я.Г. Модели в науке и технике / Я.Г. Неуймин. – Ленинград: Наука, 1984. – 283 с.
6. Острейковский, В.А. Теория систем / В.А. Острейковский. – Москва: Высшая школа, 1997. – 240 с. – ISBN 5-06-002642-6.
7. Плохотников, Э.К. Математическое моделирование и вычислительный эксперимент. Методология и практика / Э.К. Плохотников. – Москва: Едиториал УРСС, 2003. – 280 с. – ISBN 5-35400521-3.
8. Самарский, А.А. Математическое моделирование: Идеи. Методы. Примеры / А.А. Самарский, А.П. Михайлов. – Москва: Физматлит, 2005. – 320 с. – ISBN 5-9221-0120-х.

УЧЕБНАЯ ЛИТЕРАТУРА

9. Бенькович, Е.С. Практическое моделирование динамических систем / Е.С. Бенькович, Ю.Б. Колесов, Ю.Б. Сенюченков. – Санкт Петербург: БХВ-Петербург, 2002. – 464 с. – ISBN 5-94157-099-6.
10. Бережная, Е.В. Математические методы моделирования экономических систем / Е.В. Бережная, В.И. Бережной. – Москва: Финансы и статистика, 2003. – 368 с. – ISBN 5-279-02291-8.

11. Григорьев, И.И. AnyLogic за три дня. Практическое пособие по имитационному моделированию / И.И. Григорьев // Интернет портал фирмы AnyLogic: [сайт]. – URL: <https://www.anylogic.ru/resources/books> (дата обращения: 12.12.2019) 273 с.
12. Казиев, В.М. Введение в анализ, синтез и моделирование систем / В.М. Казиев. – Москва: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2006. – 244 с. – ISBN 5-9556-0060-4.
13. Карпов, Ю.Г. Имитационное моделирование систем. Введение в моделирование с AnyLogic / Ю.Г. Карпов. – Санкт-Петербург: БХВ-Петербург, 2005. – 400 с. – ISBN 5-94157-148-8.
14. Каталевский, Д.Ю. Основы имитационного моделирования и системного анализа в управлении / Д.Ю. Каталевский. – Москва: ДЕЛО РАНХ и ГС, 2015. – 485 с. – ISBN 976-5-7749-1072-4.
15. Королев, А.Л. Компьютерное моделирование / А.Л. Королев. – Москва: ЛБЗ-БИНОМ, 2010. – 232 с. – ISBN 978-5-94774-487-3.
16. Неймарк, Ю.И. Математическое моделирование как наука и искусство: учебник / Ю.И. Неймарк. – Нижний Новгород: Изд-во Нижегородского госуниверситета, 2010. – 420 с. – ISBN 978-5-91326-145-8.
17. Перегудов, Ф.И. Введение в системный анализ / Ф. И. Перегудов, Ф.П. Тарасенко. – Москва: Высшая школа, 1989. – 367 с. – ISBN 5-063-001569-6.
18. Потемкин, А.М. Трехмерное твердотельное моделирование в системе КОМПАС-3D / А.М. Потемкин. – Санкт-Петербург: БХВ-Петербург, 2004. – 512 с. – ISBN 5-94157-472-х.
19. Сиденко, Л.Н. Компьютерная графика и геометрическое моделирование / Л.Н. Сиденко. – Санкт-Петербург: Питер, 2009. – 224 с. – ISBN 978-5-388-00339-3.
20. Советов, Б.Я. Моделирование систем / Б.Я. Советов, С.А. Яковлев. – Москва: Высшая школа, 2007. – 343 с. – ISBN 5-06-003860-2.
21. Советов, Б.Я. Моделирование систем. Лабораторный практикум / Б.Я. Советов, С.А. Яковлев. – Москва: ЮРАЙТ, 2016. – 127с. – ISBN-978-5-9916-2858-7.
22. Тарасевич, Ю.Ю. Математическое и компьютерное моделирование. Вводный курс / Ю. Ю. Тарасевич. – Москва: Едиториал УРСС, 2004. – 440 с. – ISBN 5-354-00913-8.

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

23. Айвазян, С.А. Прикладная статистика: Основы моделирования и первичная обработка данных / С.А. Айвазян, И.С. Енюков, Л.Д. Мешалкин. – Москва: Финансы и статистика, 1983. – 471 с.

24. Колесов, Ю.Б. Объектно-ориентированное моделирование сложных динамических систем / Ю.Б. Колесов. – Санкт Петербург: Изд-во СПбГТУ, 2004. – 240 с.

25. Колесов, Ю.Б. Моделирование систем. Динамические и гибридные системы / Ю.Б. Колесов, Ю.Б. Сениченков. – Санкт Петербург: БХВ-Петербург, 2006. – 224 с. – ISBN 5-94157-578-5.

26. Колесов, Ю.Б. Моделирование систем. Объектно-ориентированный подход / Ю.Б. Колесов, Ю.Б. Сениченков. – Санкт Петербург: БХВ-Петербург, 2006. – 192 с. – ISBN 5-94157-579-3.

27. Колесов, Ю.Б. Моделирование систем: Практикум по компьютерному моделированию / Ю.Б. Колесов, Ю.Б. Сениченков. – Санкт-Петербург: БХВ-Петербург, 2007. – 352 с. – ISBN 5-94157-580-7.

28. Компьютеры, модели, вычислительный эксперимент. Введение в информатику с позиций математического моделирования / под ред. А.А. Самарского. – Москва: Наука, 1988. – 176 с. – ISBN 5-02-007143-9

29. Коротаев, А.В. История и синергетика. Математическое моделирование социальной динамики / А.В. Коротаев, Ю.С. Малков. – Москва: КомКнига, 2005. – 192 с. – ISBN 5-484-00239-7

30. Леоненков, А.В. Решение задач оптимизации в среде MS Excel / А.В. Леоненков. – Санкт Петербург: БХВ-Петербург, 2005. – 704 с. – ISBN 5-94157-503-3.

31. Макарова, Н.В. Статистика в Excel / Н.В. Макарова, В.Я. Трофимец. – Москва: Финансы и статистика, 2002. – 368 с. – ISBN 5-279-02282-9.

32. Моисеев, Н.Н. Математические задачи системного анализа / Н.Н. Моисеев. – Москва: Наука, 1981. – 488 с.

33. Налимов, В.В. Теория эксперимента / В.В. Налимов. – Москва: Наука, 1971. – 208 с.

34. Налимов, В.В. Логические основания планирования эксперимента / В.В. Налимов, Т.И. Голикова. – Москва: Металлургия, 1980. – 152 с.

35. Павловский, Ю.Н. Имитационное моделирование / Ю.Н. Павловский, Н.В. Белотелов, Ю.П. Бродский. – Москва: Академия, 2008. – 240 с. – ISBN 978-5-7695-3967-1.

36. Пасконов, В.М. Численное моделирование процессов тепло и массообмена / В.М. Пасконов, В.И. Полежаев, Л.А. Чудов. – Москва: Наука, 1984. – 286 с.

37. Самарский, А.А. Численные методы / А.А. Самарский, А.В. Гулин. – Москва: Наука, 1989. – 432 с.

38. Строгалев, В.П. Имитационное моделирование / В.П. Строгалев, И.О. Толкачева. – Москва: Издательство МГТУ им. Н.Э. Баумана, 2008. – 280 с. – ISBN 978-5-7038-3021-5

39. Уемов, А.И. Системный подход и общая теория систем / А.И. Уемов. – Москва: Мысль, 1978. – 257 с.

40. Фарлоу, С. Уравнения с частными производными для научных работников и инженеров / С. Фарлоу. – Москва: Мир, 1985. – 384 с.

41. Холодниок, М. Методы анализа нелинейных математических моделей / М. Холодниок, А. Клич, М. Кубичек, М. Марек. – Москва: Мир, 1991. – 368 с. – ISBN 5-03-001849-2.

ИНТЕРНЕТ ИСТОЧНИКИ

42. Афонин, В.Б. Моделирование систем / В.Б. Афонин, С.С. Федосин. – URL: <https://www.intuit.ru/studies/courses/623/479/info>

43. Боев, В.В. Компьютерное моделирование / В.В. Боев, Р.А. Сыпченко. – URL: <https://www.intuit.ru/studies/courses/643/499/info>

44. Глухов, Д.О. Моделирование систем управления: практикум / Д.О. Глухов, И.В. Петухов; под ред. Д.О. Глухова. – Йошкар-Ола: Поволжский государственный технологический университет, 2015. – 84 с. – URL: <http://www.iprbookshop.ru/75437.html>. – ISBN 978-5-8158-1546-9.

45. Данелян, Т.Я. Теория систем и системный анализ: учеб. пособие / Т.Я. Данелян. – Москва: Евразийский открытый институт, 2011. – 303 с. – ISBN 978-5-374-00324-6. – URL: <http://www.iprbookshop.ru/10867.html>.

46. Данилов, А.М. Математическое и компьютерное моделирование сложных систем: учеб. пособие / А.М. Данилов, И.А. Гарькина, Э.Р. Домке

АСВ, 2011. – 296 с. – ISBN 978-5-9282-0733-5 – URL: <http://www.iprbookshop.ru/23100.html>.

47. Ефромеева, Е.В. Имитационное моделирование: основы практического применения в среде AnyLogic: учебное пособие / Е.В. Ефромеева, Н.М. Ефромеев. – Саратов: Вузовское образование, 2020. – 120 с. – ISBN 978-5-4487-0586-1–URL: <http://www.iprbookshop.ru/86701.html>.

48. Жмудь, В.А. Системы автоматического управления. Новые концепции и структуры регуляторов: учебник / В.А. Жмудь, Л. Димитров, Я. Носек. – Саратов: Ай Пи Эр Медиа, 2019. – 157 с. – ISBN 978-5-4486-0477-5. – URL: <http://www.iprbookshop.ru/80291.html>

49. Земляков, В.Л. Основы автоматического управления: учеб. пособие / В.Л. Земляков, И.К. Цыбрий, И.В. Щербань. – Ростов-на-Дону: Издательство Южного федерального университета, 2017. – 116 с. – ISBN 978-5-9275-2373-3. – URL: <http://www.iprbookshop.ru/87457.html>.

50. Ившин, В.П. Автоматическое регулирование: учеб. пособие / В.П. Ившин, М.Ю. Перухин. – Казань: Казанский национальный исследовательский технологический университет, 2016. – 80 с. – ISBN 978-5-7882-1941-7. – URL: <http://www.iprbookshop.ru/79258.html> (дата обращения: 13.12.2019).

51. Кознов, Д.В. Визуальное моделирование: теория и практика / Д.В. Кознов. – URL: <https://www.intuit.ru/studies/courses/1041/218/info>

52. Мефодьева, Л.Я. Практика КОМПАС. Первые шаги: учеб. пособие / Л.Я. Мефодьева. – Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2014. – 123 с. – ISBN 2227-8397. – URL: <http://www.iprbookshop.ru/45482.html>.

53. Попов, В.П. Теория и анализ систем / В.П. Попов, И.В. Крайнюченко. – Саратов: Ай Пи Эр Медиа, 2018. – 250 с. – ISBN 978-5-4486-0211-5. – URL: <http://www.iprbookshop.ru/70283.html>.

54. Сёмина, В.В. Моделирование систем: методические указания для проведения лабораторных работ по дисциплине «Моделирование систем» / В.В. Сёмина. – Липецк: Липецкий государственный технический университет, ЭБС АСВ, 2016. – 17 с. – ISBN 2227-8397. – URL: <http://www.iprbookshop.ru/64869.html>.

55. Системный анализ в вопросах и ответах: учебное пособие / Е.И. Сметанина. Томск: Томский политехнический университет, 2016. – 108 с. – ISBN 978-5-4387-0678-6. – URL: <http://www.iprbookshop.ru/83984.html>.

56. Хуртасенко, А.В. Компьютерное твердотельное 3D-моделирование: практикум. Учебное пособие / А.В. Хуртасенко, И.В. Маслова. – Белгород: Белгородский государственный технологический университет им. В.Г. Шухова, ЭБС АСВ, 2014. – 128 с. – ISBN 2227-8397. – URL: <http://www.iprbookshop.ru/49710.html>.

57. Эльберг, М.С. Имитационное моделирование: учеб. пособие / М.С. Эльберг, Н.С. Цыганков. – Красноярск: Сибирский федеральный университет, 2017. – 28с. – ISBN 978-5-7638-3648-6. – URL: <http://www.iprbookshop.ru/84350.html>.

58. Образовательный сайт компании «Ascon». – URL: <http://www.eduascon.ru>.

59. Образовательный сайт имитационного моделирования. – URL: <http://www.matlab.ru>.

60. Образовательный сайт «VisSim в России». – URL: <http://www.vissim.com>.

61. Сайт компании «AnyLogic». – URL: <http://www.anylogic.ru>.

62. Сайт компании «MVStadium Group». – URL: <http://www.mvstadium.com>.

Учебное издание

Королев Александр Леонидович
Паршукова Наталья Борисовна

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ
ОБЪЕКТОВ, ПРОЦЕССОВ И СИСТЕМ
Учебное пособие

ISBN 978-5-907409-15-6

Рукопись рекомендована РИС ЮУрГГПУ

Протокол №21, от 2020 г.

Редактор *Л.Н. Корнилова*

Технический редактор *Н.А. Усова*

Издательство ЮУрГГПУ

454080, г. Челябинск, пр. Ленина, 69

Подписано в печать 24.11.2020 г.

Объем 12 уч.-изд. л. (38,4 усл. п. л.)

Формат 60*84/8 Тираж 100 экз.

Заказ

Отпечатано с готового оригинал-макета в типографии ЮУрГГПУ

454080, г. Челябинск, пр. Ленина, 69