

**Т.Н. ЛЕБЕДЕВА  
Л.С. НОСОВА  
А.А. РУЗАКОВ**

**ИНФОРМАЦИОННЫЕ  
СИСТЕМЫ  
И БАЗЫ ЗНАНИЙ**

Учебно-методическое пособие

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Южно-Уральский государственный гуманитарно-  
педагогический университет»

**Т.Н. ЛЕБЕДЕВА**  
**Л.С. НОСОВА**  
**А.А. РУЗАКОВ**

## **ИНФОРМАЦИОННЫЕ СИСТЕМЫ И БАЗЫ ЗНАНИЙ**

Учебно-методическое пособие

Челябинск  
2017

**ББК 32.973**

**УДК 004.43**

**Л 33**

**Лебедева, Т.Н. Информационные системы и базы знаний** [Текст]: учебно-методическое пособие / Т.Н. Лебедева, Л.С. Носова, А.А. Рузаков. – Челябинск: Изд-во Юж.-Урал. гос. гуман.-пед. ун-та, 2017. – 200 с.

ISBN 978-5-906908-60-5

В данном пособии представлены основные положения теории информационных систем и баз знаний от их возникновения до современного представления. Рассматриваются информационные, интеллектуальные и корпоративные системы, даются их определения, классификации, архитектуры и этапы их разработки.

Пособие предназначено для студентов, преподавателей, а также может быть полезно учителям средних образовательных учреждений при проведении учебных занятий.

Рецензенты: И.Ю. Коробейникова, канд. пед. наук, доцент

О.А. Дмитриева, канд. пед. наук, доцент

ISBN 978-5-906908-60-5

© Т.Н. Лебедева, Л.С. Носова, А.А. Рузаков, 2017

© Издательство Южно-Уральского государственного гуманитарно-педагогического университета, 2017

## СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b> .....	7
<b>1. ИНФОРМАЦИОННЫЕ СИСТЕМЫ</b> .....	9
<b>1.1. Понятие информационной системы. Классификация</b> .....	9
<b>1.2. Структура информационной системы</b> .....	15
1.2.1. Информационное обеспечение .....	15
1.2.2. Техническое обеспечение .....	18
1.2.3. Математическое и программное обеспечение ...	18
1.2.4. Организационное обеспечение .....	19
1.2.5. Правовое обеспечение .....	19
<b>1.3. Архитектура информационной системы</b> .....	20
1.3.1. Одиночные информационные системы .....	20
1.3.2. Групповые информационные системы .....	21
1.3.3. Архитектура информационной системы «Файл-сервер» .....	21
1.3.4. Архитектура информационной системы «Клиент-сервер» .....	23
1.3.5. Многоуровневая архитектура информаци- онной системы .....	27
1.3.6. Интернет/интранет архитектура информа- ционной системы .....	28
<b>1.4. Модели данных</b> .....	30
1.4.1. Иерархическая модель данных .....	30
1.4.2. Сетевая модель данных .....	33
1.4.3. Реляционная модель данных .....	34
1.4.4. Объектноориентированная модель данных ...	42

1.4.5. Модель объектов-ролей .....	44
<b>1.5. Последовательность создания информационно-логической модели .....</b>	<b>45</b>
1.5.1. Понятие информационного объекта .....	45
1.5.2. Понятие информационно-логической модели ...	46
1.5.3. Концептуальная модель .....	47
1.5.4. Логическая модель .....	48
1.5.5. Внутренняя (физическая) модель .....	49
1.5.6. Внешняя модель .....	49
1.5.7. Типы связей .....	50
<b>1.6. Обзор Microsoft Office Access .....</b>	<b>53</b>
1.6.1. Создание таблицы .....	55
1.6.2. Схема данных .....	63
1.6.3. Создание запроса .....	65
1.6.4. Создание форм .....	67
1.6.5. Создание отчетов .....	68
1.6.6. Связи с Office .....	69
1.6.7. Макросы .....	69
<b>Контрольные вопросы .....</b>	<b>70</b>
<b>Задания для самостоятельного выполнения .....</b>	<b>73</b>
<b>2. ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ .....</b>	<b>82</b>
<b>2.1. Основные понятия искусственного интеллекта .....</b>	<b>82</b>
<b>2.2. Понятие информации .....</b>	<b>89</b>
<b>2.3. Основные направления в области искусственного интеллекта .....</b>	<b>97</b>
<b>2.4. Модели представления знаний .....</b>	<b>103</b>

2.4.1. Продукционная модель представления знаний .....	104
2.4.2. Семантические сети .....	108
2.4.3. Фреймовая модель представления знаний ..	114
2.4.4. Логическая модель представления знаний	120
2.4.5. Модель, основанная на нечетких знаниях ..	125
<b>2.5. Экспертные системы .....</b>	<b>128</b>
<b>2.6. Концепции разработки правил экспертных систем .....</b>	<b>132</b>
<b>2.7. История создания языка программирования. Пролог .....</b>	<b>141</b>
<b>Контрольные вопросы .....</b>	<b>146</b>
<b>Задания для самостоятельного выполнения .....</b>	<b>147</b>
<b>3. КОРПОРАТИВНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ .....</b>	<b>148</b>
3.1. Определение корпоративной информационной системы .....	148
3.2. Требования к корпоративной информационной системе .....	153
3.3. Архитектура корпоративных информационных систем .....	159
3.4. Классификация корпоративных информационных систем .....	161
3.5. Типы корпоративных информационных систем .....	164
3.5.1. Системы стандарта MRP .....	165
3.5.2. Системы стандарта MRP II .....	167
3.5.3. Системы ERP .....	170

3.5.4. Системы класса CRM .....	172
3.5.5. Системы CSRP .....	174
3.5.6. Системы ERP II .....	175
<b>3.6. Примеры корпоративных информационных систем .....</b>	<b>177</b>
<b>Контрольные вопросы .....</b>	<b>189</b>
<b>Задания для самостоятельного выполнения .....</b>	<b>191</b>
<b>БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....</b>	<b>193</b>

## ВВЕДЕНИЕ

Накопление, распространение и передача знаний от поколений к поколениям во все времена определяли развитие человеческой цивилизации. В последние десятилетия знания, интеллектуальные ресурсы приобрели особую значимость в социальном и экономическом развитии общества. Это связано с действием ряда фундаментальных факторов и, прежде всего, с информационной революцией и возникновением новой экономики.

Информационная революция, вызванная появлением многих поколений компьютеров и программных систем, их проникновением практически во все сферы деятельности, и последовавшая за ней коммуникационная революция привели к небывалому прогрессу в обработке, хранении и передаче информации, упростили и ускорили взаимодействие между объектами и субъектами экономической, социальной и политической жизни разных стран, привели к глобализации рынка.

Новой экономике – экономике, основанной на знаниях, свойственен стремительный рост наукоемкости товаров и услуг, сокращение их жизненного цикла, интеллектуализация используемых технологий, обеспечивающих кратное повышение производительности труда, возникновение крупного сегмента рынка собственно интеллектуальных продуктов и услуг (патенты, лицензии, транзакции, консалтинг), быстрый темп обновления знаний и необходимость их постоянного пополнения.

В связи с этим возник и новый вид управленческой деятельности – управление знаниями, крупным и растущим активом современных компаний.

Учебно-методическое пособие подготовлено в соответствии с требованиями ФГОС ВО по направлениям подготовки 44.03.02 «Психолого-педагогическое образование», 09.03.02 «Информационные системы и технологии», 44.03.05 «Педагогическое образование», и программам курса «Информационные системы», «Представление знаний в информационных системах», «Информационные системы и базы знаний».

Пособие нацелено на оказание помощи студентам в формировании их общекультурных компетенций в соответствии с Федеральным государственным образовательным стандартом высшего образования по указанным направлениям подготовки.

Современные базовые учебники по изучению информационных систем и баз знаний ориентированы в основном на фундаментальное изложение теории и слишком объемны по содержанию, в них приводится недостаточное количество примеров, дающих понять и осмыслить использование основных правил управления объектами и программными приложениями.

Данное обстоятельство усложняет процесс самостоятельного изучения предмета и является причиной подготовки настоящего издания. На простых примерах демонстрируется создание и управление информационными объектами. В конце каждой главы пособия приведены обобщающие контрольные вопросы и упражнения, необходимые для проверки сформированности уровня компетентности.

# 1. ИНФОРМАЦИОННЫЕ СИСТЕМЫ

## 1.1. Понятие информационной системы. Классификация

Информационная система (ИС) – организационная совокупность технических и обеспечивающих средств, технологических процессов и кадров, реализующих функции сбора, обработки, хранения, поиска, выдачи и передачи информации.

Основной целью создания ИС является «удовлетворение информационных потребностей пользователей путем предоставления необходимой им информации на основе хранимых данных» [52, с. 8].

ИС применяется практически во всех сферах человеческой деятельности: в управлении предприятием, учреждением, производством; при организации научных исследований; в библиотечном деле, в обучении, при выполнении конструкторских и проектных работ.

Классификация ИС в зависимости от вида представлена в табл. 1 [21, с. 34].

Таблица 1

### Основные виды ИС

Вид	Описание
Измерительные	Используются для автоматического (с помощью специальных датчиков) сбора информации о состоянии и параметрах интересующего объекта (например, ИС атомной электростанции)

Продолжение таблицы 1

Вид	Описание
Информационно-справочные системы	Электронные словари, электронные энциклопедии, электронные записные книжки и т.д. (например, Яндекс.Словари; Большая энциклопедия Кирилла и Мефодия и др.)
Информационно-поисковые системы	Всемирная паутина (World Wide Web) с соответствующими поисковыми системами (Яндекс, Google и др.) и юридические информационно-поисковые системы, предназначенные, преимущественно, для хранения документов официального характера, а именно, законов, положений, инструктивных писем, изданных законодательными и исполнительными государственными органами (например, Гарант, КонсультантПлюс)
ИС, обеспечивающие автоматизацию документооборота и учета	Используются для организации документооборота на предприятиях (например, «ДЕЛО» компании «Электронные офисные системы»)
Системы автоматизированного проектирования	Реализуют информационную технологию выполнения функций проектирования и представляют собой организационно-техническую систему, предназначенную для автоматизации процесса проектирования, состоящую из персонала и комплекса технических, программных и других средств автоматизации его деятельности. Содержат наряду с другими компонентами большие массивы справочной технической информации (государственные стандарты, санитарные нормы и правила, технические условия и пр.), алгоритмы проведения расчетов определенных параметров и другую информацию (например, «Компас» компании «Аскон»)

Окончание таблицы 1

Вид	Описание
Автоматизированная система научных исследований	Программно-аппаратный комплекс на базе средств вычислительной техники, предназначенный для проведения научных исследований или комплексных испытаний образцов новой техники на основе получения и использования моделей исследуемых объектов, явлений и процессов (например, EPICS Аргоннской национальной лаборатории)
Экспертные системы и системы поддержки принятия решений	В их основе – базы знаний по конкретной предметной области. Эти системы активно используются при планировании и составлении долгосрочных прогнозов в промышленности, для постановки диагноза в медицине и т.д. (например, MYCIN)
Автоматизированные системы управления	Широкий класс информационных систем, к которым относятся системы управления отдельным технологическим процессом, системы управления всем предприятием и системы управления целой отраслью общественного производства
Геоинформационные системы	В них информация об объектах упорядочена в соответствии с пространственным размещением объектов, представленных чаще всего на географических картах (например, ArcGIS)
Обучающие ИС	Всевозможные электронные учебники, компьютерные тесты, обучающие программы, а также тренажеры, имитирующие работу какого-то устройства

Также ИС можно условно разделить на фактографические и документальные (табл. 2) [48, с. 20].

## Основные типы ИС

Тип	Описание
Фактографические	Накапливают и хранят данные в виде множества экземпляров одного или нескольких типов структурных элементов (информационных объектов). Каждый из таких экземпляров структурных элементов или некоторая их совокупность отражает сведения по какому-либо факту, событию и т.д., отделенному от всех прочих сведений и фактов. Основная идея таких систем заключается в том, что все сведения об объектах хранятся в каком-то заранее оговоренном формате (например, дата – в виде ДД.ММ.ГГГГ). Информация, с которой работает фактографическая ИС, имеет четкую структуру, позволяющую компьютеру отличить одни данные от других. Поэтому фактографическая ИС способна давать однозначные ответы на поставленные вопросы (например, «Кто из студентов 3 курса посещает факультативные занятия по такому-то предмету?»)
Документальные	Обслуживают принципиально иной класс задач, которые не предполагают однозначного ответа на поставленный вопрос. Базы данных таких систем образует совокупность неструктурированных текстовых документов (статьи, книги, рефераты и т.д.) и графических объектов, снабженную тем или иным формализованным аппаратом поиска. Цель данных систем – выдать в ответ на запрос пользователя список документов или объектов, в какой-то мере удовлетворяющих сформулированным в запросе условиям (например, выдать список всех статей,

Тип	Описание
	в которых встречается словосочетание «информационные системы»). Принципиальная особенность документальной ИС – способность, с одной стороны, выдавать ненужные пользователю документы (например, где данное слово употребляется в ином смысле), а с другой стороны – не выдавать нужные (например, если автор употребил какой-то синоним или ошибся в написании)

Фактографические ИС часто работают с неструктурированными блоками информации (текстом, графикой, звуком, видео), снабженными структурированными описателями.

Современной формой ИС являются банки данных, которые включают в свой состав вычислительную систему, одну или несколько баз данных, систему управления базами данных и набор прикладных программ. Основными функциями банков данных являются:

- хранение данных и их защита;
- изменение (обновление, добавление и удаление) хранимых данных;
- поиск и отбор данных по запросам пользователей;
- обработка данных и вывод результатов.

База данных (БД) – именованная совокупность данных, отражающая состояние объектов и их отношений в конкретной предметной области.

«Система управления базами данных (СУБД) – совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями» [52, с. 7].

Прикладные программы (приложения) в составе банков данных служат для обработки данных, вычислений и формирования выходных документов по заданной форме.

Процесс создания ИС обычно включает следующие этапы:

- проектирование БД;
- создание БД (формирование и связывание таблиц, первичный ввод данных);
- создание меню, форм, запросов и отчетов;
- создание приложения.

Приведенный перечень этапов не является строгим в смысле очередности и обязательности. Процесс создания ИС, как правило, имеет итерационный характер.

Приложение представляет собой программу или комплекс программ, использующих БД и обеспечивающих автоматизацию обработки информации из некоторой предметной области. Приложения могут создаваться как в среде СУБД, так и вне СУБД – с помощью системы программирования, использующей средства доступа к БД.

Для работы с БД во многих случаях можно обойтись только средствами СУБД, создавая запросы и отчеты. Приложения разрабатывают главным образом в тех случаях, когда требуется обеспечить удобство работы с БД неквалифицированным пользователям или интерфейс СУБД не устраивает пользователя.

## 1.2. Структура информационной системы

Структуру ИС составляет совокупность отдельных ее частей, называемых подсистемами (часть системы, выделенная по какому-либо признаку).

При рассмотрении структуры ИС как совокупности подсистем «подсистемы называют обеспечивающими» [32, с. 133].

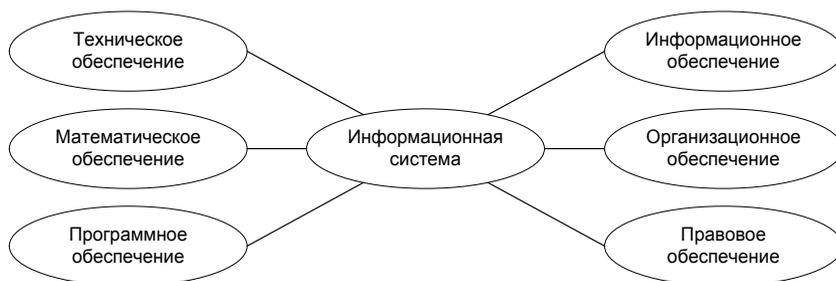


Рис. 1. Структура ИС

### 1.2.1. Информационное обеспечение

Назначение подсистемы информационного обеспечения состоит в своевременном формировании и выдаче достоверной информации для принятия управленческих решений.

Информационное обеспечение – совокупность единой системы классификации и кодирования информации, унифицированных систем документации, схем информационных потоков, циркулирующих в организации, а также методология построения БД.

Унифицированные системы документации создаются на различных уровнях. Главная цель – это обеспечение сопоставимости показателей различных сфер общественного производства.

Разработаны стандарты, где устанавливаются требования:

- к унифицированным системам документации;
- к унифицированным формам документов различных уровней управления;
- к составу и структуре атрибутов и показателей;
- к порядку внедрения, ведения и регистрации унифицированных форм документов.

Но при обследовании большинства организаций постоянно выявляется целый комплекс типичных недостатков:

- чрезвычайно большой объем документов для ручной обработки;
- одни и те же показатели часто дублируются в разных документах;
- работа с большим количеством документов отвлекает специалистов от решения непосредственных задач;
- имеются показатели, которые создаются, но не используются, и др.

Поэтому устранение указанных недостатков является одной из задач, стоящих при создании информационного обеспечения.

Схемы информационных потоков отражают маршруты движения информации и ее объемы, места возникновения первичной информации и использования результатной информации.

Методология построения БД базируется на теоретических основах их проектирования.

Основные её идеи в виде двух последовательно реализуемых на практике этапов:

1-й этап – обследование всех функциональных подразделений организации с целью:

- понять специфику и структуру её деятельности;
- построить схему информационных потоков;
- проанализировать существующую систему документооборота;
- определить информационные объекты и соответствующий состав атрибутов (параметров, характеристик), описывающих их свойства и назначение.

2-й этап – построение информационно-логической модели данных для обследованной на 1-м этапе сферы деятельности. В этой модели должны быть установлены и оптимизированы все связи между объектами и их атрибутами. Информационно-логическая модель является фундаментом, на котором будет создана БД.

Для создания информационного обеспечения необходимо:

- ясное понимание целей, задач, функций всей системы управления организацией;
- выявление движения информации от момента возникновения и до ее использования на различных уровнях управления, представленной для анализа в виде схем информационных потоков;
- совершенствование системы документооборота;
- наличие и использование системы классификации и кодирования;
- владение методологией создания концептуальных информационно-логических моделей, отражающих взаимосвязь информации;

- создание массивов информации на машинных носителях, что требует наличия современного технического обеспечения.

### *1.2.2. Техническое обеспечение*

Техническое обеспечение – комплекс технических средств, предназначенных для работы ИС, а также соответствующая документация на эти средства и технологические процессы.

Комплекс технических средств составляют:

- компьютеры;
- устройства сбора, накопления, обработки, передачи и вывода информации;
- устройства передачи данных и линий связи;
- оргтехника и устройства автоматического съема информации;
- эксплуатационные материалы и др.

Документацией оформляются предварительный выбор технических средств, организация их эксплуатации, технологический процесс обработки данных, технологическое оснащение.

### *1.2.3. Математическое и программное обеспечение*

Математическое и программное обеспечение – совокупность математических методов, моделей, алгоритмов и программ для реализации целей и задач ИС, а также нормального функционирования комплекса технических средств.

К средствам математического обеспечения относятся:

- средства моделирования процессов управления;
- типовые задачи управления;
- методы математического программирования, математической статистики, теории массового обслуживания и др.

#### *1.2.4. Организационное обеспечение*

Организационное обеспечение – совокупность методов и средств, регламентирующих взаимодействие работников с техническими средствами и между собой в процессе разработки и эксплуатации ИС.

Организационное обеспечение реализует следующие функции:

- анализ существующей системы управления организацией, где будет использоваться ИС, и выявление задач, подлежащих автоматизации;
- подготовку задач к решению на компьютере, включая техническое задание на проектирование ИС и технико-экономическое обоснование ее эффективности;
- разработку управленческих решений по составу и структуре организации, методологии решения задач, направленных на повышение эффективности системы управления.

Организационное обеспечение создается по результатам предпроектного обследования на 1-м этапе построения БД.

#### *1.2.5. Правовое обеспечение*

Правовое обеспечение – совокупность правовых норм, определяющих создание, юридический статус и функционирование информационных систем, регламентирующих порядок получения, преобразования и использования информации.

Главной целью правового обеспечения является укрепление законности.

### 1.3. Архитектура информационной системы

Эффективность функционирования ИС во многом зависит от ее архитектуры. Под архитектурой ИС понимается ее построение.

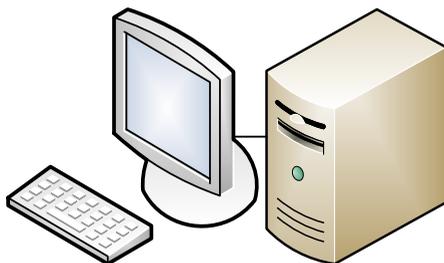
ИС классифицируются по различным признакам.

Классификация ИС по масштабу:

- одиночные;
- групповые;
- корпоративные (рассмотрены в разделе 3 настоящего пособия) [19, с. 28].

#### 1.3.1. Одиночные информационные системы

Реализуются, как правило, на автономном персональном компьютере (ПК) в виде автоматизированного рабочего места (см. рис. 2).



Автоматизированное рабочее место  
(прикладная часть,  
управление БД,  
хранение данных)

Рис. 2. Одиночная ИС

ИС может содержать несколько простых приложений, работающих с общей БД, и быть рассчитанной на работу одного пользователя или группы пользователей, разделяющих по времени одно рабочее место. Для создания БД используются настольные (локальные) СУБД, например, Microsoft Office Access 2016.

### *1.3.2. Групповые информационные системы*

Групповые ИС ориентированы на совместное использование информации из БД.

По способу организации групповые и корпоративные ИС можно классифицировать на:

- файл-серверные;
- клиент-серверные;
- многоуровневые (многозвенные);
- интернет/интранет.

### *1.3.3. Архитектура информационной системы «Файл-сервер»*

Архитектура ИС «Файл-сервер» схожа по своей структуре с архитектурой одиночной ИС и использует сетевой ресурс (файловый сервер) для хранения приложений и БД (см. рис. 3) [7, с. 31]. В подобных системах СУБД позволяли создавать приложения, одновременно работающие с одной БД на файловом сервере. Основная проблема подобной технологии работы заключается в явной или неявной обработке транзакций, а также неизбежно возникающей при коллективном доступе проблеме обеспечения смысловой и ссылочной целостности БД при одновременном изменении одних и тех же данных.

Отметим общие недостатки «файл-серверного» подхода для реализации ИС:

- вся тяжесть вычислительной нагрузки при доступе к БД ложится на приложение клиента, т.к. обработка запросов выполняется на стороне клиента, а для этого все необходимые таблицы БД должны скопироваться на клиентский компьютер;

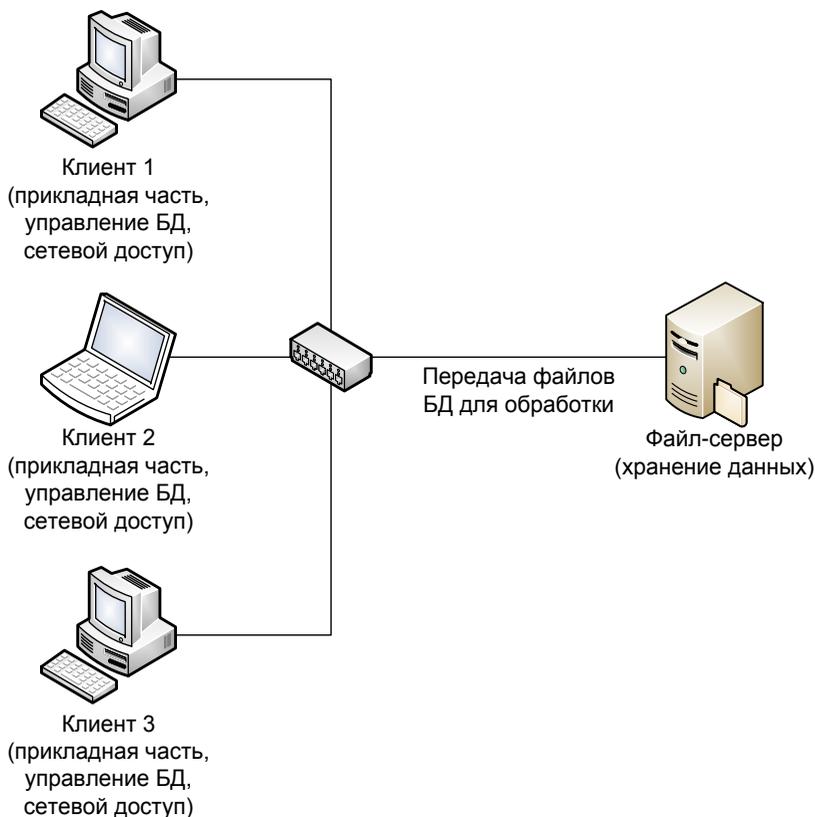


Рис. 3 Файл-серверная архитектура ИС

- не оптимально расходуются ресурсы клиентского компьютера и сети, например, если результатом запроса

являются 2 записи из таблицы объемом 10 000 записей, то вначале все 10 000 записей будут скопированы с файл-сервера на клиентский компьютер, и только затем выполнится отбор; в результате возрастает сетевой трафик и увеличиваются требования к аппаратным мощностям клиентского компьютера;

- необходимость постоянного увеличения вычислительных мощностей клиентского компьютера, обусловленная не столько развитием программного обеспечения как такового, сколько возрастанием обрабатываемых объемов информации;

- возможность проектирования и использования взаимоисключающих бизнес-правил, т.к. они реализуются в приложениях;

- недостаточно развитый аппарат транзакций для локальных СУБД служит потенциальным источником ошибок как с точки зрения одновременного внесения изменений в одну и ту же запись, так и с точки зрения отката транзакции.

Основным достоинством данной архитектуры является простота организации. Проектировщики и разработчики ИС находятся в привычных и комфортных условиях в среде Windows 10 (Windows Server 2016).

#### *1.3.4. Архитектура информационной системы «Клиент-сервер»*

Приведенные недостатки архитектуры «Файл-сервер» решаются при переводе ИС в архитектуру «Клиент-сервер», являющуюся следующим этапом в развитии ИС. В данной архитектуре вычислительная нагрузка перенесена на сервер БД. Это способствует максимальной разгрузке приложения клиента от вычислительной работы, а также

существенному укреплению безопасности данных – как от злонамеренных, так и просто ошибочных изменений.

БД хранится на сетевом сервере, как и в архитектуре «Файл-сервер», но прямого доступа к БД из приложений клиентов не происходит. Функции прямого доступа к БД осуществляет специальная управляющая программа – сервер БД, поставляемая разработчиком СУБД (см. рис. 4) [7, с. 34; 19, с. 32].

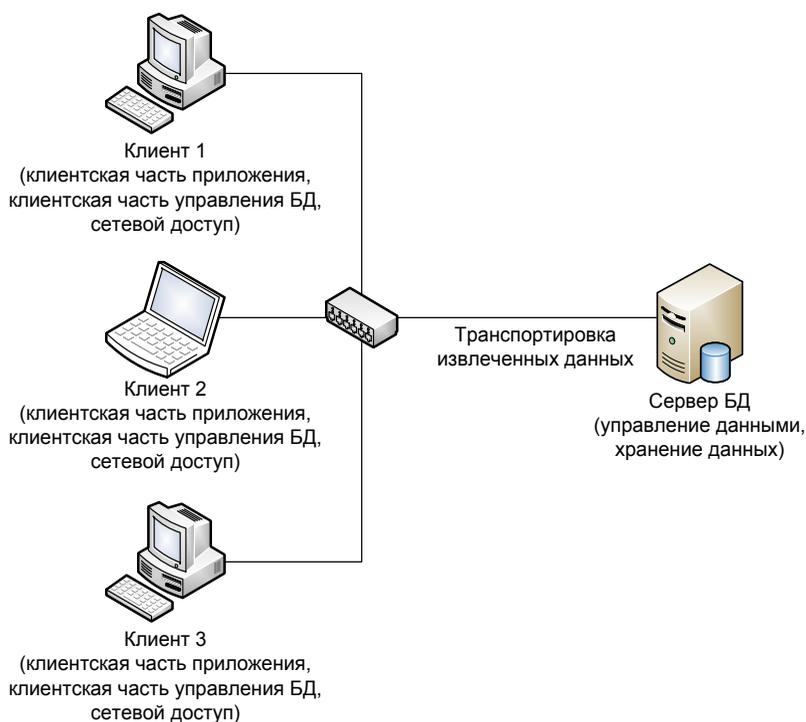


Рис. 4. Клиент-серверная архитектура ИС

Типовое взаимодействие сервера БД и приложения-клиента происходит следующим образом:

1. Клиент формирует запрос на языке структурированных запросов SQL (англ. Structured Query Language).
2. Клиент отсылает запрос серверу БД.
3. Сервер БД принимает запрос.
4. Сервер БД производит компиляцию полученного запроса.
5. Сервер БД выполняет откомпилированный запрос.
6. Сервер БД отправляет результаты выполнения запроса клиенту.

В клиентском приложении в основном осуществляется интерпретация полученных от сервера данных, реализация интерфейса с пользователем и ввод данных, а также реализация части бизнес-правил.

Преимущества архитектуры ИС «Клиент-сервер»:

- большинство вычислительных процессов происходит на сервере, за счет этого снижаются требования к вычислительным мощностям компьютеров клиентов;
- снижается сетевой трафик за счет посылки сервером клиенту только тех данных, которые он запрашивал; например, если результатом запроса являются 2 записи из таблицы объемом 10 000 записей, то сервер выполнит запрос и перешлет клиенту набор данных именно из 2 записей;
- в условиях развития программного обеспечения и возрастания обрабатываемых объемов информации проще и чаще дешевле усилить вычислительные мощности на сервере или полностью его заменить на более мощный, нежели наращивать мощности или полностью заменять клиентские компьютеры;

- на сервере БД, как правило, хранится в виде единого файла, доступ к которому контролируется серверной частью СУБД; взломать такую БД, даже при наличии умысла, тяжело; кроме этого, значительно увеличивается защищенность БД от ввода неправильных значений, т.к. сервер БД проводит автоматическую проверку соответствия вводимых значений наложенным ограничениям и автоматически выполняет необходимые бизнес-правила, а также отслеживает уровни доступа для каждого пользователя и блокирует осуществление попыток выполнения неразрешенных для пользователя действий;

- сервер реализует управление транзакциями и предотвращает попытки одновременного изменения одних и тех же данных.

Недостатки архитектуры «Клиент-сервер»:

- необходимость администрирования сервера БД;
- высокая стоимость оборудования;
- бизнес-логика приложений осталась в клиентской части приложения;
- в случае большого числа пользователей возникают проблемы своевременной и синхронной замены версий клиентских приложений на рабочих станциях.

Для реализации архитектуры ИС «Клиент-сервер» применяют так называемые «промышленные» СУБД, такие как Microsoft SQL Server 2016, Oracle Database 12c, MySQL и т.д.

Данный вид архитектуры также называют архитектурой с «толстым» клиентом.

### *1.3.5. Многоуровневая архитектура информационной системы*

Часть недостатков архитектуры ИС «Клиент-сервер» решается в рамках многоуровневой (многозвенной) архитектуры. Данная архитектура позволяет максимально снизить требования к клиентским компьютерам. По этой причине значительная часть клиентского приложения переносится в промежуточное звено – сервер приложений, который может быть расположен отдельно (для крупных ИС их может быть несколько) или на сервере БД. В функции сервера приложений входит непосредственное взаимодействие клиентов с сервером БД, реализуя посредничество между ним и остальными клиентскими компьютерами. В результате получается «облегченный» клиент, который не требует больших ресурсов памяти и может загружаться с сетевого компьютера, – это главное преимущество многоуровневой архитектуры (см. рис. 5) [7, с. 37; 21, с. 34].

К дополнительным достоинствам можно отнести:

- размещение большей части бизнес-правил на сервере приложений позволяет гибко их изменять без необходимости обновления на многочисленных клиентских приложениях;
- уменьшение сетевого трафика за счет отложенного обновления и регулируемого объема пакетов данных;
- простоту распространения новых версий клиентского программного обеспечения, т.к. больше нет необходимости устанавливать на клиентских компьютерах механизмы доступа к данным.

Данный вид архитектуры также называют архитектурой с «тонким» клиентом.

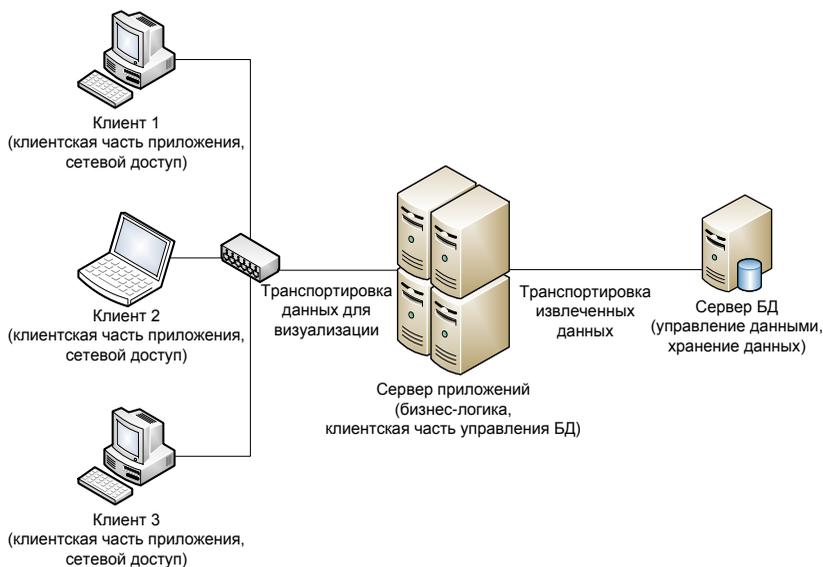


Рис. 5. Многоуровневая архитектура ИС

### 1.3.6. Интернет/интранет архитектура информационной системы

Если в качестве сервера приложений используется веб-сервер, то мы переходим к интернет/интранет архитектуре ИС (см. рис. 6). В данном случае работа клиентов осуществляется посредством веб-браузера. Выполняемые клиентом действия на веб-странице (ввод запроса на поиск, ввод данных в форме и т.д.) веб-браузером отправляются на веб-сервер. Веб-сервер выполняет обработку поступивших запросов, обращается к серверу БД и формирует результирующие веб-страницы, отправляемые затем клиентам [7, с. 37; 21, с. 35].

Для создания современных веб-приложений применяются веб-сервисы (веб-службы), не зависящие от платформы,

объектной модели и клиента программные компоненты, которые можно вызывать из клиентских веб-приложений.

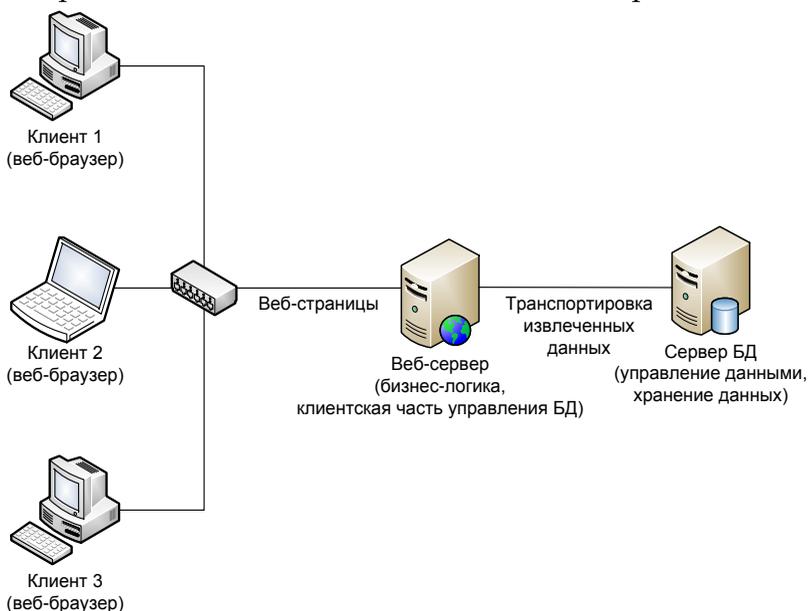


Рис. 6. Интернет/интранет архитектура ИС

Архитектура ИС, использующая сервисы (службы), получила название *сервис-ориентированная архитектура SOA* (англ. Service-Oriented Architecture).

Применяя SOA, организации могут создавать гибкие корпоративные ИС, позволяющие оперативно разрабатывать новые бизнес-процессы и многократно использовать одни и те же программные компоненты.

SOA подразумевает предоставление программного обеспечения как сервиса SaaS (англ. Software as a Service). В качестве сервисов могут также выступать инфраструктура

IaaS (англ. Infrastructure as a Service) или аппаратная платформа PaaS (англ. Platform as a Service). Все это объединяется общим понятием – *облачные вычисления* (англ. cloud computing).

#### **1.4. Модели данных**

Независимость данных от приложений является важнейшим достоинством применения БД в ИС. Благодаря этому приложения не обременяются проблемами представления данных на физическом уровне: размещения данных в памяти, методов доступа к ним и т.д. Независимость достигается поддерживаемым СУБД многоуровневым представлением данных в БД на логическом (пользовательском) и физическом уровнях.

В основе любой БД лежит модель данных (МД). МД представляет собой множество структур данных, ограниченной целостности и операций манипулирования данными. Объекты предметной области и взаимосвязи между ними представляются с помощью МД.

МД – совокупность структур данных и операций их обработки.

БД и СУБД строятся на использовании иерархической, сетевой или реляционной МД, на комбинации этих МД или на некотором их подмножестве [15, с. 182]. С конца 1980-х гг. начала использоваться объектная (объектно-ориентированная) МД. Также в начале 1970-х гг. была предложена модель объектов-ролей [21, с. 77].

### 1.4.1. Иерархическая модель данных

Иерархическая МД представляет совокупность элементов, связанных между собой по определенным правилам. Объекты связываются иерархическими отношениями, образующими ориентированный граф (см. рис. 7) [15, с. 182; 37, с. 27].

Основные понятия иерархической МД: уровень, элемент (узел), связь. Узел (вершина графа) – это совокупность атрибутов данных, описывающих некоторый объект. Каждый узел на более низком уровне связан только с одним узлом, находящимся на более высоком уровне. Иерархическое дерево имеет только одну вершину (корень дерева), не подчиненную никакой другой вершине и находящуюся на самом верхнем (первом) уровне. Зависимые (подчиненные) узлы находятся на втором, третьем и т.д. уровнях. Количество деревьев в базе данных определяется числом корневых записей.

К каждой записи базы данных существует только один (иерархический) путь от корневой записи. Например, как видно из рис. 7, для записи С5 путь проходит через записи А и В3.

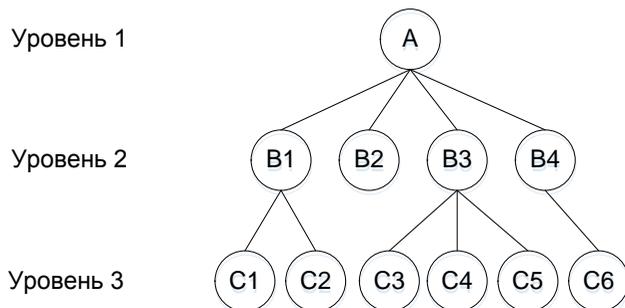


Рис. 7. Иерархическая МД

На рис. 8 приведен пример иерархической МД для хранения информации о студентах факультета, данная структура правомерна, т.к. каждый студент учится в определенной (только одной) группе, которая относится к определенному (только одному) факультету.



Рис. 8. Пример иерархической МД

Иерархическая МД является исторически первой, на ее основе в конце 1960-х – начале 1970-х гг. были разработаны первые профессиональные СУБД – IMS (англ. Information Management System) фирмы IBM, отечественные «ОКА» и «ИНЭС».

В иерархической модели устанавливается строгий порядок обхода дерева (сверху вниз, слева направо) и следующие операции над данными:

- найти указанное дерево;
- перейти от одного дерева к другому;
- перейти от одной записи к другой;
- перейти от одной записи к другой в порядке обхода иерархии;

- удалить текущую запись.

В иерархической МД автоматически поддерживается целостность ссылок между предками и потомками. Основное правило: никакой потомок не может существовать без своего родителя.

#### 1.4.2. Сетевая модель данных

Сетевая МД является расширением иерархической и широко применялась в 1970-е гг. в первых СУБД, использовавшихся крупными корпорациями для создания ИС [15, с. 184].

В сетевой МД при тех же основных понятиях (уровень, узел, связь) каждый элемент может быть связан с любым другим элементом (см. рис. 9).

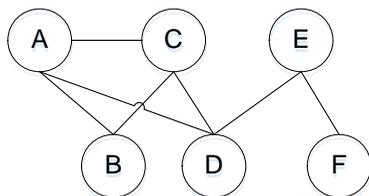


Рис. 9. Сетевая МД

Примером сложной сетевой структуры может служить структура БД, содержащей сведения о студентах, посещающих факультативные курсы. Один студент может посещаться несколько факультативных курсов, а также возможно посещение несколькими студентами одного факультативного курса (см. рис. 10). Единственное отношение представляет собой сложную связь между записями в обоих направлениях.



Рис. 10. Пример сетевой МД

Примеры сетевых СУБД: IDS (англ. Integrated Data Store) компании General Electric, IDMS (англ. Integrated Database Management System) компании Cullinet Software Inc., отечественные «СЕТЬ», «БАНК», «СЕТОР». Эталонный вариант сетевой МД, разработанный с участием Чарльза Бахмана, был описан в проекте «Рабочей группы по базам данных» КОДАСИЛ (англ. DBTG CODASYL).

В сетевой МД устанавливаются следующие операции над данными:

- найти конкретную запись (экземпляр) в наборе однотипных записей;
- перейти от предка к первому потомку по некоторой связи;
- создать новую запись;
- уничтожить запись;
- модифицировать запись;
- включить в связь;
- исключить из связи;
- переставить в другую связь.

#### 1.4.3. Реляционная модель данных

Понятие реляционной МД (англ. relation – отношение) связано с разработками известного американского специалиста в области систем БД Эдгара Франка Кодда [15, с. 186].

Реляционная МД характеризуется простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных.

Реляционная МД некоторой предметной области представляет собой набор отношений, изменяющихся во времени. При создании ИС совокупность отношений позволяет хранить данные об объектах предметной области и моделировать связи между ними [52].

Элементы реляционной МД и формы их представления приведены в табл. 3.

Таблица 3

### Элементы реляционной МД

Элемент	Форма представления
Отношение	Таблица
Схема отношения	Строка заголовков таблицы
Кортеж	Строка таблицы
Сущность	Набор свойств объекта
Атрибут	Заголовок столбца таблицы
Первичный ключ	Один или несколько атрибутов
Тип данных	Тип значений элементов таблицы

Отношение представляет собой двумерную таблицу, содержащую некоторые данные.

Сущность – это объект любой природы, данные о котором хранятся в БД. Данные о сущности хранятся в отношении.

Атрибуты представляют собой свойства, характеризующие сущность.

Математически отношение можно описать следующим образом. Пусть даны  $N$  множеств  $D_1, D_2, \dots, D_N$ , тогда  $R$  есть отношение над этими множествами, если  $R$  есть множество упорядоченных  $n$ -кортежей вида  $\langle d_1, d_2, \dots, d_n \rangle$ , где  $d_1$  – элемент из  $D_1$ ,  $d_2$  – элемент из  $D_2$ , ... и  $d_n$  – элемент из  $D_N$ .  $D_1, D_2, \dots, D_N$  называются доменами отношения  $R$  [53, с. 126].

На рис. 11 приведен пример представления отношения *Преподаватель*.

Отношение *Преподаватель* (таблица)      Атрибут Стаж (заголовок столбца)      Схема отношения (строка заголовка)

Код преподавателя	Фамилия	Имя	Отчество	Код должности	Стаж
1	Иванов	Иван	Михайлович	1	5
2	Петров	Михаил	Иванович	2	7
3	Сидоров	Николай	Григорьевич	1	10

Кортеж (строка)      Значение атрибута (значение поля в записи)

Рис. 11. Представление отношения *Преподаватель*

Множество всех допустимых значений каждого атрибута отношения образует домен. Отношение *Преподаватель* включает 6 доменов:

- 1 содержит коды всех преподавателей;
- 2 – фамилии всех преподавателей;
- 3 – имена всех преподавателей;
- 4 – отчества всех преподавателей;
- 5 – коды всех должностей;

- 6 – стаж работы преподавателей.

Каждый домен образует значения одного типа, например, числовые или символьные данные.

Отношение *Преподаватель* содержит 3 кортежа. Кортеж рассматриваемого отношения состоит из 6-и элементов, каждый из которых выбирается из соответствующего домена. Каждому кортежу соответствует строка таблицы.

Схема отношения представляет собой список имен атрибутов. В нашем случае схема отношения имеет вид – *Преподаватель* (Код преподавателя, Фамилия, Имя, Отчество, Код должности, Стаж).

Число столбцов в отношении называется степенью (кардинальным числом). Текущее число кортежей в отношении называется мощностью. Степень отношения обычно не изменяется после создания отношения, но мощность будет колебаться по мере добавления новых и удаления старых кортежей.

Ключом отношения (первичным ключом) называется атрибут отношения (набор атрибутов), однозначно идентифицирующий каждый из его кортежей [19, с. 132]. В отношении *Преподаватель* (Код преподавателя, Фамилия, Имя, Отчество, Код должности, Стаж) ключевым является атрибут *Код преподавателя*. В дальнейшем первичный ключ в отношении будем подчеркивать – Код преподавателя.

Первичный ключ должен обладать следующими свойствами:

- иметь уникальные значения;
- не содержать пустых (неопределенных) значений;

- быть компактным, т.е. содержать только такие атрибуты, удаление любого из которых может привести к утрате уникальности.

В качестве первичного ключа может использоваться:

- естественный ключ – набор атрибутов сущности, уникально её идентифицирующий (например, при отсутствии в отношении Преподаватель атрибута Код преподавателя, роль первичного ключа мог бы сыграть набор естественных атрибутов: Фамилия, Имя, Отчество при условии отсутствия полных тезок в разрабатываемой ИС);

- суррогатный ключ – автоматически сгенерированное поле, никак не связанное с информационным содержанием записи. Обычно в роли суррогатного ключа выступает автоинкрементное поле целого типа.

Существует два мнения:

- Суррогатные ключи должны использоваться, только если естественных ключей не существует. Если же естественный ключ существует, то идентификация записи внутри БД осуществляется по имеющемуся естественному ключу.

- Суррогатные ключи должны добавляться в любую таблицу, на которую существуют ссылки из других таблиц, и связи между таблицами должны организовываться только при помощи суррогатных ключей. Поиск записи и представление ее пользователю по-прежнему производятся на основании естественного ключа.

Достоинства суррогатных ключей:

- упрощение сопровождения. Поскольку операции связи между таблицами отделены от логики «внутри таблиц», и то и другое можно менять независимо и не затрагивая остального;

- уменьшение размера БД;
- увеличение скорости выборки данных;
- увеличение скорости обновления данных.

При организации связи между двумя отношениями одно из них будет являться родительским (главным), а другое – дочерним (подчиненным). Таким образом, в родительском отношении описана родительская сущность, в дочернем – дочерняя.

Внешний ключ существует только для дочерней сущности и является ссылкой на значение ключа родительской сущности. При создании связей (отношений) между сущностями в дочернюю сущность передаются атрибуты, составляющие первичный ключ родительской сущности. Эти атрибуты и составляют внешний ключ.

С помощью внешних ключей устанавливаются связи между отношениями. Например, имеются два отношения *Преподаватель* (Код преподавателя, Фамилия, Имя, Отчество, Код должности, Стаж) и *Предмет* (Код предмета, Название предмета), которые связываются отношением *Преподает* (Код преподавателя, Код предмета) (см. рис. 12). В связующем отношении атрибуты *Код преподавателя* и *Код предмета* образуют составной первичный ключ. Эти же атрибуты представляют собой внешние ключи, являющиеся первичными ключами других отношений.

*Преподаватель*



Рис. 12. Связь отношений

Реляционная МД накладывает на внешние ключи ограничение для обеспечения целостности данных, называемое ссылочной целостностью. Это означает, что каждому значению внешнего ключа должны соответствовать строки в связываемых отношениях.

Условия и ограничения, накладываемые на отношения, выполнение которых позволяет таблицу считать отношением:

1. Все строки таблицы должны быть уникальны, т.е. не может быть строк с одинаковыми первичными ключами.
2. В таблице не должно быть столбцов с повторяющимися именами.
3. Все строки одной таблицы должны иметь одну структуру, соответствующую именам и типам столбцов.
4. Имена столбцов таблицы должны быть различными, а значения их простыми, т.е. не допустима группа значений в одном столбце одной строки.
5. Порядок размещения строк в таблице может быть

произвольным.

К отношениям можно применять систему операций, позволяющую получать одни отношения из других. Например, результатом запроса к реляционной БД может быть новое отношение, вычисленное на основе хранящихся в базе отношений. Отсюда появляется возможность разделить обрабатываемые данные на хранимую и вычисляемую части.

Основной единицей обработки данных в реляционных БД является отношение, а не отдельные его кортежи (записи), как это принято в традиционных языках программирования.

Операции, выполняемые над отношениями, можно разделить на две группы.

Первую группу составляют операции над множествами: объединение, пересечение, разность, деление и декартово произведение.

Вторую группу составляют специальные операции над отношениями: проекция, соединение, выбор.

В реляционных СУБД для выполнения операций над отношениями используют две группы языков, имеющие в качестве своей математической основы реляционную алгебру и реляционное исчисление соответственно.

В реляционной алгебре операнды и результаты всех действий являются отношениями. Языки реляционной алгебры являются процедурными, так как отношение, являющееся результатом запроса к реляционной БД, вычисляется при выполнении последовательности операций над хранимыми в ней отношениями. В основном языки СУБД являются процедурными.

Языки реляционного исчисления являются непроедурными. Запрос к БД, выполненный с использованием подобного языка, содержит лишь информацию о желаемом результате. Для этих языков характерно наличие наборов правил для записи запросов. В частности, к языкам этой группы относится SQL.

Между реляционной алгеброй и реляционным исчислением существует связь с помощью так называемой процедуры редукции, которая сводит любое выражение реляционного исчисления к набору стандартных операций реляционной алгебры и наоборот.

Сейчас основным предметом критики реляционной МД является некоторая ограниченность при использовании в нетрадиционных областях (системы автоматизации проектирования), в которых требуются предельно сложные структуры данных.

Выделяют еще один недостаток: невозможность адекватного отражения семантики предметной области (возможности представления знаний о семантической специфике предметной области в реляционных системах очень ограничены).

Их устранение привело к созданию объектно-ориентированных БД.

#### *1.4.4. Объектно-ориентированная модель данных*

Объектно-ориентированная МД представляет собой объединение двух МД, реляционной и сетевой, и используется для создания крупных БД со сложными структурами данных.

Направление объектно-ориентированных баз данных (ООБД) возникло сравнительно давно. Публикации появлялись уже в середине 1980-х г. Однако наиболее активно это направление развивается в последние годы. С каждым годом увеличивается число публикаций и реализованных коммерческих и экспериментальных систем [24].

Среди языков и систем программирования наибольшее первичное влияние на ООБД оказал SmallTalk.

Большое число опубликованных работ не означает, что все проблемы ООБД полностью решены. Как отмечается в Манифесте группы ведущих ученых, занимающихся ООБД, современная ситуация с ООБД напоминает ситуацию с реляционными системами середины 1970-х г. При наличии большого количества экспериментальных проектов (и даже коммерческих систем) отсутствует общепринятая объектно-ориентированная МД, и не потому, что нет ни одной разработанной полной модели, а по причине отсутствия общего согласия о принятии какой-либо модели. На самом деле имеются и более конкретные проблемы, связанные с разработкой декларативных языков запросов, выполнением и оптимизацией запросов, формулированием и поддержанием ограничений целостности, синхронизацией доступа и управлением транзакциями и т.д.

В настоящее время ведется очень много экспериментальных и производственных работ в области объектно-ориентированных СУБД. Больше всего университетских работ, которые в основном носят исследовательский характер.

Основными понятиями, с которыми оперирует эта МД, являются следующие:

- объекты, обладающие внутренней структурой и однозначно идентифицируемые уникальным внутрисистемным ключом;

- классы, являющиеся, по сути, типами объектов;

- операции над объектами одного или разных типов, называемые методами;

- инкапсуляция структурного и функционального описания объектов, позволяющая разделять внутреннее и внешнее описания (в терминологии предшествовавшего объектно-модульного программирования – модульность объектов);

- наследуемость внешних свойств объектов на основе соотношения «класс-подкласс».

К достоинствам объектно-ориентированной МД относят:

- возможность для пользователя системы определять свои сколь угодно сложные типы данных (используя имеющийся синтаксис и свойства наследуемости и инкапсуляции);

- наличие наследуемости свойств объектов;

- повторное использование программного описания типов объектов при обращении к другим типам, на них ссылающимся.

Несколько лет назад отмечалось существование, по меньшей мере, тринадцати коммерчески доступных систем ООБД. Среди них – O2, ORION, ONTOS, UniSQL, GemStone и Iris.

#### *1.4.5. Модель объектов-ролей*

В отличие от реляционной МД в ней нет атрибутов, а основные понятия – это объекты и роли, описывающие

их. Роли могут быть как «изолированные», присущие исключительно какому-нибудь объекту, так и существующие как элемент какого-либо отношения между объектами. Модель служит для понятийного моделирования, что отличает ее от реляционной МД. Имеются и другие отличия, и интересные особенности: например, для нее помимо графического языка разработано подмножество естественного языка, не допускающее неоднозначностей, и, таким образом, пользователь (заказчик) не только общается с аналитиком на естественном языке, но и видит представленный на том же языке результат его работы по формализации задачи. Многие пользователи, в отличие от аналитиков, с трудом разбираются в описывающих их деятельность рисунках и схемах. Модель «объектов-ролей» сейчас привлекает большое внимание специалистов, однако до промышленных масштабов ее использования, сравнимых с реляционной и объектно-ориентированной, ей пока далеко [21, с. 77].

## **1.5. Последовательность создания информационно-логической модели**

### *1.5.1. Понятие информационного объекта*

Информационный объект (ИО) – это описание некоторой сущности (реального объекта, явления, процесса, события) в виде совокупности логически связанных атрибутов (информационных элементов).

ИО определенного атрибутивного состава и структуры образует класс (тип), которому присваивается уникальное имя (например, *Факультет, Группа, Студент*).

ИО имеет множество реализации – экземпляров, каждый из которых представлен совокупностью конкретных значений атрибутов и идентифицируется значением ключа. Остальные атрибуты ИО являются описательными. При этом одни и те же атрибуты в одних ИО могут быть ключевыми, а в других – описательным [48, с. 40].

Приведем пример описания структуры и экземпляров ИО *Студент*:

- ключевой атрибут – Номер студента;
- описательные атрибуты – Фамилия, Имя, Отчество.

#### 1.5.2. Понятие информационно-логической модели

Проектирование БД состоит в построении комплекса взаимосвязанных МД [48, с. 479]. На рис. 13 условно отображены этапы процесса проектирования БД.



Рис. 13. Этапы процесса проектирования БД

Важнейшим этапом проектирования БД является разработка информационно-логической (инфологической) модели (ИЛМ) предметной области, не ориентированной на СУБД. В ИЛМ средствами структур данных в интегрированном виде отражают состав и структуру данных, а также информационные потребности приложений (задач и запросов).

ИЛМ предметной области отражает предметную область в виде совокупности информационных объектов и их структурных связей.

Данная модель строится первой. Предварительная ИЛМ строится еще на предпроектной стадии и затем уточняется на более поздних стадиях проектирования БД. На ее основе строятся концептуальная, логическая, внутренняя (физическая) и внешняя модели.

На рис. 14 показана многоуровневая архитектура представления данных [16, с. 75; 20, с. 478].

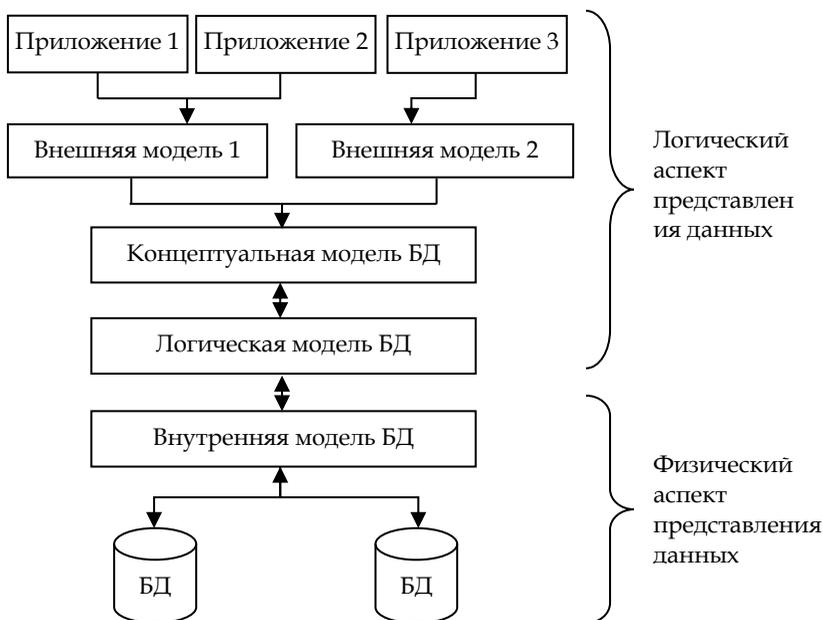


Рис. 14. Многоуровневая архитектура представления данных.

Различают концептуальный, логический, внутренний и внешний уровни представления данных БД, которым соответствуют модели аналогичного назначения.

### 1.5.3. Концептуальная модель

Процесс создания ИС начинается с определения концептуальных требований пользователей системы. Требования отдельных пользователей интегрируются в едином обобщенном представлении – концептуальной модели [16, с. 82].

Концептуальный уровень соответствует логическому аспекту представления данных предметной области в интегрированном виде. Концептуальная модель представляет

объекты и их взаимосвязи без указания способов их физического хранения, т.е. она не зависит от СУБД, в которой будет реализована БД, средств вычислительной техники.

Модель является представлением предметной области, в ней отображается взаимосвязь между данными и их структура, обеспечивается достаточность информации, исключая её избыточность.

Для построения концептуальной модели применяется метод «Сущность–Связь» (англ. Entity Relationship). Используемые в данном методе диаграммы «Сущность–Связь» (англ. Entity Relationship Diagram, ERD) позволяют разрабатывать МД и обеспечивают стандартный способ определения данных и отношения между ними, включает в себя сущности и взаимосвязи, отражающие основные правила предметной области. Такая диаграмма не слишком детализирована, в нее включаются основные сущности и связи между ними, которые удовлетворяют основным требованиям, предъявляемым к ИС. Как правило, диаграммы «Сущность–Связь» используется для презентации и обсуждения структуры данных с экспертами предметной области.

#### *1.5.4. Логическая модель*

Логический уровень представляет собой абстрактный взгляд на данные, которые представляются согласно реальному миру. Логическая модель является отображением концептуальной модели БД, реализация которой возможна в выбранной СУБД. Модель будет отражать логические связи между элементами данных вне зависимости от их содержания и среды хранения.

Логическая МД может быть реляционной, иерархической, сетевой или объектно-ориентированной. Диаграмма «Сущность–Связь», отражающая логическую модель, включает описание всех сущностей с указанием первичных и внешних ключей, а также дополняется диаграммой описания данных, определяющей каждое поле в БД (имя, тип и т.д.).

Логическая модель БД представляет собой полную атрибутивную модель, которая дает наиболее детальное представление структуры данных.

Логическая модель отображается в физическую модель БД, которая определяет размещение данных, методы доступа к данным и технику индексирования.

#### *1.5.5. Внутренняя (физическая) модель*

Внутренний уровень отображает требуемую организацию данных в среде хранения и соответствует физическому аспекту представления данных. Внутренняя модель состоит из отдельных экземпляров записей, физически хранимых во внешних носителях [16, с. 83].

Физическая модель данных зависит от конкретной СУБД. В ней содержится информация обо всех объектах БД. Физическая структура БД прозрачна для конкретного пользователя системы и определяется ядром СУБД.

#### *1.5.6. Внешняя модель*

Внешний уровень поддерживает частные представления данных, требуемые конкретным пользователями. Внешняя модель является подмножеством концептуальной модели. Возможно пересечение внешних моделей по данным. Частная логическая структура данных для отдельного

приложения или пользователя соответствует внешней модели или подсхеме БД. С помощью внешних моделей поддерживается санкционированный доступ к данным БД приложений [16, с. 79].

Появление новых или изменение информационных потребностей существующих приложений требуют определения для них корректных внешних моделей, при этом на уровне концептуальной, логической и внутренней модели данных изменений не происходит. Изменения в концептуальной, логической модели, вызванные появлением новых видов данных или изменением их структур, могут затрагивать не все приложения, т.е. обеспечивается определенная независимость программ от данных. Изменения в логической модели должны отражаться на внутренней модели, и при неизменной логической модели возможна самостоятельная модификация внутренней модели БД с целью улучшения ее характеристик. Таким образом, БД реализует принцип относительной независимости логической и физической организации данных.

#### 1.5.7. Типы связей

Все ИО предметной области связаны между собой. Различаются несколько типов связей, для которых введены следующие обозначения [20, с. 477]:

- один к одному (1:1);
- один ко многим (1:M);
- многие ко многим (M:M).

Пример. Дана совокупность ИО, отражающих учебный процесс в вузе:

- *Студент* (Номер студента, Фамилия, Имя, Отчество);

- *Сессия* (Номер студента, Оценка1, Оценка2, Оценка3, Оценка4, Результат);
- *Стипендия* (Результат, Процент);
- *Преподаватель* (Номер преподавателя, Фамилия, Имя, Отчество).

Связь один к одному (1:1) предполагает, что в каждый момент времени одному экземпляру ИО А соответствует не более одного экземпляра ИО В и наоборот.

Примером связи 1:1 может служить связь между ИО *Студент* и *Сессия*: каждый студент имеет один и только один определенный набор экзаменационных оценок в сессию.

Связь один ко многим (1:M) предполагает, что одному экземпляру ИО А соответствует 0, 1 или более экземпляров ИО В, но каждый экземпляр ИО В связан не более чем с 1 экземпляром ИО А.

Примером связи 1:M служит связь между ИО *Стипендия* и *Сессия*: установленный размер стипендии по результатам сдачи сессии может повторяться многократно для различных студентов.

Связь многие ко многим (M:M) предполагает, что в каждый момент времени одному экземпляру ИО А соответствует 0, 1 или более экземпляров ИО В и наоборот.

Примером является связь между ИО *Студент* и *Преподаватель*: один студент обучается у многих преподавателей, один преподаватель обучает многих студентов.

На рис. 15 представлена концептуальная модель для представленного ранее примера (на рисунке используются

следующие обозначения: РК – первичный ключ, ВК – внешний ключ).

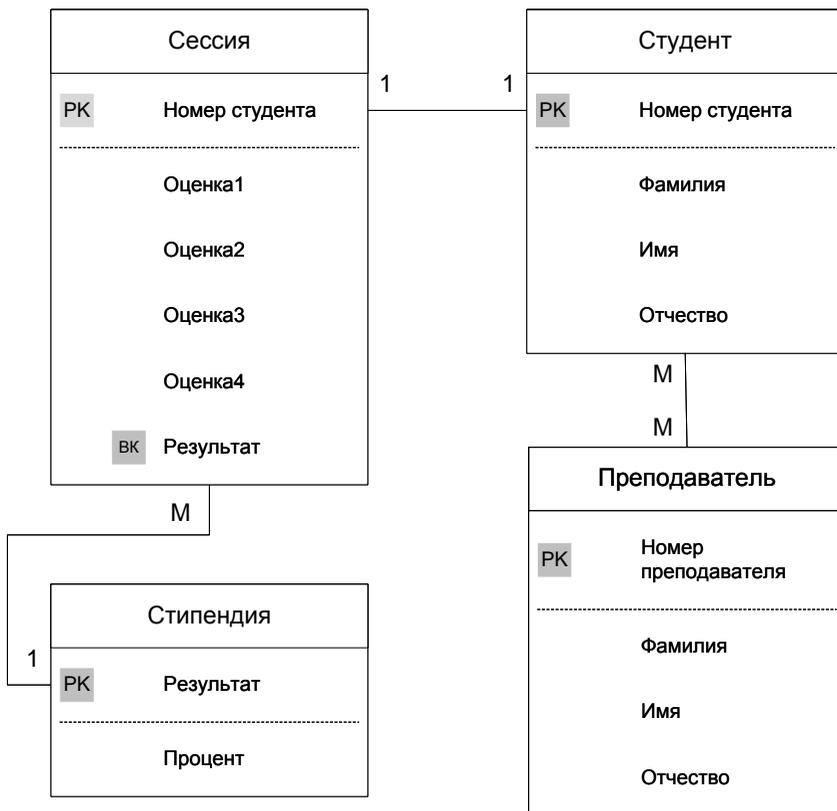


Рис. 15. Концептуальная модель

## 1.6. Обзор Microsoft Office Access

Офисный пакет приложений Microsoft Office является самым распространенным пакетом автоматизации работы с данными. СУБД Microsoft Office Access (далее в тексте Access), входящая в этот пакет, стала стандартной средой создания БД, используемой в современном бизнесе. Характерной особенностью Access является наличие многочисленных и разнообразных мастеров, которые помогают при выполнении большинства действий, не требуя от пользователя глубоких знаний в области БД [2].

После запуска Access появляется окно (см. рис. 16), в котором пользователь может открыть ранее открывавшиеся файлы, создать БД рабочего стола или веб-приложение. В последних двух случаях возможно использование разнообразных шаблонов, которые содержат встроенные таблицы, запросы, формы и отчеты, готовые к работе.

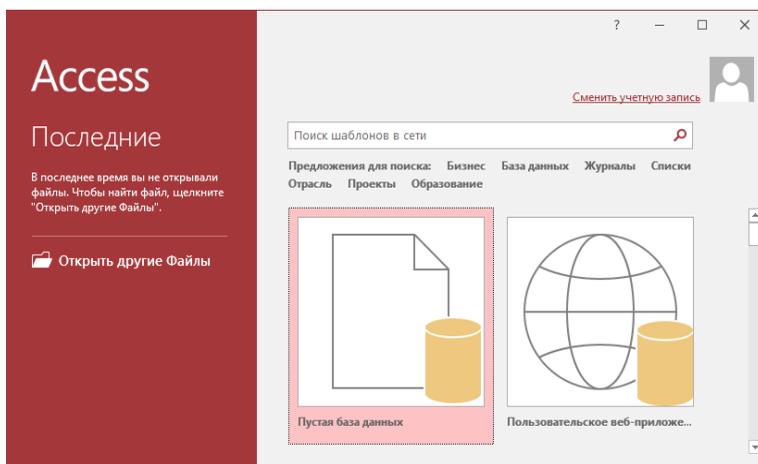


Рис. 16. Начальное окно Access

Мы будем рассматривать только создание классических БД. При создании пустой БД необходимо указать имя файла и нажать кнопку «Создать» (см. рис. 17).

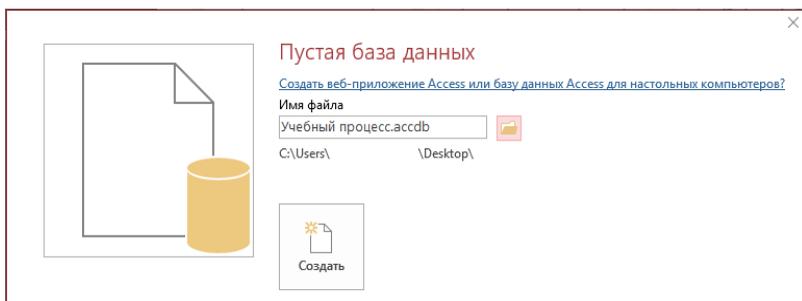


Рис. 17. Создание пустой БД

После создания БД пользователю будет показана пустая таблица, открытая в режиме таблицы, в которую можно добавлять данные (рис. 18).

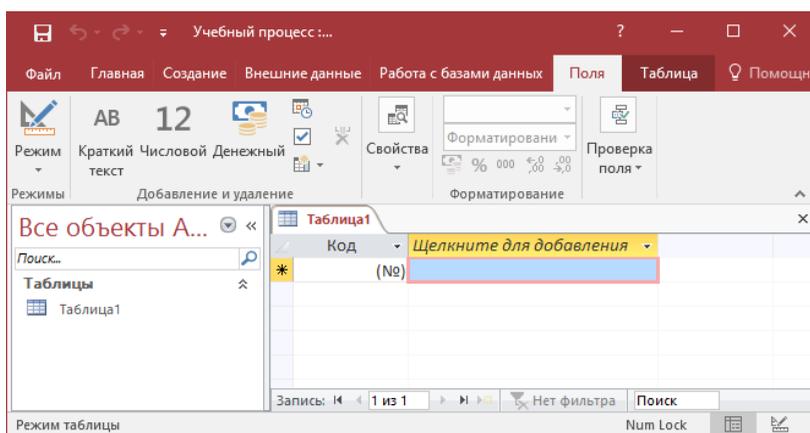


Рис. 18. Пустая таблица для ввода данных

Access предоставляет пользователю создание следующих объектов:

- *таблицы* – содержат данные;
- *запросы* – выборка данных из одной или нескольких таблиц, при которой используются некоторые критерии отбора, (фактически запросы – это представления/виды);
- *формы* – используются для организации интерфейса взаимодействия с пользователем БД;
- *отчеты* – выборка данных из таблиц/запросов, отформатированная для вывода на печать.

### 1.6.1. Создание таблицы

Создание таблицы в режиме *Таблица* осуществляется путем ввода данных в пустую таблицу. После сохранения таблицы остаются только те столбцы, которые были переименованы или в которые были введены данные (см. рис. 19).

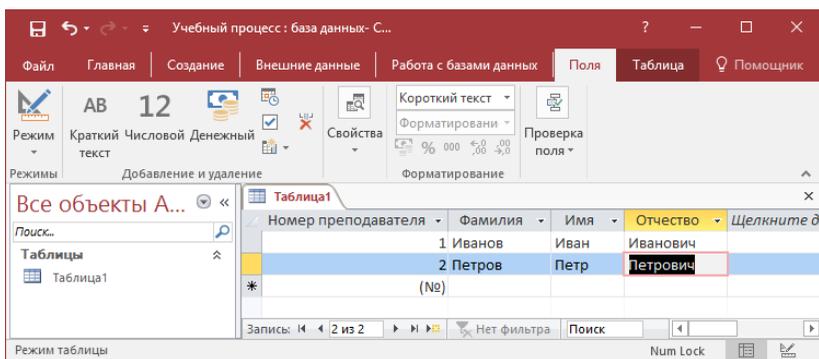


Рис. 19. Создание таблицы в режиме Таблица

Также таблицы можно создавать в режиме конструктора, в котором описывается список ее столбцов: указывается имя поля и тип данных, длина поля, является ли поле

первичным ключом таблицы, допустимо ли указание в поле пустых значений и т.д. (см. рис. 20).

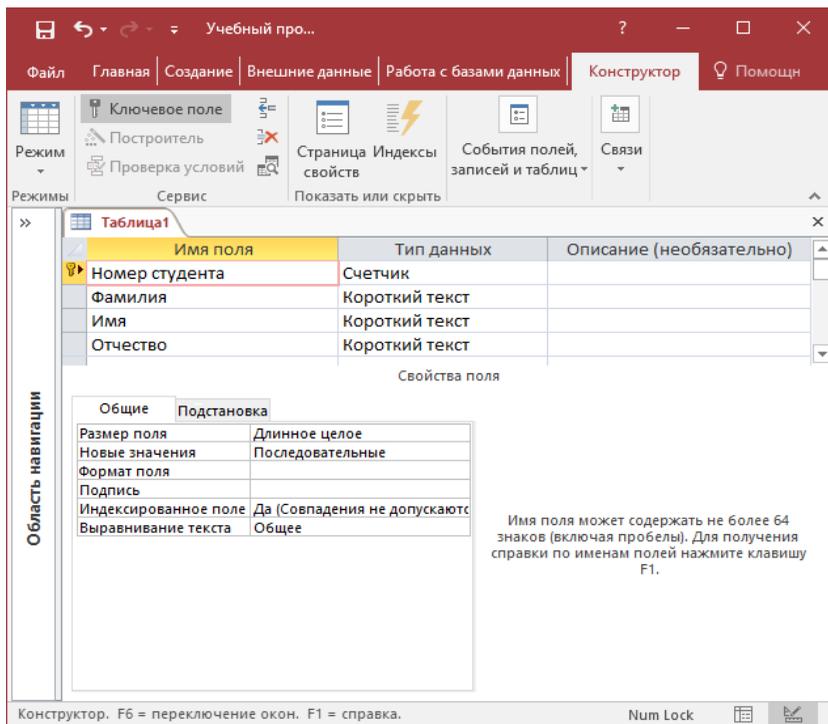


Рис. 20. Создание таблицы в режиме конструктора

Можно создавать связи с различными внешними источниками данных, такими как другие БД, текстовые файлы и книги Excel. Access может использовать связи с внешними данными, как если бы они были таблицами. В зависимости от внешнего источника данных и способа создания связи можно изменять данные в связанной таблице и создавать отношения с ней. Однако изменять структуру внешних данных с помощью связи нельзя.

В БД можно создать таблицу, и настроить в нее импорт данных из списка SharePoint или создать связь с ним. Можно также создать новый список SharePoint на основе готового шаблона.

Также можно создать таблицу, подключенную к данным на веб-сайте с интерфейсом веб-службы.

При создании таблицы в режиме конструктора или при последующем редактировании таблицы можно создать столбец подстановок. Столбец подстановок – это столбец, в который значения вводятся не вручную, а из раскрывающегося списка.

Создадим столбец подстановок для поля «Номер студента» таблицы «Сессия» (см. рис. 21).

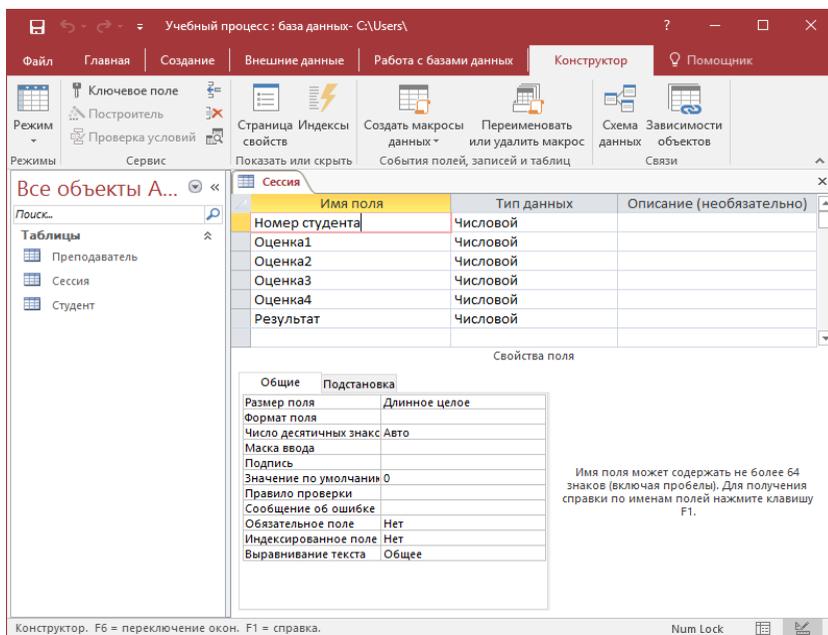


Рис. 21. Таблица «Сессия»

Выберем для поля «Номер студента» тип данных *Мастер подстановок...* (см. рис. 22).

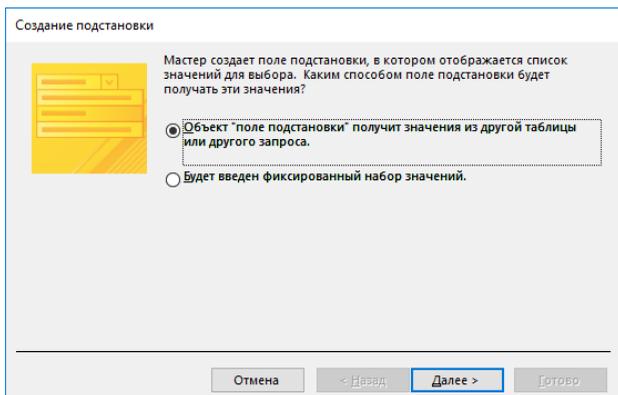


Рис. 22. Окно «Создание подстановки»

Можно ввести фиксированный набор значений или значения будут выбираться из другой таблицы или другого запроса. Остановимся на выборе значений из другой таблицы – таблицы *Студент* (см. рис. 23).

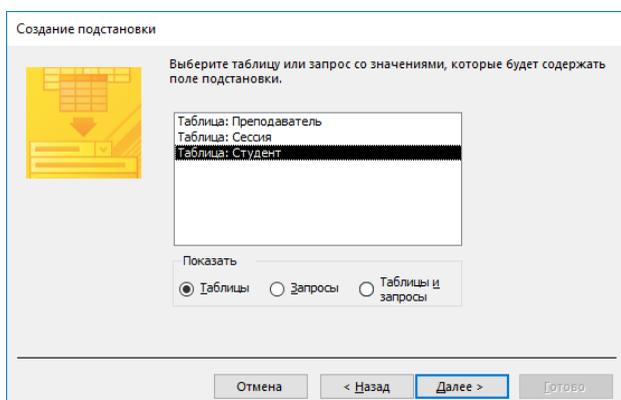


Рис. 23. Выбор таблицы

В качестве полей подстановки выберем поля *Фамилия*, *Имя* и *Отчество* (см. рис. 24).

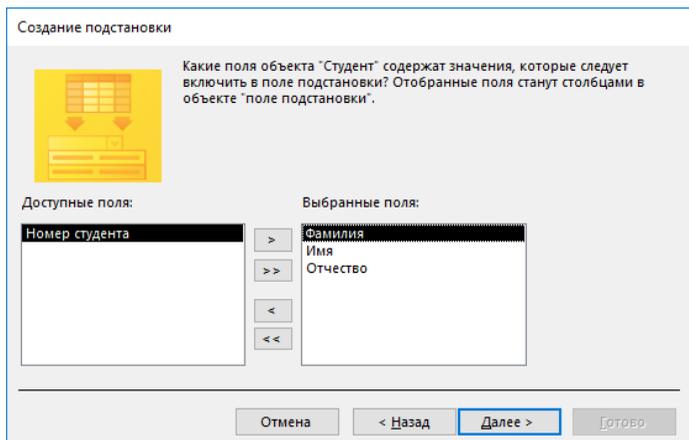


Рис. 24. Выбор полей

Настроим сортировку записей в алфавитном порядке согласно рис. 25.

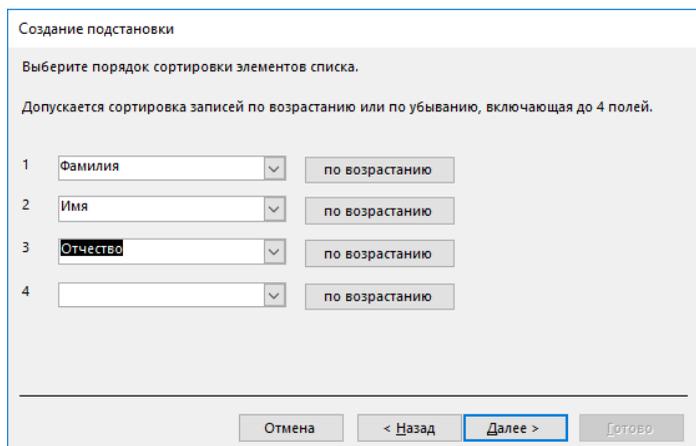


Рис. 25. Настройка сортировки

Зададим ширину столбцов (см. рис. 26).

Создание подстановки

Задайте ширину столбцов, которые содержит поле подстановки.

Перетащите правую границу заголовка столбца на нужную ширину или дважды щелкните ее для автоматического подбора ширины.

Скрыть ключевой столбец (рекомендуется)

Фамилия	Имя	Отчество			

Отмена < Назад Далее > Готово

Рис. 26. Настройка ширины столбцов

Зададим подпись для поля подстановки *ФИО студента* (см. рис. 27).

Создание подстановки

Задайте подпись, которую содержит поле подстановки.

ФИО студента

Включить проверку целостности данных для этих таблиц?

Включить проверку целостности данных

Каскадное удаление

Ограничить удаление

Сохранить несколько значений для этой подстановки?

Разрешить несколько значений

Указаны все сведения, необходимые мастеру, чтобы создать поле подстановки.

Отмена < Назад Далее > Готово

Рис. 27. Задание подписи для поля подстановки

Поле подстановки *ФИО студента* определим как ключевое (см. рис. 28).

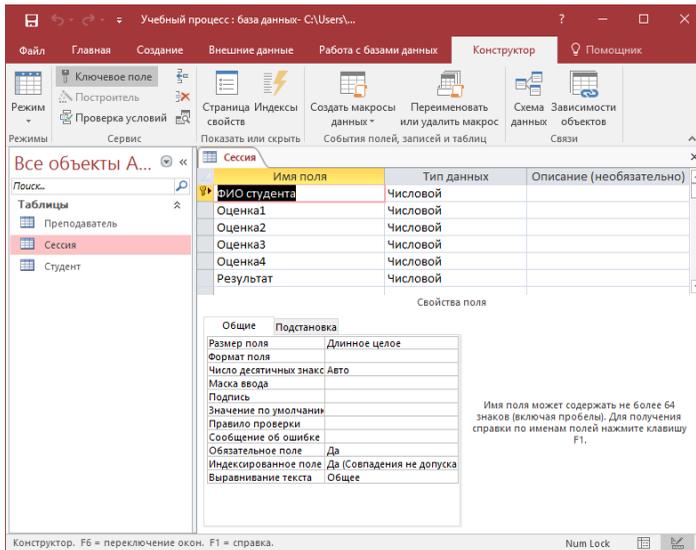


Рис. 28. Задание ключевого поля

Теперь при выборе поля *ФИО студента* будет появляться выпадающий список (см. рис. 29).

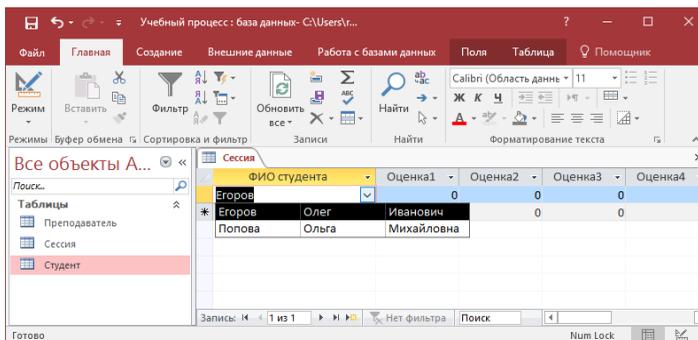


Рис. 29. Пример работы с полем подстановки

Сделаем более удобный вид для поля подстановки *ФИО студента*: для этого откроем таблицу *Сессия* в режиме конструктора, выберем поле *ФИО студента* и на вкладке *Подстановка* в поле *Источник строк* откорректируем SQL запрос (табл. 4) и уменьшим значение поля *Число столбцов* с 4-х до 2-х.

Таблица 4

### SQL запрос

<b>Исходное значение</b>	SELECT [Студент].[Номер студента], [Студент].[Фамилия], [Студент].[Имя], [Студент].[Отчество] FROM Студент ORDER BY [Фамилия], [Имя], [Отчество];
<b>Результующее значение</b>	SELECT [Студент].[Номер студента], [Студент].[Фамилия]&" "&[Студент].[Имя]&" "&[Студент].[Отчество] FROM Студент ORDER BY [Фамилия], [Имя], [Отчество];

Удобная работа с полем подстановки показана на рис. 30.

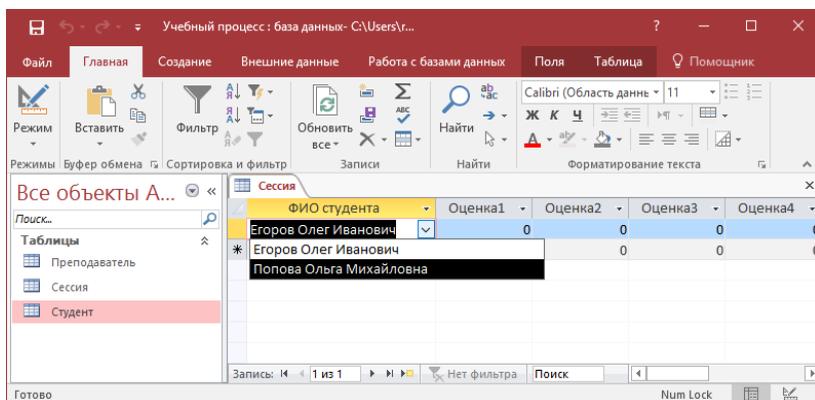


Рис. 30. Пример работы с полем подстановки

При работе с таблицами Access предусмотрена возможность сортировки и фильтрации данных. Можно установить фильтрацию на основании выделенного значения, или воспользоваться *Расширенным фильтром* для задания сложных условий фильтрации с использованием интерфейса, аналогичного интерфейсу при создании запросов. Сортировка данных по одному столбцу возможна при помощи кнопок *По возрастанию* или *По убыванию*. Однако если необходима сортировка по двум или более столбцам одновременно, то необходимо воспользоваться возможностями сортировки через расширенный фильтр.

### 1.6.2. Схема данных

Схема данных позволяет определить связи между таблицами, используя понятия *первичный ключ* и *внешний ключ* реляционной МД. Она также позволяет установить правила каскадного удаления и обновления строк в связанных таблицах.

Схема данных доступна на вкладке *Работа с базами данных*. Пример схемы данных представлен на рис. 31.

Для установления связей между таблицами необходимо перетащить поле одной таблицы на связываемое поле другой таблицы (поля в обеих таблицах не обязательно должны иметь одинаковые имена, но они должны иметь одинаковые типы данных и значения полей должны содержать одну и ту же информацию).

Если установить флажок *Обеспечение целостности данных*, то Access не позволит добавить в связанную таблицу записи, для которых нет соответствующих записей в главной таблице или же изменить (удалить) записи в главной

таблице, для которых имеются подчиненные записи в связанной таблице.

Если дополнительно установить флажок *Каскадное обновление* связанных полей, любое изменение значения в ключевом поле главной таблицы приведет к автоматическому обновлению соответствующих значений во всех связанных записях. Флажок *Каскадное удаление* – удаление записи в главной таблице приведет к автоматическому удалению связанных записей в подчиненной таблице.

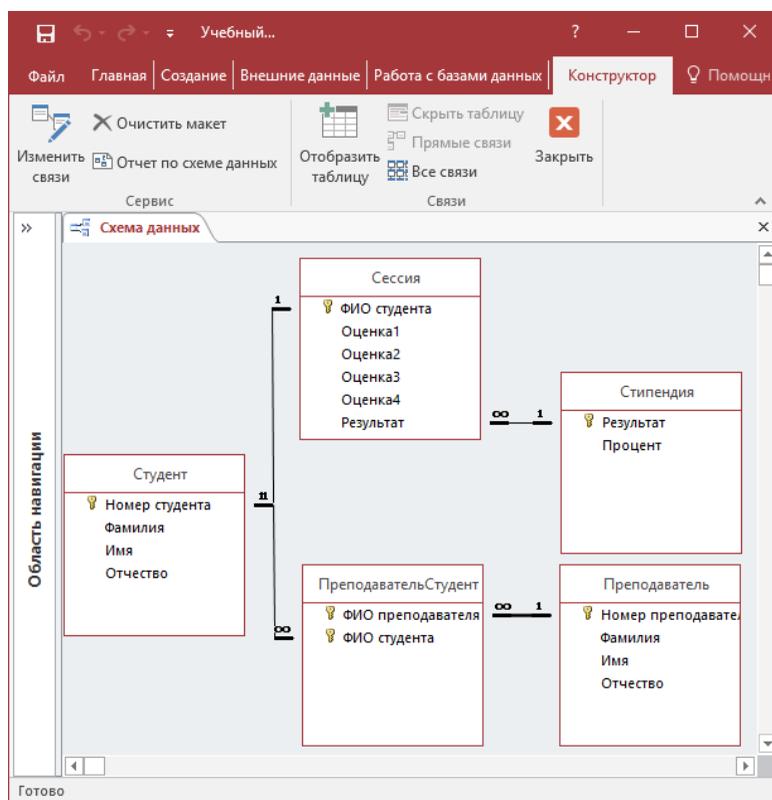


Рис. 31. Схема данных

Тип создаваемой в схеме данных связи зависит от полей, для которых определяется связь:

- Отношение «один ко многим» создается в том случае, когда только одно из полей является ключевым или имеет уникальный индекс.
- Отношение «один к одному» создается в том случае, когда оба связываемых поля являются ключевыми или имеют уникальные индексы.

### *1.6.3. Создание запроса*

Наиболее гибким вариантом является создание запроса в режиме конструктора, при этом задается имя таблицы, имена отбираемых столбцов, условия отбора, сортировки и вывода на экран. Условия, находящиеся в одной строке конструктора объединяются логическим оператором «и», а в разных строках – оператором «или». Можно также создать запрос по одному из следующих вариантов:

- простой запрос – позволяет указать таблицы, из которых выбираются данные и имена столбцов;
- перекрестный запрос – используется при создании в рамках одной таблицы, напоминает сводную таблицу Excel (необходимо указать строки, столбцы, данные и функцию обработки данных). Для того чтобы создать перекрестный запрос на основании нескольких таблиц, необходимо сначала создать простой запрос, содержащий все необходимые столбцы из этих таблиц;
- повторяющиеся значения – выводит повторяющиеся записи из таблицы: указывается столбец, по которому надо искать повторы и другие столбцы, данные из которых надо выводить для каждой из повторяющихся записей (например,

для одинакового «Результата» вывести «ФИО студента»). Если дополнительные столбцы не указаны, то выводится «значение» – «количество повторов»;

- записи без подчиненных – поиск записей, не имеющих связей с записями в другой таблице (например, клиенты, не имеющие заказов).

Помимо создания запроса одним из вышеперечисленных способов, можно задать запрос и непосредственно в виде инструкций SQL. Для этого необходимо перейти из режима *Конструктора запроса* в *Режим SQL*.

Пример создания запроса с помощью конструктора показан на рис. 32.

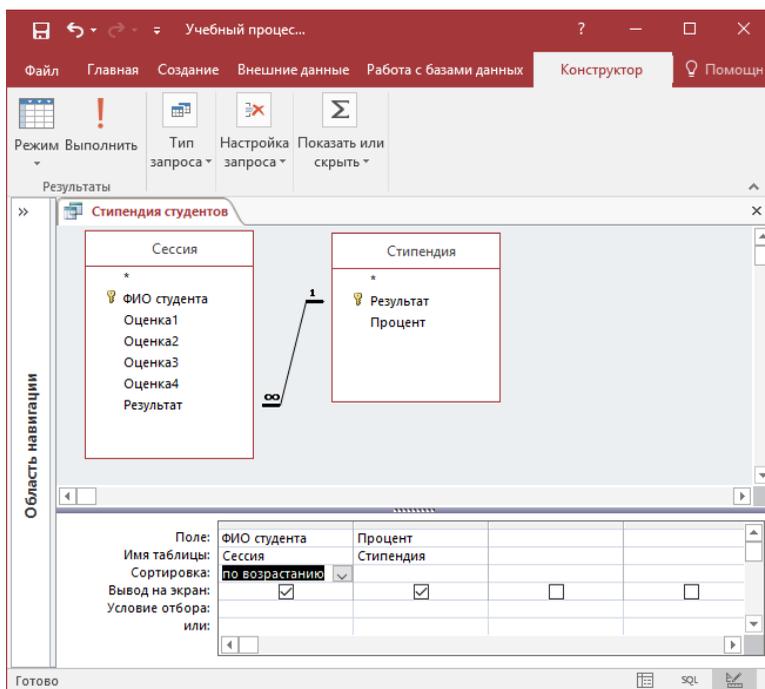


Рис. 32. Создание запроса в режиме конструктора

### 1.6.4. Создание форм

Для создания форм используются инструменты из группы *Форма* вкладки *Создание*. Наиболее гибким инструментом создания форм является конструктор форм, но на начальном этапе проще создавать формы с помощью мастера форм. *Мастер форм* позволяет автоматически создавать формы на основе указания таблицы, на основании которой строится форма, выбор полей таблицы и внешний вид формы: в один столбец (слева – название, справа – данные); ленточная (колонки, сверху названия столбцов); табличная (аналогична листу Excel), выровненная. Возможно также быстрое создание формы по всем полям таблицы с использованием кнопки *Форма* (см. рис. 33).

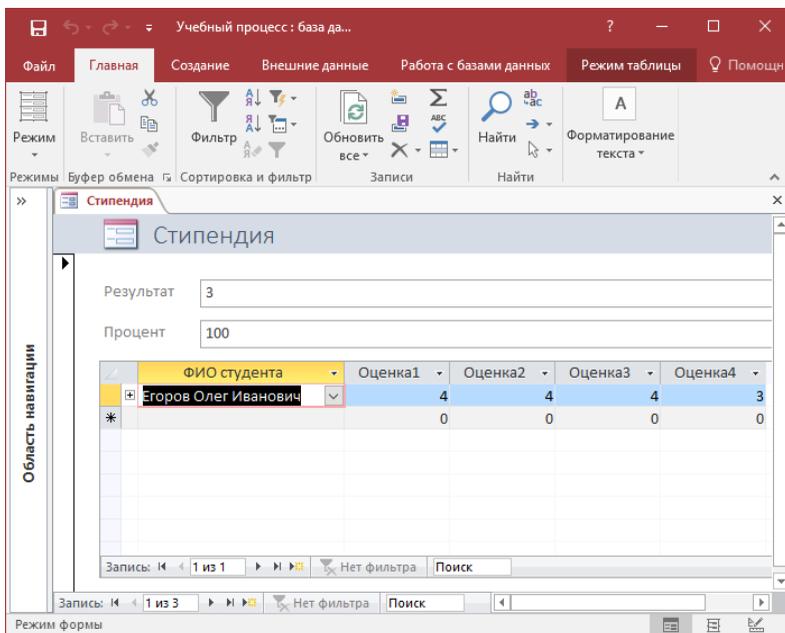


Рис. 33. Пример формы

При создании формы в режиме конструктора в форме выделяется несколько областей:

- заголовок формы;
- колоннитулы;
- область данных;
- примечание формы.

Можно вручную на линейке изменить размеры областей. Используя панель элементов, можно создавать в форме следующие элементы управления:

- *Поле* – поля данных или вычислений;
- *Надпись* – любой текст;

• *Кнопка* – если нажата кнопка *Использовать мастера*, то выводится мастер создания кнопок. Если кнопка *Использовать мастера* не нажата, то необходимо нарисовать кнопку на форме и обработать события: *Нажатие кнопки*, *Двойное нажатие кнопки*, *Получение фокуса* и т.д.;

• *Переключатель*, *Выключатель*, *Флажок*, *Группа переключателей* – в группу входят переключатели или флажки, в зависимости от их состояния, вся группа в целом принимает некоторое значение;

• *Поле со списком*, *Список*, *Вкладка*, *Гиперссылка*, *Рисунок*, *Диаграмма* и др.

#### 1.6.5. Создание отчетов

Создание отчетов аналогично созданию форм. В режиме *Конструктора* выделяется те же области: заголовок отчета, верхний колоннитул, область данных, нижний колоннитул, примечание отчета. Можно создавать отчеты с группировкой, также возможно создание отчетов с использованием мастера отчетов.

### 1.6.6. Связи с Office

Используя инструменты экспорта с вкладки «Внешние данные», возможно слияние базы данных Access с документом Microsoft Word, публикация базы данных Access в Microsoft Word (преобразование таблицы Access в таблицу Word), анализ в Microsoft Excel (преобразование таблицы Access в файл Excel) и т.д.

Слияние документов позволяет использовать базу данных Access для создания большого количества однотипных документов Word. Например, можно создать в Word шаблон приглажительного письма, данные в которое будут подставляться из базы данных Access.

### 1.6.7. Макросы

Макрокоманды, которые можно объединять в макросы, совершают разнообразные действия, выполнимые в Access, а с помощью параметров этим действиям можно придать гибкость, которой иначе можно добиться только путем программирования (см. рис. 34).

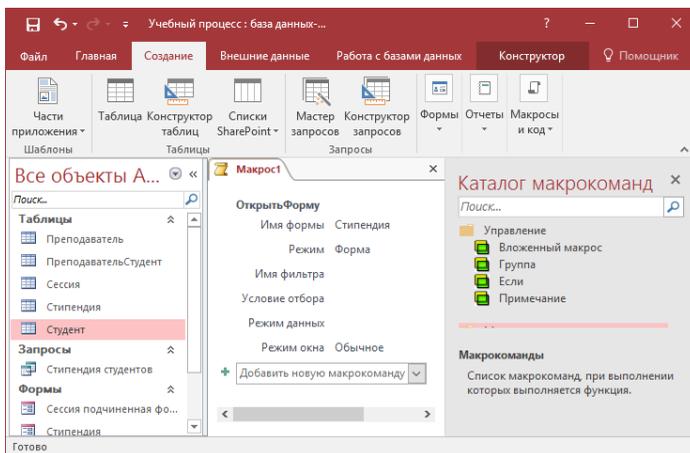


Рис. 34. Создание макроса

Для построения более сложных макросов, требующих выполнения определенных условий для запуска отдельных макрокоманд, необходимо добавить блок *Если*. Также есть дополнительные блоки *Группа*, *Вложенный макрос* и *Примечание*.

### **Контрольные вопросы**

1. Дайте определения понятиям: ИС, БД, СУБД.
2. В чем заключается основная цель создания ИС?
3. В каких сферах человеческой деятельности применяются ИС?
4. Охарактеризуйте основные виды ИС. Приведите примеры.
5. Охарактеризуйте основные типы ИС.
6. Что входит в состав банка данных?
7. Перечислите основные функции банков данных.
8. Какие этапы входят в процесс создания ИС?
9. Какие подсистемы образуют структуру ИС?
10. В чем заключается назначение подсистемы информационного обеспечения?
11. Что входит в состав подсистемы технического обеспечения?
12. Что относят к средствам математического обеспечения?
13. Какие функции реализует организационное обеспечение?
14. В чём заключается главная цель правового обеспечения?
15. Что входит в понятие архитектуры ИС?

16. Охарактеризуйте одиночную ИС.
17. Охарактеризуйте архитектуру ИС «Файл-сервер», приведите в пример ее недостатки и достоинства.
18. Охарактеризуйте архитектуру ИС «Клиент-сервер», приведите в пример ее недостатки и достоинства.
19. Каким образом осуществляется типовое взаимодействие сервера БД и приложения-клиента в архитектуре ИС «Клиент-сервер»?
20. Охарактеризуйте многоуровневую архитектуру ИС, приведите в пример ее недостатки и достоинства.
21. Охарактеризуйте интернет/интранет архитектуру ИС, приведите в пример ее недостатки и достоинства.
22. Какую архитектуру ИС называют также архитектурой с «тонким» клиентом?
23. Какую архитектуру ИС называют также архитектурой с «толстым» клиентом?
24. Дайте определение МД.
25. Охарактеризуйте иерархическую МД. Приведите примеры.
26. Охарактеризуйте сетевую МД. Приведите примеры.
27. Охарактеризуйте реляционную МД. Приведите примеры.
28. Дайте определения понятиям реляционной МД: отношение, схема отношения, степень, мощность, первичный ключ, кортеж.
29. Какими свойствами должен обладать первичный ключ?
30. Поясните использование естественного и суррогатного ключа в качестве первичного.

31. Перечислите достоинства суррогатных ключей.
32. Каким образом осуществляется связь между двумя отношениями в реляционной МД?
33. Перечислите условия и ограничения, накладываемые на отношения, выполнение которых позволяет таблицу считать отношением.
34. Охарактеризуйте объектно-ориентированную МД. Приведите примеры.
35. Охарактеризуйте модель объектов-ролей.
36. Дайте определения понятиям: ИО, ИЛМ, концептуальная модель, логическая модель, внутренняя (физическая) модель, внешняя модель.
37. Приведите пример ИЛМ с использованием следующих типов связей: один к одному, один ко многим и многие ко многим.
38. Какие основные объекты позволяет создавать пользователю Access?
39. Какими способами можно создавать таблицы в Access?
40. Для чего нужна схема данных в Access?
41. Какими способами можно создавать запросы в Access?
42. Какими способами можно создавать формы в Access?
43. Какими способами можно создавать отчеты в Access?
44. Можно ли из Access передать данные в Microsoft Word? В Microsoft Excel?
45. Что позволяют реализовывать макросы в Access?

### **Задания для самостоятельного выполнения**

1. Создайте БД «Учет семейного бюджета», содержащую следующие таблицы:

- *Член семьи* (Код члена семьи, Фамилия, Имя, Отчество, Степень родства);
- *Статья расхода* (Код статьи расхода, Наименование статьи расхода);
- *Статья дохода* (Код статьи дохода, Наименование статьи дохода);
- *Расход* (Код члена семьи, Код статьи расхода, Дата расхода, Сумма расхода);
- *Доход* (Код члена семьи, Код статьи дохода, Дата дохода, Сумма дохода).

Ввод данных организуйте с помощью форм.

Создайте запросы:

- вывести все расходы за определенный месяц;
- вывести все доходы определенного члена семьи.

Сформируйте отчеты:

- по статьям расходов и членам семьи с итогами расходов;
- по статьям доходов и членам семьи с итогами доходов;

2. Создайте БД «Учет переселенцев», содержащую следующие таблицы:

- *Область* (Код области, Наименование области);
- *Населенный пункт* (Код населенного пункта, Наименование населенного пункта, Код области);
- *Национальность* (Код национальности, Наименование национальности);

- *Переселенец* (Код переселенца, Фамилия, Имя, Отчество, Дата рождения, Код населенного пункта, Код национальности).

Ввод данных организуйте с помощью форм.

Создайте запросы:

- вывести всех переселенцев из конкретного населенного пункта;

- вывести количество переселенцев по каждой национальности.

Сформируйте отчеты:

- по областям и населенным пунктам с итогами (число переселенцев);

- по национальности с итогами (число переселенцев).

3. Создайте БД «Учет производства сельскохозяйственных культур», содержащую следующие таблицы:

- *Район* (Код района, Наименование района);

- *Сельскохозяйственное предприятие* (Код сельскохозяйственного предприятия, Наименование сельскохозяйственного предприятия, Адрес, Телефон);

- *Сельскохозяйственная культура* (Код сельскохозяйственной культуры, Наименование сельскохозяйственной культуры);

- *Выращивание* (Код района, Код сельскохозяйственного предприятия, Код сельскохозяйственной культуры, Площадь, Урожай).

Ввод данных организуйте с помощью форм.

Создайте запросы:

- вывести все предприятия, выращивающие определенную сельскохозяйственную культуру;

- вывести итоговые данные по урожаю каждой сельскохозяйственной культуры.

Сформируйте отчеты:

- по сельскохозяйственным культурам и районам с итогами (урожаи и площади);

- по предприятиям и сельскохозяйственным культурам с итогами (урожаи и площади).

4. Создайте БД «Учет выпуска изделий», содержащую следующие таблицы:

- *Город* (Код города, Наименование города);

- *Предприятие* (Код предприятия, Наименование предприятия, Адрес, Телефон, Код города);

- *Изделие* (Код изделия, Наименование изделия);

- *Выпуск* (Код предприятия, Код изделия, Дата выпуска, Количество изделий, Отпускная цена).

Ввод данных организуйте с помощью форм.

Создайте запросы:

- вывести все предприятия, выпускающие определенное изделие;

- вывести итоговые данные по количеству выпускаемых изделий.

Сформируйте отчеты:

- по предприятиям и изделиям с итоговой стоимостью выпущенных изделий;

- по городам и изделиям с итоговой стоимостью выпущенных изделий.

5. Создайте БД «Учет поставок товаров», содержащую следующие таблицы:

- *Город* (Код города, Наименование города);

- *Предприятие* (Код предприятия, Наименование предприятия, Адрес, Телефон, Код города);
- *Товар* (Код товара, Наименование товара);
- *Поставка* (Код предприятия, Код товара, Дата поставки, Количество, Единица измерения, Цена).

Ввод данных организуйте с помощью форм.

Создайте запросы:

- вывести всех поставщиков определенного товара;
- вывести итоговые данные по количеству поставляемых товаров.

Сформируйте отчеты:

- по городам и поставщикам с итоговой стоимостью поставленных товаров;
- по городам и товарам с итоговой стоимостью поставленных товаров.

6. Создайте БД «Учет сбросов отравляющих веществ в окружающую среду», содержащую следующие таблицы:

- *Город* (Код города, Наименование города);
- *Организация* (Код организации, Наименование организации, Адрес, Телефон, Код города);
- *Отравляющее вещество* (Код отравляющего вещества, Наименование отравляющего вещества);
- *Сброс* (Код организации, Код отравляющего вещества, Дата сброса, Концентрация, Размер сброса, Единица измерения, Сумма ущерба).

Ввод данных организуйте с помощью форм.

Создайте запросы:

- вывести все организации, осуществившие сброс в определенный месяц;
- вывести итоговые данные по размерам сбросов.

Сформируйте отчеты:

- по городам и предприятиям с итоговыми суммами ущерба;
- по городам и сбросам с итоговыми размерами сбросов.

7. Создайте БД «Учет рождаемости», содержащую следующие таблицы:

- *Область* (Код области, Наименование области);
- *Населенный пункт* (Код населенного пункта, Наименование населенного пункта, Код области);
- *Национальность* (Код национальности, Наименование национальности);
- *Новорожденный* (Код новорожденного, Фамилия, Имя, Отчество, Дата рождения, Вес, Рост, Код населенного пункта, Код национальности).

Ввод данных организуйте с помощью форм.

Создайте запросы:

- вывести всех новорожденных определенного населенного пункта, родившихся в указанную дату;
- вывести итоговые данные количеству новорожденных по каждому населенному пункту.

Сформируйте отчеты:

- по областям и населенным пунктам с итогами (число новорожденных);
- по национальностям и населенным пунктам с итогами (число новорожденных).

8. Создайте БД «Учет проданных товаров», содержащую следующие таблицы:

- *Город* (Код города, Наименование города);

- *Предприятие* (Код предприятия, Наименование предприятия, Адрес, Телефон, Код города);
- *Товар* (Код товара, Наименование товара);
- *Продажа* (Код предприятия, Код товара, Дата продажи, Количество, Единица измерения, Цена).

Ввод данных организуйте с помощью форм.

Создайте запросы:

- вывести все товары, проданные в указанную дату;
- вывести итоговые данные количества проданных товаров каждого типа.

Сформируйте отчеты:

- по торгующим предприятиям и типам товаров с итогами (стоимость проданного товара);
- по типам товаров и наименованиям товаров с итогами (стоимость проданного товара).

9. Создайте БД «Учет малых предприятий», содержащую следующие таблицы:

- *Область* (Код области, Наименование области);
- *Город* (Код города, Наименование города, Код области);
- *Вид деятельности* (Код вида деятельности, Наименование вида деятельности);
- *Предприятие* (Код предприятия, Наименование предприятия, Адрес, Телефон, Численность, Объем выполненных работ за год, Дата создания предприятия, Код вида деятельности, Код города);

Ввод данных организуйте с помощью форм.

Создайте запросы:

- вывести все малые предприятия определенного населенного пункта;

- вывести итоговые данные по количеству малых предприятий каждого населенного пункта.

Сформируйте отчеты:

- по областям и населенным пунктам с итогами (численность и объем выполненных за год работ);
- по населенным пунктам и малым предприятиям с итогами (численность и объем выполненных за год работ).

10. Создайте БД «Учет больных в больнице», содержащую следующие таблицы:

- *Страховая компания* (Код страховой компании, Наименование страховой компании);

- *Пациент* (Код пациента, Фамилия, Имя, Отчество, Дата рождения, Адрес, Код страховой компании);

- *Диагноз* (Код диагноза, Наименование диагноза);

- *Подразделение* (Код подразделения, Наименование подразделения);

- *Лечение* (Код пациента, Код диагноза, Код подразделения, Дата начала лечения, Дата выписки, Стоимость лечения за один день).

Ввод данных организуйте с помощью форм.

Создайте запросы:

- вывести всех больных указанной страховой компании;

- вывести итоговые данные по количеству больных каждого подразделения.

Сформируйте отчеты:

- по страховым компаниям и подразделениям с итогами стоимости лечения;

- по страховым компаниям и подразделениям с итогами по количеству больных.

11. Создайте БД «Учет кадров», содержащую следующие таблицы:

- *Подразделение* (Код подразделения, Наименование подразделения);
- *Специальность* (Код специальности, Наименование специальности);
- *Должность* (Код должности, Наименование должности);
- *Сотрудник* (Код сотрудника, Фамилия, Имя, Отчество, Дата рождения, Адрес, Образование, Дата начала трудовой деятельности, Код специальности);
- *Работа* (Код сотрудника, Код должности, Код подразделения, Дата приема, Оклад).

Ввод данных организуйте с помощью форм.

Создайте запросы:

- вывести всех сотрудников указанного подразделения;
- вывести итоговые данные по количеству сотрудников по каждому подразделению.

Сформируйте отчеты:

- по подразделениям и должностям с итогами (число сотрудников);
- по подразделениям с указанием итоговых данных по окладам.

12. Создайте БД «Учет успеваемости школьников», содержащую следующие таблицы:

- *Предмет* (Код предмета, Наименование предмета);
- *Школьник* (Код школьника, Фамилия, Имя, Отчество, Дата рождения, Пол, Адрес, Класс);

- *Учитель* (Код учителя, Фамилия, Имя, Отчество, Дата рождения, Пол, Адрес);

- *Успеваемость* (Код предмета, Код школьника, Код учителя, Дата, Оценка).

Ввод данных организуйте с помощью форм.

Создайте запросы:

- вывести всех школьников определенного класса;
- вывести итоговые данные по количеству школьников в каждом классе.

Сформируйте отчеты:

- по классам и предметам со средней оценкой;
- по классам и школьникам со средней оценкой.

## 2. ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

### 2.1. Основные понятия искусственного интеллекта

Термин *интеллект* (intelligence) был предложен в 1956 году на семинаре с аналогичным названием в Дартмутском колледже (США). Термин происходит от латинского *intellectus* – что означает ум, рассудок, разум; мыслительные способности человека.

«Интеллект – совокупность умственных способностей человека и некоторых высших животных, относительно устойчивая система умственных действий, операций и способностей, определяющая когнитивный стиль субъекта, направленная на формирование адекватной схемы окружающего мира и образа-Я» [50, с. 281].

Историю развития искусственного интеллекта, насчитывающую несколько веков, принято связывать с этапами развития компьютерной техники (см. табл. 5). Из таблицы видно, что каждый этап развития компьютеров сопровождался рождением соответствующих технологий искусственного интеллекта, порождая свои специфические понятия, термины и методы, пытаясь дать описание философской идеи о замене человека умной машиной. «Технология следующего уровня чаще всего их заимствует, однако возникают и новые понятия, образующие новую теорию» [56, с. 14].

Интеллектуальной называется система, способная целенаправленно, в зависимости от состояния информационных

входов, изменять не только параметры функционирования, но и сам способ своего поведения, причем способ поведения зависит не только от текущего состояния информационных входов, но также и от предыдущих состояний системы (см. рис. 35).



Рис. 35. Построение интеллектуальной системы

Следует различать «...целеустремленные и целенаправленные системы. Примером первого типа может служить артиллерийский выстрел, второй – самонаводящаяся ракета» [57, с. 15].

«Искусственный интеллект – это наука о концепциях, позволяющих вычислительным машинам делать такие вещи, которые у людей выглядят разумными... Центральные задачи искусственного интеллекта состоят в том, чтобы сделать вычислительные машины более полезными и понять принципы, лежащие в основе интеллекта...» [57, с. 56].

А.Д. Дубровин предлагает вместо термина «искусственный интеллект» использовать термин «компьютерная модель человеческого интеллекта», т.к. интеллект человека моделируется с помощью компьютера, а любая модель всегда является лишь подобием реальности, описывая только некоторые рассматриваемые характеристики [17].

Таблица 5

### Этапы развития компьютерной техники

Название этапа	Классы решаемых задач, примеры	Имена ученых, инженеров	Техника
1	2	3	4
Домашинный (XVII в.)	Автоматизация расчетов (в основном арифметических)	Г. Лейбниц Б. Паскаль Р. Луллий	Арифмометр Логарифмическая линейка Круг и т.п.
Механические и механико-электрические вычислительные машины (XIX в. – середина XX в.)	Учет (учет товаров, перепись населения, наука) Шифрование сообщений	Ч. Бэббидж А. Лавлейс Г. Голлерит И. Чебышев	Машина Голлерита («Голлерит») Арифмометры, в том числе с электроприводом Немецкая шифромашина «Энигма»
Первые ЭВМ (начиная с 40-х гг. XX в.)	Перебор значительного количества вариантов при относительно небольшом объеме вводимых данных	Дж. фон Нейман (теоретические основы) Э. Кодт (реляционные базы данных, 60-е гг.)	ENIAC (США, расчеты для ядерной программы) Мейнфреймы (разные модификации) ЕС ЭВМ (70-е гг.)
Первые управляющие вычислительные машины (начиная с 50-х гг. XX в.)	Управление параметрами функционирования простейших технических объектов Управление параметрами функционирования больших технических систем, контроль за значениями параметров	Н. Винер (теоретические основы) К. Шеннон (инженерная реализация) Д.А. Поспелов	Встроенные ЭВМ, иногда аналоговые – первые бортовые РЛС, системы космических ракет (СССР) и др. Релейные линии связи

## Окончание таблицы 5

1	2	3	4
Управляющие вычислительные комплексы (70-е гг. XX в.)	Автоматическая диагностика неисправностей в больших технических системах Управление функционированием больших технических систем, выбор не только параметров, но и режимов	Р. Белман Л.С. Понтрягин М. Мински	Измерительно-вычислительные комплексы на базе СМ ЭВМ (PDP-11) Минатом-пром, Минавиапром (СССР) Системы автоматизированного проектирования и гибкого производства, включая микро-ЭВМ, «безлюдные заводы»
Персональные компьютеры, а также большие ЭВМ (70-е гг.; широкое применение с конца 80-х гг. XX в.)	Сложный анализ многомерных данных, поиск причин произошедшего события, объяснение этих причин с учетом их последовательности и момента начала Поиск путей выхода из сложившейся ситуации: генерация альтернатив, оценка альтернатив и выбор наилучшего решения	Аристотель Дж. Буль Л. Заде М. Мински С. Пейперт Р. Кини Х. Райфа Р. Ковальски и др.	Микро-ЭВМ PDP-11, IBM/370, новые модификации СМ ЭВМ Серия ПК класса IBM 286, 386 и т.д.

Нас повсюду окружают интеллектуальные системы: живые организмы, поведение которых основано на опыте предшествующей деятельности и выбора знаний управления действиями в зависимости от условий функционирования, окружающей среды, т.е. их адаптации к разнообразным обстоятельствам. Сегодня создаются различные технические системы (роботы), которые могут вести себя подобно человеку, решать задачи, управлять сложными процессами. Однако при изменении среды функционирования, внешних факторов и пр. система даст сбой.

Исходя из этого, под *искусственным интеллектом* (artificial intelligence) мы будем понимать свойство автоматических систем брать на себя отдельные функции интеллекта человека, например, выбирать и принимать оптимальные решения на основе ранее полученного опыта (знаний) и рационального анализа внешних воздействий (способов действий).

Термин *знание* не является новым – он давно и широко используется в образовании, науке и в других сферах человеческой деятельности. Вместе с тем, когда речь идет о системах управления знаниями, этот общеизвестный термин нуждается во всестороннем рассмотрении. От его понимания и толкования зависят как цель создания такого рода систем, так и, в определенной мере, их функциональная структура и состав.

Так, по мнению Д.В. Смолина [56], знания представляют собой результат применения к исходным данным (декларативной информации) некоторых методов обработки,

подключения внешних процедур (присоединенных процедур, процедурной информации, правил вывода).

Рассмотрим такой пример. Человек снимает показания термометра. Что он видит? Данные? Информацию? Знания? С точки зрения визуализации – это картинка с изображенными на ней шкалой и соответствующей отметкой температуры. По нервным окончаниям в мозг поступает электрический сигнал, в результате чего формируется изображение (картинка). Далее, на основе умственных усилий, человек пытается определить показания термометра на основе используемой шкалы. Получив показания термометра (декларативную информацию), человек решает вопрос об их дальнейшем применении.

Другими словами, знания не лежат на поверхности, они возникают в результате сопоставления информации (ее отдельных частей, единиц), нахождения и разрешения противоречий между ними. Понятия «данные», «информация», «знания» можно представить графически (см. рис. 36).

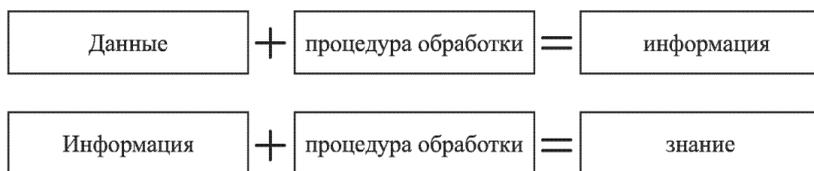


Рис. 36. Вывод на знаниях

Р. Аккофом [4] была предложена следующая иерархия понятий (см. рис. 37).

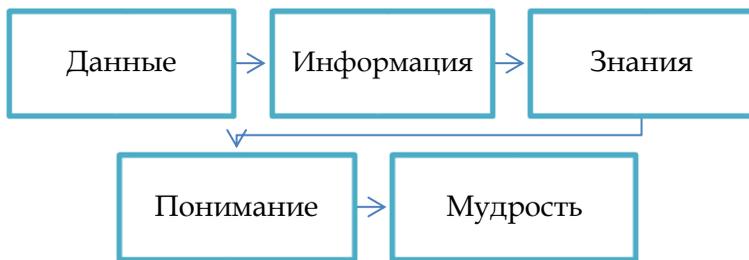


Рис. 37. Иерархия понятий

*Данные*, по Р. Аккофу, – это символы, представляющие свойства объектов, событий и их окружения, т.е. результаты наблюдений. Другими словами, данные представляют собой набор объективных фактов об объектах, событиях, явлениях, процессах, это все то, что регистрируется, описывается и воспринимается человеком. Данные могут быть цифровыми (факты, результаты измерений), графическими, аудио, видео и т.п. Они могут описываться на различных языках (символьном, математическом, графическом и т.п.).

Информация извлекается из данных с помощью логических рассуждений на основе анализа; состоит из описаний, ответов на вопросы, которые начинаются со слов: кто, что, где, когда, сколько [4]. Более подробно понятие «информации» будет рассмотрено нами в следующем параграфе.

Знание Р. Аккоф причислил к «ноу-хау». Именно знание позволяет преобразовать информацию в инструкцию. Приобрести знание можно двумя способами: либо передачей его от тех, кто им обладает, либо извлечением его из собственного практического опыта. «В любом случае приобретение знания есть обучение» [4].

*Понимание* – это осознание закономерностей, содержащихся в разрозненных знаниях, позволяющее ответить на вопрос: «Почему?»

Вынесение диагноза и предписаний есть понимание. Мудрость – взвешенное, оцененное понимание закономерностей с точки зрения прошлого и будущего.

Перечислим наиболее важные свойства знаний:

- Знания внутренне интерпретируемы. Это однозначное определение (сопоставление) с известным образцом (единицей информации).
- Знания имеют структурированную основу, т.е. между рассматриваемыми единицами информации должны быть установлены отношения, например, род, вид, класс и пр.
- Знания действуют в некотором пространстве, помогая формализовать рассматриваемую предметную область, абстрагируясь от несущественных признаков, действуя в определенной терминологии.

## **2.2. Понятие информации**

Информация играет важную роль в формировании современного общества. С рождения человек получает, обрабатывает, хранит, передает информацию поколениям. Говоря научным языком, по современным представлениям именно информация является одной из исходных данных мироздания.

Теория информации применима не только к науке, но и может быть востребована на производстве. «Современное

производство стало очень сложным; в связи с этим на первый план в нем выдвигаются вопросы организации и управления, основанные на процессах передачи и преобразования информации» [56, с. 15].

Понятие информации находится на стыке многих наук: кибернетики, философии, информатики и математики. До некоторых пор существовало мнение, что нет точного определения понятия «информация». Однако в каждом направлении, определенной области знания, в каждой дисциплине понятие «информация» трактуется по-своему и имеет собственный смысл и значение, что выражается в других понятиях: данные, факты, знания, аксиомы. Известно, что в 20-30-х годах прошлого века понятие информации в нашей стране было напрямую связано с теорией журналистики и было созвучно с описанием фактов.

В словаре иностранных слов дается следующее определение информации: «Информация [лат.; информировать] – 1) сведения, передаваемые людьми устным, письменным или техническим способом; средства массовой информации (СМИ) – печатные средства, радио, телевидение; 2) сведения, являющиеся объектом хранения, переработки и передачи» [55, с. 192].

С.И. Ожегов понятие информации трактует, с одной стороны, как сведения об окружающем мире и протекающих в нем процессах, воспринимаемых человеком или специальным устройством, и с другой – как сведения о положении дел, о состоянии чего-нибудь, понимая под сведениями некоторые познания в какой-либо области, известия, сообщения, знания, представление о чем-либо [40].

Из этого сделаем вывод, что вышеуказанные определения информации приводятся с интуитивной точки зрения, т.к. рассматриваемое понятие выражается через какие-то сведения. Академик Н.Н. Моисеев утверждал, что универсального определения понятия информации не существует ввиду его широты и многогранности [36]. Другими словами, понятие информации необходимо рассматривать с математической точки зрения, точно определяя все понятия, которые входят в определение, для решения теоретических и практических задач современной науки и техники.

Исходя из различных трактовок понятия информации, можно увидеть, что информация всегда проявляется в виде сигналов материально-энергетической формы.

Отметим вначале, что до 50-х годов прошлого века кибернетика в нашей стране вообще была запрещена из-за наличия марксистско-ленинской диалектики и была представлена как лженаука. С точки зрения математики понятие информации базируется на теории вероятностей и математической статистике, а также прикладной алгебре. В науке «Информатика» существует отдельная теория информации, которая получила развитие в трудах советских и зарубежных ученых: А.И. Берга Л. Бриллюэна, Н. Винера, Ф.М. Вудворта, Д. Габора, В.М. Глушкова, С. Гольдмана, Р.Л. Добрушина, А.Н. Железнова, А.Н. Колмогорова, В.А. Котельникова, В. Макмиллана, М.С. Пинскера, В.И. Сифорова, А. Файнштейна, Р.М. Фано, Л.М. Финка, Р. Хартли, А.Я. Хинчина, К. Шеннона и др.

Исторически возникновение теории информации связывают с публикацией американских ученых К. Шеннона

и У. Уивера «Математическая теория связи» (1948 г.), в которой рассматривались вероятностные методы для определения количества передаваемой информации [58]. Согласно данной теории К. Шеннон и У. Уивер попытались описать знаковую структуру информации, но этого было недостаточно, чтобы измерить заложенный в ней смысл (в сообщении, сведениях).

Теория информации включает также и работы Р. Хартли (количественная мера информации, 1928 г.), В.А. Котельникова (теорема о возможности представления непрерывной функции совокупностью ее значений в отдельных точках отсчета, 1933 г.; оптимальные методы приема сигналов на фоне помех, 1946 г.), А.Н. Колмогорова (статистическая теория колебаний как математическая основа теории информации, 1941 г.), У.Р. Эшби (книга «Введение в кибернетику», 1948 г.) и многих других отечественных ученых, внесших посильный вклад в развитие науки информации.

Н. Винер демонстрировал информацию как обозначение содержания, которое можно получить из внешнего мира в процессе приспособления человека к нему и приспособления к нему наших органов чувств [10]. Тем самым Н. Винер предложил кибернетику представлять как науку об управлении в живых организмах и технических системах. Вследствие этого в понятие информации стали включать сведения для принятия решения, обеспечивающих достижение цели управления, акцентируя внимание на новизне и полезности этих сведений.

Информация, по мнению В.В. Лидовского, это «нематериальная сущность, при помощи которой с любой

точностью можно описывать реальные (нематериальные), виртуальные (возможные) и понятийные сущности» [29, с. 10].

По мнению В.М. Глушкова, информация (в самом общем ее понимании) – это есть мера неоднородности распределения материи и энергии в пространстве и во времени, мера изменений, которыми сопровождаются все протекающие в мире процессы [14]. Листы книги, человеческая речь, солнечный свет, складки горного хребта, шум водопада, шелест листвы и многое другое – это все окружающая нас информация.

В.Г. Матвейкин, Б.С. Дмитриевский, К.А. Садов рассматривают понятие информации в следующем аспекте: «Информация – это сведения об окружающем мире (объектах, явлениях, событиях, процессах и т.д.), которые уменьшают имеющуюся степень неопределенности, полноты знаний, отчужденные от их создателя и ставшие сообщениями (выраженными на определенном языке в виде знаков, в том числе и записанными на материальном носителе), которые можно воспроизводить путем передачи устным, письменным или другим способом (с помощью устных сигналов, технических средств, вычислительных средств и т.д.)» [34, с. 7].

А.А. Боканов связывает понятие информации с отражением на носителе информации (памяти). По его мнению, содержанием информации являются сведения о некотором объекте (материальном, идеальном) [8].

В Федеральном законе «Об информации, информационных технологиях и о защите информации» под информацией понимаются сведения (сообщения, данные) независимо от формы их представления [59].

Итак, понятие информации выражается через другие понятия; используется в различных науках: физике, химии, информатике, математике, педагогике, психологии, экономике и пр. Информация будет приносить пользу человечеству только тогда, когда она будет воплощена в технику и технологию, знания и опыт человека, формы общения, социальных отношений. Информация будет являться сообщением только тогда, когда она будет выражена на определенном языке с помощью определенных символов в символично-знаковой системе.

В истории становления понятия информации, науки информации известны различные философские концепции. Первая утверждала, что информация являлась свойством всего сущего, всех материальных объектов мира (атрибутивная информация). Другая философская концепция изучала живые организмы и человеческое общество на основе функционирования сложных самоорганизующихся систем (функциональная теория). Третья говорила об антропоцентризме информации, когда информация существует лишь в человеческом сознании, в человеческом восприятии [54]. Р.Ф. Абдеев первые два подхода противопоставлял друг другу [3].

Попытки философского анализа шенноновской теории в работах советских философов И.А. Акчурина и И.И. Гришкина привели к выводу, что «в самом общем случае понятие информации связано с философскими категориями возможности и действительности... Всюду, где имеют место различные возможности, из которых реализуется, переходит в действительность, приобретает

бытие только одна, имеет смысл говорить об информации, которую несет с собой эта реализовавшаяся возможность» [58, с. 10].

Существуют также подходы, которые причисляют информацию к другим теориям: структурной, статистической, семантической. Если информация рассматривает структуру построения отдельных информационных сообщений, то имеет место быть структурная теория информации, в которой минимальной единицей количества информации является квант. Оценка информации с помощью энтропии, т.е. неопределенности, присуща статистической теории информации. Изучение смысловых характеристик информации (ценности, содержательности, полезности) рассматривается в семантической теории информации.

Информация имеет свои свойства [25]. Информацию будут характеризовать на основе полноты, актуальности, репрезентативности, адекватности, доступности и достоверности представленных сведений (см. рис. 38).



Рис. 38. Свойства информации

*Актуальность информации* зависит от интервала времени, прошедшего с момента появления данной информации, быстроты изменения ситуации и определяется степенью важности представленных (или получаемых сведений) для человека (или общества) посредством описания или использования их в конкретной ситуации при решении поставленной проблемы.

Из этого свойства можно увидеть ее *полезность*, т.е. получение информации в текущий момент, предсказание о будущих событиях (например, ураганы, землетрясения или другие стихийные бедствия).

*Доступность информации* – это преобразование или представление информации в понятной форме в зависимости от ее получателя.

*Достоверность информации* – это свойство информации, при котором она определяется отражением состояния реально существующего объекта, процесса или явления. Ясно, что недостоверная информация может привести к неверному пониманию ситуации и принятию неправильного решения.

*Полнота информации* характеризуется наличием минимального и достаточного набора данных для принятия правильного (а иногда и рационального) решения.

Соответствие образа, создаваемого с помощью полученной информации (информационной модели), реальному объекту, процессу или явлению являются *характеристикой адекватности информации*, степень которой влияет на правильность решений, принятых человеком.

*Репрезентативность* – это свойство информации, которое связано с правильностью ее отбора (т.е. наличие

похожей информации, полученных из разных источников) и формирования для адекватного отражения свойств объекта.

Принято также информацию определять через ее объективность, т.е. через независимость ее от человеческого фактора.

### **2.3. Основные направления в области искусственного интеллекта**

Интеллектуальные информационные системы проникают во все сферы нашей жизни, поэтому трудно провести строгую классификацию направлений, по которым ведутся активные и многочисленные исследования в области искусственного интеллекта. Рассмотрим кратко некоторые из них.

*Разработка интеллектуальных информационных систем или систем, основанных на знаниях.* Это одно из главных направлений искусственного интеллекта. Основной целью построения таких систем являются выявление, исследование и применение знаний высококвалифицированных экспертов для решения сложных задач, возникающих на практике. При построении систем, основанных на знаниях (СОЗ), используются знания, накопленные экспертами в виде конкретных правил решения тех или иных задач. Это направление преследует цель имитации человеческого искусства анализа неструктурированных и слабоструктурированных проблем. В данной области исследований осуществляется

разработка моделей представления, извлечения и структурирования знаний, а также изучаются проблемы создания баз знаний, образующих ядро СОЗ. Частным случаем СОЗ являются экспертные системы.

*Разработка естественно-языковых интерфейсов и машинный перевод.* Проблемы компьютерной лингвистики и машинного перевода разрабатываются в теории искусственного интеллекта с 1950-х гг. Системы машинного перевода с одного естественного языка на другой обеспечивают быстроту и систематичность доступа к информации, оперативность и единообразие перевода больших потоков, как правило, научно-технических текстов. Системы машинного перевода строятся как интеллектуальные системы, поскольку в их основе лежат базы знаний в определенной предметной области и сложные модели, обеспечивающие дополнительную трансляцию «исходный язык оригинала – язык смысла – язык перевода». Они базируются на структурно-логическом подходе, включающем последовательный анализ и синтез естественно-языковых сообщений. Кроме того, в них осуществляется ассоциативный поиск аналогичных фрагментов текста и их переводов в специальных базах данных. Данное направление охватывает также исследования методов и разработку систем, обеспечивающих реализацию процесса общения человека с компьютером на естественном языке (так называемые системы ЕЯ-общения).

*Генерация и распознавание речи.* Системы речевого общения создаются в целях повышения скорости ввода информации в ЭВМ, разгрузки зрения и рук, а также для реализации речевого общения на значительном расстоянии.

В таких системах под текстом понимают фонемный текст (как слышится).

*Обработка визуальной информации.* В этом научном направлении решаются задачи обработки, анализа и синтеза изображений. Задача обработки изображений связана с трансформированием графических образов, результатом которого являются новые изображения. В задаче анализа исходные изображения преобразуются в данные другого типа, например, в текстовые описания. При синтезе изображений на вход системы поступает алгоритм построения изображения, а выходными данными являются графические объекты (системы машинной графики).

*Обучение и самообучение.* Эта актуальная область ИИ включает модели, методы и алгоритмы, ориентированные на автоматическое накопление и формирование знаний с использованием процедур анализа и обобщения данных. К данному направлению относятся не так давно появившиеся системы добычи данных (Data-mining) и системы поиска закономерностей в компьютерных базах данных (Knowledge Discovery).

*Распознавание образов.* Это одно из самых ранних направлений искусственного интеллекта, в котором распознавание объектов осуществляется на основании применения специального математического аппарата, обеспечивающего отнесение объектов к классам, а классы описываются совокупностями определенных значений признаков.

*Игры и машинное творчество.* Машинное творчество охватывает сочинение компьютерной музыки, стихов, интеллектуальные системы для изобретения новых объектов.

Создание интеллектуальных компьютерных игр является одним из самых развитых коммерческих направлений в сфере разработки программного обеспечения. Кроме того, компьютерные игры предоставляют мощный арсенал разнообразных средств, используемых для обучения.

*Программное обеспечение систем искусственного интеллекта.* Инструментальные средства для разработки интеллектуальных систем включают специальные языки программирования, ориентированные на обработку символьной информации (LISP, SMALLTALK, РЕФАЛ), языки логического программирования (PROLOG), языки представления знаний (OPS 5, KRL, FRL), интегрированные программные среды, содержащие арсенал инструментальных средств для создания систем ИИ (KE, ARTS, GURU, G2), а также оболочки экспертных систем (BUILD, EMYCIN, EXSYS Professional, ЭКСПЕРТ), которые позволяют создавать прикладные экспертные системы, не прибегая к программированию.

*Новые архитектуры компьютеров.* Это направление связано с созданием компьютеров не фон-неймановской архитектуры, ориентированных на обработку символьной информации. Известны удачные промышленные решения параллельных и векторных компьютеров, однако в настоящее время они имеют весьма высокую стоимость, а также недостаточную совместимость с существующими вычислительными средствами.

*Интеллектуальные роботы.* Создание интеллектуальных роботов составляет конечную цель робототехники. В настоящее время в основном используются программируемые манипуляторы с жесткой схемой управления,

названные роботами первого поколения. Несмотря на очевидные успехи отдельных разработок, эра интеллектуальных автономных роботов пока не наступила. Основными сдерживающими факторами в разработке автономных роботов являются нерешенные проблемы в области интерпретации знаний, машинного зрения, адекватного хранения и обработки трехмерной визуальной информации.

Следует отметить, что в области искусственного интеллекта существует и другая классификация направлений исследований, представленная в обобщенном виде [41]. Согласно данной классификации можно выделить два главных направления – прагматическое и бионическое.

Прагматическое направление основано на предположении о том, что мыслительная деятельность человека представляет собой «черный ящик». «...Если результат функционирования искусственной системы в некотором смысле совпадает с результатом деятельности эксперта, то такую систему можно признать интеллектуальной независимо от способов получения этого результата» [41]. При таком подходе не ставится вопрос об адекватности используемых в компьютере структур и методов по отношению к тем структурам и методам, которыми пользуется в аналогичных ситуациях человек, а рассматривается лишь конечный результат решения конкретных задач.

С точки зрения конечного результата в прагматическом направлении можно выделить три целевые области:

- создание инструментария. Инструментарий – языки для систем искусственного интеллекта; дедуктивные и индуктивные методы автоматического синтеза программ;

лингвистические процессоры; системы анализа и синтеза речи; базы знаний; оболочки, прототипы систем; системы когнитивной графики;

- разработка методов представления и обработки знаний является одной из основ современного периода развития искусственного интеллекта;

- интеллектуальное программирование разбивается на несколько групп. К ним относят игровые программы, естественно-языковые программы (системы машинного перевода, автоматического реферирования, генерации текстов), распознающие программы, программы создания произведений живописи и графики.

Общим для перечисленных программ является широкое использование поисковых процедур и методов решения переборных задач, связанных с поиском и просмотром большого числа вариантов. Эти методы применяются при машинном решении игровых задач, в задачах выбора решений, при планировании целесообразной деятельности в интеллектуальных системах.

Бионическое направление исследований в области искусственного интеллекта основано на предположении о том, что если в искусственной системе воспроизвести структуры и процессы человеческого мозга, то и результаты решения задач такой системой будут подобны результатам, получаемым человеком. В этом направлении исследований выделяются:

- нейробионический подход. В его основе лежат системы элементов, способные подобно нейронам головного

мозга воспроизводить некоторые интеллектуальные функции. Прикладные системы, разработанные на основе этого подхода, называются нейронными сетями;

- структурно-эвристический подход. В его основе лежат знания о наблюдаемом поведении объекта или группы объектов и соображения о тех структурах, которые могли бы обеспечить реализацию наблюдаемых форм поведения. Примером подобных систем служат мультиагентные системы;

- гомеостатический подход. В этом случае решаемая задача формулируется в терминах эволюционирующей популяции организмов – совокупности противоборствующих и сотрудничающих подсистем, в результате функционирования которых обеспечивается нужное равновесие (устойчивость) всей системы в условиях постоянно изменяющихся воздействий среды. Такого рода подход реализован в прикладных системах на основе генетических алгоритмов.

## **2.4. Модели представления знаний**

В теории искусственного интеллекта существует множество моделей представления знаний для различных предметных областей.

Д.И. Кашириным и И.Ю. Кашириным были описаны известные модели представления знаний с выявлением соответствующих классов задач, в которых эти модели наиболее эффективны:

- семантические сети;
- концептуальные сети;
- ситуационное управление;

- логическое представление знаний;
- универсальный семантический код;
- Ф-язык;
- фреймы для представления знаний;
- растущие пирамидальные сети;
- признаковые структуры;
- онтологии для описания семантики [23].

Большинство же моделей может быть сведено к следующим классам:

- продукционные модели;
- семантические сети;
- фреймы;
- формальные логические модели;
- знания, основанные на нечетких множествах.

Рассмотрим подробнее каждый класс представления знаний.

#### *2.4.1. Продукционная модель представления знаний*

Информационная модель в интеллектуальных информационных системах представлена совокупностью двух разновидностей знаний – декларативных и процедурных.

*Декларативными* принято называть знания о свойствах сущностей предметной области и об отношениях между ними, а *процедурными* – знания о допустимых правилах управления такой информацией. Этот тип моделей представления знаний получил название в связи с тем, что оно основано на конструкциях, позволяющих использовать сочетания уже известных фактов (образующих ситуации или условия) для получения новых знаний или ситуаций.

Правила, на основании которых такие преобразования в конкретной предметной области имеют смысл, называются «производящими правилами» или «продукциями». Продукция представляет собой обусловленную опытом и (или знаниями) абстрактных истин причинно-следственную конструкцию, состоящую из двух частей: антецедента и консеквента. Антецедент играет роль посылки. Это образное предложение, по которому идет поиск в базе знаний. Консеквент – вывод, обусловленный истинностью посылки, т.е. действие на основании факта.

В продукционных моделях представления знаний правила выражаются формулой:

ЕСЛИ условие (антецедент)

ТО действие (консеквент)

Для образования сложных предложений допустимо использование логических операций – И, ИЛИ, НЕ.

Запишем обобщенное продукционное правило:

$$i : S; C; A \rightarrow B; P,$$

где  $i$  – номер продукции;

$S$  – описание класса ситуаций, в котором эта структура может использоваться;

$C$  – условие, при котором данная продукция активизируется;

$A \rightarrow B$  – ядро продукции (например, «ЕСЛИ  $A_1, A_2, \dots, A_n$ , ТО  $B$ »);

$P$  – постусловие продукционного правила, определяющее действия, которые необходимо произвести после выполнения  $B$ .

В зависимости от количества условий и действий в соответствующих перечнях различают следующие типы правил:

- простое – одно условие и одно действие,
- составное – много условий и действий,
- фиксирующее – много условий и одно действие,
- разветвляющееся – одно условие и много действий.

Приведем примеры представления знаний в виде продукционной модели:

ЕСЛИ («испытываешь жажду») ТО («выпей воды или съешь фрукт»)

ЕСЛИ («день солнечный»  
И «шторы на окнах подняты»)  
ТО («следует выключить искусственное освещение»)

ЕСЛИ («ученик ответил неверно»  
ИЛИ воспользовался подсказкой») ТО («оценка его ответа должна быть неудовлетворительной»)

ЕСЛИ («спрос на товар НЕ растет») ТО («следует прекратить торги»)

В теории систем с продукционной моделью представления знаний имеются две схемы получения результирующего вывода – прямой и обратный [22]. Прямой

логический вывод – это организация перехода «от фактов к заключению» или «от данных к поиску цели». Обратный логический вывод – поиск в базе знаний такого сочетания фактов, которые могут опровергнуть или подтвердить истинность некоторой гипотезы – «от гипотезы – к фактам» [18].

Сферами применения продукционных моделей чаще всего являются промышленные экспертные системы, т.к. они обеспечивают простоту смысловой интерпретации, модульность, легкость корректировки и логического вывода.

Достоинство продукционных моделей заключается в «прозрачности» разрабатываемой на их основе экспертной системы, т.е. ее способности к объяснению принятых решений и полученных результатов. Кроме того, с помощью продукций может быть записана большая часть человеческих знаний ввиду отсутствия логики рассуждений. Исходя из этого, человеку просто создать и понять отдельные правила, записанные на языке продукционной модели. «Разбиение системы продукций на сферы (декомпозиция) позволяет эффективно использовать ресурсы и сократить время поиска решения» [5].

Основным недостатком продукционных моделей являются трудности при добавлении правил, зависящих от уже имеющихся в базе знаний, а также при отладке программ. В этом случае необходимо двигаться сверху-вниз для изменения такой системы при внесении изменений.

Для разработки продукционных моделей представления знаний, реализуемых на ЭВМ, используются алгоритмические языки PROLOG, CLISP, ESTA, OPS5, EXSYS, GURU, Кара и др. [61]. Пример работы экспертной системы с использованием программы ESTA приведен на рисунке 39.

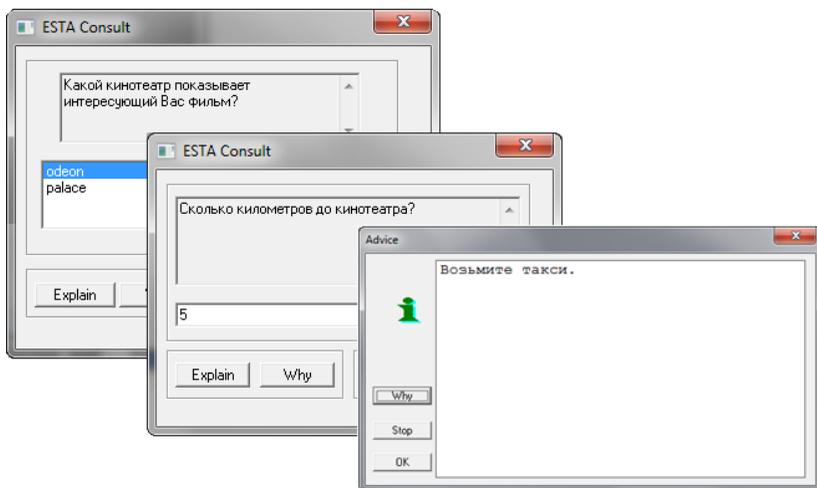


Рис. 39. Окна режима консультации с экспертной системой в программе ESTA

#### 2.4.2. Семантические сети

*Семантическая сеть* – это метод представления знаний, который позволяет описывать объекты, явления и понятия предметной области с помощью сетевых структур, основанных на применении теории графов.

В основе данного метода лежит идея о том, что любые знания можно представить в виде совокупности объектов (понятий) и связей (отношений) между ними. Иначе говоря, семантическая сеть – ориентированный граф, в котором вершины определяют некоторые понятия из рассматриваемой предметной области, а отношения между ними являются дугами графа. Каждое отношение является, по сути, предикатом, простым или составным, описывающим имеет ли место быть данное отношение или нет.

Рассмотрим пример семантической сети, представленной на рис. 40.

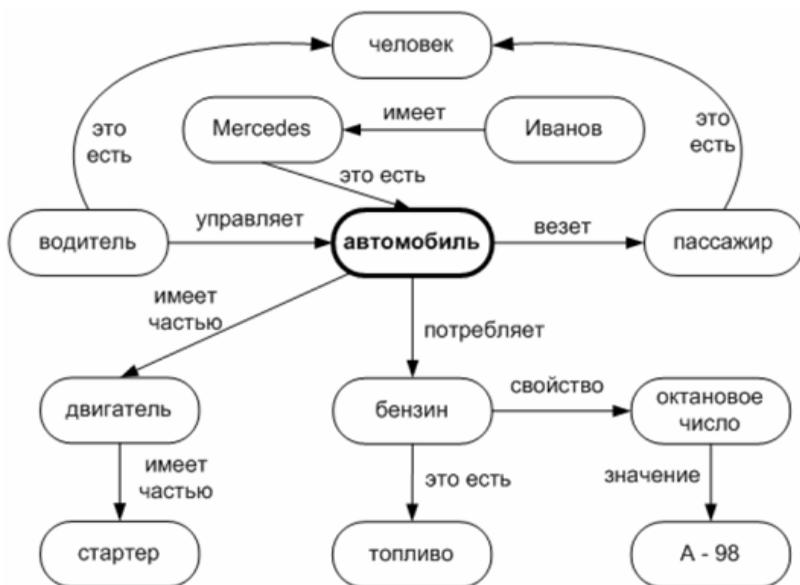


Рис. 40. Пример семантической сети «Автомобиль»

Данный пример хорошо иллюстрирует отличие модели знаний от базы данных. Семантическая сеть наглядно отражает взаимосвязь входящих в нее объектов.

Например, если в базу данных об автомобилях добавить новую запись «БМВ – это автомобиль», то мы будем знать только этот факт. Однако если добавить этот факт в данную семантическую сеть, то можно определить, что БМВ – это автомобиль, который имеет владельца, определенные технические характеристики, заданные в рассматриваемой сети.

В виде семантической сети можно представить различные системы:

- «Школьный урок». Система состоит из следующих элементов: Ученик, Учитель, Учебник, Тетрадь, Классный журнал, Классная доска, Мел, Рабочее место ученика, Рабочее место учителя, Классная комната.

- Круговорот воды в природе.

- Система высших органов власти Российской Федерации.

С помощью семантической сети можно легко построить организационную диаграмму.

Основу семантической сети составляют события, атрибуты, комплексы признаков и процедуры.

*События* – это суждения, факты, результаты наблюдений, рекомендации. Могут представляться словосочетаниями и числами. Группируются тематически или функционально в разделы. Делятся на характеризующие (события-признаки, например, «идет дождь» для события «дождливая погода»).

*Атрибут* – это характеризующее событие, имеющее несколько значений, например, «погода» – атрибут «времени года».

*Процедура* – это специфический компонент сети, выполняющий преобразование информации. Она позволяет вычислять значения одних атрибутов на основании других, оперируя как с числами, так и с символами.

Рассмотрим следующие примеры описания отношений:

1. У Павла есть отец по имени Алексей (отношение 1:1).
2. Для Павла найдется отец из множества мужчин (отношение 1:M).

3. Найдется человек, для которого Алексей – отец (отношение M:1).

4. У каждого человека есть отец из множества мужчин (отношение M:M).

Математическая запись предложений может быть представлена следующим образом:

- $\exists$  Павел &  $\exists$  Алексей: отец (Алексей, Павел);
- $\exists$  Павел  $\rightarrow \exists x \in$  мужчины: отец (x, Павел);
- $\exists$  Алексей  $\rightarrow \exists y \in$  мужчины: отец (Алексей, y);
- $\forall y \in$  люди  $\rightarrow \exists x \in$  мужчины: отец (x, y).

Семантические сети классифицируют по арности (т.е. количеству связей между объектами сети – бинарные и N-арные), количеству типов отношений (однородные и неоднородные). Отметим, что количество типов отношений в семантической сети определяется ее создателем, исходя из конкретных целей. В реальном мире их число стремится к бесконечности.

В зависимости от типов связей различают классифицирующие, функциональные сети и сценарии. Классифицирующие сети – сети, в которых используются отношения структуризации, они позволяют вводить в базы знаний различные иерархические отношения между элементами множества объектов. Полученная такая сеть будет иметь собственную внутреннюю структуру, что в дальнейшем позволит разделить сеть на подсети (пространства) и установить отношения не только между вершинами, но и между пространствами.

Функциональные сети представляют собой вычислительные модели, характеризующиеся наличием функциональных отношений, они позволяют описывать процедуры

вычислений одних информационных единиц через другие. Скорость работы с базой знаний будет зависеть от того, насколько эффективно сделаны программы обработки нужных отношений.

Сценарии основаны на использовании причинно-следственных или устанавливающих связей одних явлений или фактов с другими, или отношения типа «средство – результат», «орудие – действие» и т.д.

Отметим основные достоинства сетевой модели представления знаний:

- выразительные возможности;
- наглядность системы знаний, представленной графически;
- близость структуры сети, представляющей систему знаний, семантической структуре фраз на естественном языке;
- соответствие современным представлениям об организации долговременной памяти человека.

Однако сетевая модель представления знаний имеет ряд недостатков:

- сетевая модель не дает, точнее не содержит, ясного представления о структуре предметной области, которая ей соответствует, поэтому формирование и модификация такой модели затруднительны;
- сетевые модели представляют собой пассивные структуры, для обработки которых необходим специальный аппарат формального вывода и планирования – проблема поиска решения в базе знаний типа семантической сети сводится к задаче поиска фрагмента сети, соответствующего некоторой подсети поставленной задачи;
- сложность поиска вывода на семантической сети.

Современное применение семантических сетей является обширным – это и семантическая паутина (организация гиперссылок на web-странице), и машинный перевод языков путем трансляция исходного текста в так называемую промежуточную форму (т.е. образование семантической сети) и последующего ее перевода на нужный язык. Сегодня семантические сети нашли свое применение и в образовании – с помощью них можно построить интеллектуальные (или ментальные) карты (см. рис. 41). «Интеллект-карта – это графическое выражение процесса мышления (иногда с элементами творчества), познания нового или совершенства усвоенного материала посредством эффективной структуризации, систематизации информации» [62, с. 5].

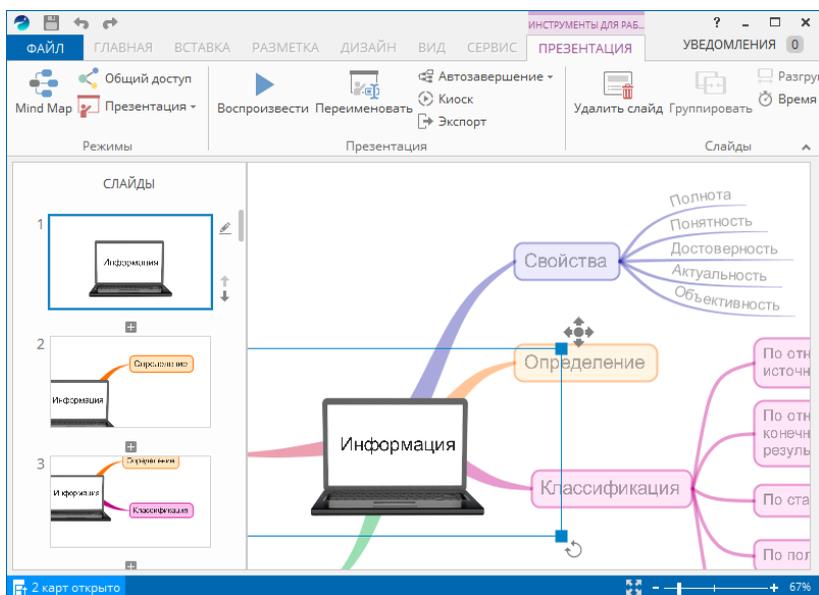


Рис. 41. Построение ментальной карты

### 2.4.3. Фреймовая модель представления знаний

Фреймовая модель представления знаний была предложена Марвином Ли Минским в 70-х годах XX века, с помощью которой можно обозначить структуру знаний для восприятия пространственных сцен.

*Фрейм* – это некоторая структура для представления знаний пространственных сцен, которая при ее заполнении соответствующими значениями превращается в описание конкретного факта, события или ситуации. М.А. Малязина представляет понятие фрейма как форму представления знаний – каркас, схема, план, рамка, сценарий, шаблон, трафарет, гештальт, прототип, парадигма (в языкознании), дисциплинарная матрица (в науковедении). «Фрейм обозначает обширный круг понятий, связанных со структурированием реальности. В частности, это процедурное знание, то есть последовательность действий, раскрывающих креативный, либо функциональный аспект» [33, с. 2]. «Фрейм представляет собой не одну конкретную ситуацию, а наиболее характерные, основные моменты ряда близких ситуаций (действий, рассуждений, зрительных образов, повествований и т.д.), принадлежащих одному классу» [35, с. 124]. «Фреймы не только описывают существеннейшие аспекты онтологии изучаемой реальности или даже всего бытия, но также снабжают науку определенным строительным материалом, каркасами и блоками для сооружения всего целого, т.е. теоретического знания, или здания науки...» [51, с. 38].

Каждый фрейм можно рассматривать как семантическую сеть, состоящую из выделенных вершин и связей. Фреймовая модель основана на принципе фрагментации

знаний. Основу данной модели составляет слот, который состоит из имени некоторого признака, значений этого признака и связи с другими слотами.

Рассмотрим следующий пример описания ситуации: «Студент Иванов получил книгу А.Я. Архангельского «100 компонентов Delphi» в библиотеке ТГПУ им. Л.Н. Толстого в г. Туле»:

ПОЛУЧЕНИЕ:

ОБЪЕКТ (КНИГА: (Автор, А.Я. Архангельский), (Название, 100 компонентов Delphi));

АГЕНТ (СТУДЕНТ: (Фамилия, Иванов));

МЕСТО: (БИБЛИОТЕКА: (Название, ТГПУ), (Расположение, Тула)).

В этом примере ОБЪЕКТ, АГЕНТ и МЕСТО – это роли, которые играют слоты КНИГА, СТУДЕНТ и БИБЛИОТЕКА в рамках фрейма ПОЛУЧЕНИЕ.

Общий вид описания фрейма выглядит следующим образом:

$$f = [(r,v), \dots, (r,v)],$$

где  $f$  – имя фрейма;  $r$  – имя слота;  $v$  – значение слота.

Допустимо также использование и другой формы описания фрейма:

(

Имя фрейма:

(имя 1-го слота: значение 1-го слота),

(имя 2-го слота: значение 2-го слота),

...

(имя n-го слота: значение n-го слота)

)

Значением слота может быть практически что угодно (числа или математические соотношения, тексты на естественном языке или программы, правила вывода или ссылки на другие слоты данного фрейма или других фреймов). В качестве значения слота может выступать набор слотов более низкого уровня, что позволяет во фреймовых представлениях реализовать «принцип матрешки» [38].

Фреймовую модель можно представить в виде таблицы, у которой в отличие от реляционной модели данных есть ряд особенностей:

- возможность смешанного заполнения слотов константами и переменными;
- возможность наличия пустых слотов;
- размещение в слотах указателей на другие фреймы для создания сети;
- размещение в слотах имен выполняемых процедур.

Пример задания фрейма в виде таблицы приведен на рисунке 42.

Имя фрейма			
имя слота	тип слота	значение слота	присоединенная процедура
...	...	...	...

Рис. 42. Задание фрейма в виде таблицы

В теории фреймов выделяют следующие виды моделей:

- фреймы-структуры, которые применяются для обозначения объектов и понятий (заем, залог, вексель);
- фреймы-роли для описания организационных диаграмм, должностных инструкций;

- фреймы-сценарии для описания (банкротство, празднование именин);

- фреймы-ситуации (авария, тревога) [60].

Перечислим основные параметры задания фреймов:

1. Имя фрейма. Служит для идентификации фрейма и должно быть уникальным. Фрейм представляет собой совокупность слотов и присоединенных процедур.

2. Имя слота. Представляет собой идентификатор атрибута сущности, который наделен определенной семантикой. Должно быть уникальным в пределах фрейма. В качестве имени слота может выступать произвольный текст (например, «Количество студентов в группе», «Возраст») или зарезервированное обозначение. Примеры имен системных слотов (зарезервированных обозначений): IS-A (это), PART-OF (является частью), RELATIONS (относиться к) и т.д. Системные слоты служат для редактирования баз знаний и управления выводом во фреймовой системе.

3. Указатели наследования. Определяют правила заполнения значений слота. Указатели наследования характерны для фреймовых систем иерархического типа, основанных на отношениях типа «абстрактное – конкретное». В конкретных системах указатели наследования могут быть организованы различными способами и иметь разные обозначения:

- по умолчанию (англ. default) от фрейма-образца;
- через наследование от родительского фрейма, указанного в слоте IS-A;
- по умолчанию или через наследование, но в случае необходимости значение слота может быть изменено (англ. override – заменить);

- значение слота для всех фреймов-экземпляров одного типа должно быть уникальным (англ. unique);

- значение слота для всех фреймов-экземпляров одного типа должно находиться в пределах, указанных для слота во фрейме-образце (англ. range – диапазон);

- по формуле, указанной в слоте;
- через демон;
- явно из диалога с пользователем и т.п.

4. Тип данных. Определяет тип значения слота. Наиболее употребляемые типы:

- frame – указатель на фрейм;
- real – вещественное число;
- integer – целое число;
- boolean – логический тип;
- text – фрагмент текста;
- list – список;
- table – таблица;
- expression – выражение;
- lisp – присоединенная процедура и т.д.

5. Значение слота. Должно соответствовать указанному типу данных и условию наследования.

6. Демон. Демоном называется присоединенная процедура, автоматически запускаемая при выполнении некоторого условия. Демон с условием IF-NEEDED (англ. needed – нужно, необходимо) запускается, если в момент обращения к слоту его значение не было установлено. Демон типа IF-ADDED (англ. added – добавлять) запускается при попытке создания фрейма-экземпляра. Демон IF-REMOVED (англ. removed – удалять) или IF-DELETED (англ. deleted –

удалять) запускается при попытке удаления значения слота. Возможны также другие типы демонов.

7. Присоединенная процедура. Совокупность процедур определяет поведение фрейма. Присоединенная процедура может запускаться автоматически (для демонов) или при обращении к ней из процедур этого же или других фреймов. Процедурные знания являются средством управления выводом во фреймовых системах [5; 6; 49].

Пример описания фрейма приведен на рисунке 43.

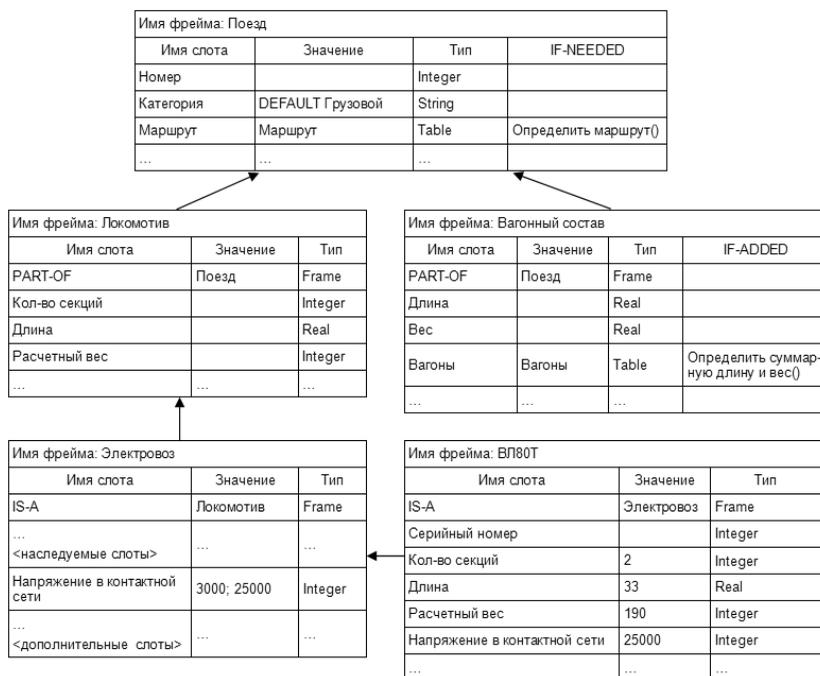


Рис. 43. Пример описания фрейма

Преимущество фреймовой модели перед семантическими сетями состоит в том, что, во-первых, сложные

объекты описываются (или представляются) в виде единой неделимой сущности, а не в виде однородной семантической структуры (подсети). Во-вторых, фреймовая модель позволяет естественным образом представлять стереотипные сущности (фреймы-образцы), наследование, полиморфизм и т.п. В-третьих, фрейм имеет развитую процедуру наследования значений слотов и предоставляет возможность организации процедур контроля ссылочной целостности данных.

По мнению Х. Уэно, недостатком фреймовой модели является отсутствие специального механизма управления выводом, в связи с чем разработчики должны реализовать данный механизм с помощью присоединенных процедур [49].

#### *2.4.4. Логическая модель представления знаний*

Логическая модель представления знаний предназначена для описания математического знания на основе законов математической логики. Основная идея при построении логических моделей знаний состоит в том, что вся информация, необходимая для решения прикладных задач, рассматривается как совокупность фактов и утверждений, которые представляются как формулы в некоторой логике, и знания отображаются совокупностью таких формул, а получение новых знаний сводится к реализации процедур логического вывода.

Общая формула логической модели представления знаний может быть задана следующим образом:

$$S = \langle A, F, Ax, R \rangle,$$

где  $A$  – счетное множество базовых символов (т.е. алфавит системы);  $F$  – множество, называемое формулами;  $Ax$  – выделенное подмножество априори истинных формул

(аксиом);  $R$  – конечное множество отношений между формулами, называемое правилами вывода.

В логических моделях знаний слова, описывающих сущности предметной области, называются *термами* (константы, переменные, функции), а слова, описывающие отношения сущностей – предикатами.

*Предикат* – логическая  $N$ -арная пропозициональная функция, определенная для предметной области и принимающая значения либо истинности, либо ложности.

*Пропозициональной* называется функция, которая ставит в соответствие объектам из области определения одно из истинностных значений («истина», «ложь»). Предикат принимает значения «истина» или «ложь» в зависимости от значений, входящих в него термов.

Рассмотрим следующий пример:

любит (Иван, Марья).

Здесь «любит» – предикат, «Иван», «Марья» – объекты, входящие в предикат.

Вывод в формальной логической системе является процедурой, которая из заданной группы выражений выводит отличное от заданных семантически правильное выражение. Если группа выражений, образующая посылку, является истинной, то должно гарантироваться, что применение правила вывода обеспечит получение истинного выражения в качестве заключения.

В теории логической модели представления знаний представлено два метода вывода: метод правил вывода (или метод дедуктивных правил) и метод резолюций. Рассмотрим подробнее каждый из данных методов.

Метод правил вывода (или метод дедуктивных правил) заключается в том, что в логике предикатов используется правило, которое из двух выражений  $A$  и  $A \rightarrow B$  выводит новое выражение  $B$ . «Если известно, что высказывание  $A$  влечет (имплицирует) высказывание  $B$ , а также известно, что высказывание  $A$  истинно, то, следовательно,  $B$  истинно» [11].

Пример: Даны утверждения:

- «Сократ – человек»;
- «Человек – это живое существо»;
- «Все живые существа смертны».

Требуется доказать утверждение «Сократ смертен».

Решение:

Шаг 1. Представим высказывания в предикатной форме:

Утверждение	Предикатная форма
«Человек – живое существо»	$\forall(X) \text{ (Человек } (X) \rightarrow \text{ Живое\_существо } (X))$
«Сократ – человек»	Человек (Сократ)
«Все живые существа смертны»	$\forall(Y) \text{ (Живое\_существо } (Y) \rightarrow \text{ Смертно } (Y))$

Шаг 2. На основе правила вывода и подстановки (Сократ/ $X$ ) в первом предикате получим утверждение: «Сократ – Живое существо»/

Шаг 3. На основе правила вывода и подстановки (Сократ/ $Y$ ) в третьем предикате получим утверждение: «Сократ смертен».

Другой метод – метод резолюций – предложен в 1930 г. в докторской диссертации Жака Эрбрана для доказательства теорем в формальных системах первого порядка [13]. Метод резолюций опирается на исчисление резольвент. Существует теорема, утверждающая, что вопрос о доказуемости произвольной формулы в исчислении предикатов сводится к вопросу о доказуемости пустого списка в исчислении резольвент. Поэтому доказательство того, что список формул в исчислении резольвент пуст, эквивалентно доказательству ложности формулы в исчислении предикатов.

Приведем решение методом резолюций для описанного ранее примера.

Решение:

Шаг 1. Преобразуем высказывания в дизъюнктивную форму:

Предикатная форма	Дизъюнктивная форма
$\forall(X) \text{ (Человек (X) } \rightarrow \text{ Живое\_существо (X))}$	$\neg(\text{Человек}(X) \vee \text{ Живое\_существо (X)})$
Человек (Сократ)	Человек (Сократ)
$\forall(Y) \text{ (Живое\_существо (Y) } \rightarrow \text{ Смертно (Y))}$	$\neg(\text{Живое\_существо (Y) } \vee \text{ Смертно (Y)})$

Шаг 2. Запишем отрицание целевого выражения (требуемого вывода), т.е. «Сократ бессмертен»:

$$\neg \text{Смертно (Сократ)}$$

Шаг 3. Составим конъюнкцию всех дизъюнктов, включив в нее отрицание целевого выражения:

$$\begin{aligned} & (\neg(\text{Человек}(X) \vee \text{ Живое\_существо}(X))) \wedge \\ & (\neg(\text{Живое\_существо}(Y) \vee \text{ Смертно}(Y))) \\ & \wedge (\text{Человек (Сократ)}) \wedge (\neg \text{Смертно(Сократ)}) \end{aligned}$$

Шаг 4. В цикле проведем операцию поиска резольвент над каждой парой дизъюнктов:

1 дизъюнкт	∨	1 дизъюнкт
$\neg(\text{Человек}(X) \vee \text{Живое\_существо}(X))$		$\neg(\text{Живое\_существо}(Y) \vee \text{Смертно}(Y))$
Резольвента: $\neg\text{Человек}(Y) \vee \text{Смертно}(Y)$		
$\neg\text{Человек}(Y) \vee \text{Смертно}(Y)$		Человек (Сократ)
Резольвента: Смертно (Сократ)		
Смертно (Сократ)		$\neg\text{Смертно}(Сократ)$
Пустой дизъюнкт		

Получение пустого дизъюнкта означает, что высказывание «Сократ бессмертен» ложно, значит истинно высказывание «Сократ смертен» [11].

Перечислим основные достоинства логической модели представления знаний:

- использование классического аппарата математической логики, методы которой достаточно хорошо изучены и формально обоснованы;
- существуют достаточно эффективные процедуры вывода, в том числе реализованные в языке логического программирования «Пролог»;
- в базах знаний можно хранить лишь множество аксиом, а все остальные знания получать из них по правилам вывода [30].

Однако логическим моделям представления знаний свойственны и недостатки: чрезмерный уровень формализации знаний; слабая наглядность, трудность прочтения логических формул и сложность их понимания; низкая

производительность систем искусственного интеллекта при обработке знаний и большая требуемая память; громоздкость при описании больших объемов знаний [31].

#### *2.4.5. Модель, основанная на нечетких знаниях*

Проблемы, которые возникают перед человеком в разных сферах познаний, считаются по своей природе слишком сложными и многосторонними при использовании для их решения точных, хорошо определенных моделей и алгоритмов.

«Многие понятия, вследствие человеческого мышления, приближенного характера умозаключений, лингвистического их описания являются нечеткими по своей природе и требуют для своего описания соответствующего аппарата, в частности, аппарата теории нечетких множеств, предложенные Л. Заде» [9].

Перечислим основные компоненты нечеткости знаний и дадим им краткую характеристику.

*Недетерминированность выводов.* Недетерминированность означает, что путь решения конкретной задачи в пространстве ее состояний заранее невозможно определить. Поэтому методом проб и ошибок выбирается некоторая цепочка логических заключений, согласующихся с имеющимися знаниями, а в случае, если она не приводит к успеху, организуется перебор с возвратом для поиска другой цепочки и т.д. Такой подход предполагает определение некоторого первоначального пути.

*Многозначность знаний.* При понимании естественного языка серьезными проблемами становятся многозначность смысла слов, их подчиненности, порядка слов в предложении

и т.п. Проблемы понимания смысла возникают в любой системе, взаимодействующей с пользователем на естественном языке. Распознавание графических образов также связано с решением проблемы многозначной интерпретации.

*Ненадежность знаний и выводов.* Ненадежность знаний означает, что для оценки их достоверности нельзя применить двухбалльную шкалу (1 – абсолютно достоверные; 0 – недостоверные знания). Для более тонкой оценки достоверности знаний применяется вероятностный подход, основанный на теореме Байеса, и другие методы. Например, в экспертной системе MYCIN, предназначенной для диагностики и выбора метода лечения инфекционных заболеваний, разработан метод вывода с использованием коэффициентов уверенности.

*Неполнота знаний.* Абсолютно полных знаний не бывает, поскольку процесс познания бесконечен. В связи с этим состояние базы знаний должно изменяться с течением времени. В отличие от простого добавления информации, как в базах данных, при добавлении новых знаний возникает опасность получения противоречивых выводов, т.е. выводы, полученные с использованием новых знаний, могут опровергать те, что были получены ранее. Еще хуже, если новые знания будут находиться в противоречии со «старыми», тогда механизм вывода может стать неработоспособным. Многие экспертные системы первого поколения были основаны на модели закрытого мира, обусловленной применением аппарата формальной логики для обработки знаний. Модель закрытого мира предполагает жесткий отбор знаний, включаемых в базу, а именно: база

знаний заполняется исключительно верными понятиями, а все, что ненадежно или неопределенно, заведомо считается ложным. Другими словами, все, что известно базе знаний, является истиной, а остальное – ложью. Такая модель имеет ограниченные возможности представления знаний и таит в себе опасность получения противоречий при добавлении новой информации. Тем не менее эта модель достаточно распространена, например, на ней базируется язык PROLOG.

*Неточность знаний.* Известно, что количественные, лингвистические данные (знания) могут быть неточными, при этом существуют количественные оценки такой неточности (доверительный интервал, уровень значимости, степень адекватности и т.д.). Для учета их неточности используется теория нечетких множеств, предложенная Л. Заде в 1965 г. Этому ученому принадлежат слова: «Фактически нечеткость может быть ключом к пониманию способности человека справляться с задачами, которые слишком сложны для решения на ЭВМ». Развитие исследований в области нечеткой математики привело к появлению нечеткой логики и нечетких выводов, которые выполняются с использованием знаний, представленных нечеткими множествами, нечеткими отношениями, нечеткими соответствиями и т.д.

Областями внедрения алгоритмов нечеткой логики являются нелинейный контроль за процессами (производство), самообучающиеся системы (или классификаторы), исследование рискованных и критических ситуаций, распознавание образов, финансовый анализ (рынки ценных бумаг), исследование данных (корпоративные хранилища), совершенствование стратегий управления и координации действий, например, сложное промышленное производство.

Перечислим некоторые программные решения с использованием нечетких множеств (отрасль: экономика и финансы):

- Система оптимизации фондового портфеля (Siemens Business Services, СПб).

- IRC – калькулятор для оценки риска прямых инвестиций.

- Система для оценки инвестиционной привлекательности фондовых активов.

- МАСТЕР ФИНАНСОВ: Анализ и планирование (КГ «Воронов и Максимов», СПб.).

- AnyLogic – платформа для моделирования сложных финансовых и производственных систем (XJ Technologies, СПб.).

- Интеллектуальные экспертные системы (INEX, Киев).

Недостатками теории нечетких систем являются:

- отсутствие стандартной методики конструирования нечетких систем;

- невозможность математического анализа нечетких систем существующими методами;

- применение нечеткого подхода по сравнению с вероятностным не приводит к повышению точности вычислений.

## 2.5. Экспертные системы

В различных областях человеческой деятельности существует ряд задач, отличающихся следующими признаками:

- алгоритм решения неизвестен или не может быть использован из-за ограниченности ресурсов компьютера;
- задача не может быть определена в числовой форме;
- цели и задачи не могут быть выражены в терминах точно определенной целевой функции.

Попытка решения таких задач привела к созданию в области искусственного интеллекта отдельного направления – инженерии знаний. Суть подхода: использование знаний и превращения их в базу знаний. Решение задачи реализуется с помощью механизма логического вывода, использующего эти знания.

Для решения плохо формализованных задач применяются экспертные системы, нечеткая логика, нейросетевые алгоритмы.

*Экспертные системы* – это прикладные системы искусственного интеллекта, в которых база знаний представляет собой формализованные эмпирические знания высококвалифицированных специалистов (экспертов) в какой-либо узкой предметной области [27].

Экспертные системы предназначены для замены при решении задач экспертов в силу их недостаточного количества, недостаточной оперативности в решении задачи или в опасных (вредных) для них условиях.

Классы задач, решаемых экспертными системами:

- диагностика,
- прогнозирование,
- идентификация,
- управление,
- проектирование,

- мониторинг.

Области применения экспертных систем:

- медицина,
- вычислительная техника,
- военное дело,
- микроэлектроника,
- радиоэлектроника,
- юриспруденция,
- экономика,
- экология,
- геология (поиск полезных ископаемых),
- математика.

Архитектура экспертных систем включает в себя:

- базу знаний (хранилище единиц знаний);
- программный инструмент доступа и обработки знаний, состоящий из механизма вывода заключений (решений), приобретения знаний, объяснения получаемых результатов и интеллектуального интерфейса.

*База знаний* – совокупность единиц знаний, которая представляет собой формализованные с помощью некоторого метода представления знаний отражения некоторых объектов предметной области и их взаимосвязь, действий над объектами и, возможно, неопределенностей, с которыми эти действия осуществляются.

Свойства экспертных систем:

- компетентность, т.е. достижение в конкретной области того же профессионализма, что и люди-эксперты;
- поддержка символических рассуждений;

- глубина знаний, т.е. экспертная система должна эффективно работать в узкой предметной области, содержащей труд, нетривиальные задачи.

Классификация экспертных систем:

1. По способу формирования решения:

- аналитические (выбор из множества альтернатив, определения характеристик объекта);
- синтетические (генерация неизвестных решений, формирование).

2. По способу учета временного признака:

- статические (решают задачи при неизменяемых в процессе решения данных и знаний). Осуществляет монотонное непрерывное решение задачи от ввода исходных данных до конечного результата;

- динамические (допускают изменение). Предусматривает возможность изменения полученных ранее результатов.

3. По виду использования данных и знаний:

- с детерминированными (четко определенными знаниями);
- неопределенными (их неполнота, отсутствие, не достоверность, неточность измерений, двусмысленность или многозначность).

4. По числу источников знаний:

- одного источника знаний;
- несколько источников знаний.

Над процессом создания, конструирования экспертных систем занят коллектив разработчиков:

- инженер по знаниям – специалист в области искусственного интеллекта, выступающий в роли промежуточного буфера между экспертами и базой знаний;

- эксперт – специалист в предметной области знаний, который будет описывать (решать) экспертную систему;
- программист – специалист по разработке инструментальных средств (ИС), которые будет содержать экспертная система, а также осуществлять сопровождение ИС в той среде, где она будет использована;
- пользователь.

## 2.6. Концепции разработки правил экспертных систем

Разработка правил для экспертных систем базируется на двух известных концепциях – с прямой и обратной цепочкой рассуждений. Каждая концепция определяет стратегию выбора в конкретных условиях.

### *Стратегия прямого вывода*

Стратегия прямого вывода состоит в том, что задается последовательность вопросов, построенных таким образом, что каждый из них позволяет отбросить большую группу потенциальных ответов, благодаря чему значительно сужается пространство поиска. Так продолжается до тех пор, пока не остается один правильный ответ.

### *Стратегия обратного вывода*

В некоторых задачах имеется всего несколько решений при наличии огромных объемов входной информации. В этом случае целесообразно в каждый момент времени рассматривать только одно из возможных решений, а затем собрать и проверить все свидетельства, которые могут его подтвердить или опровергнуть. Существует много

приложений такого рода. К ним относятся, в частности, классификационные экспертные системы (найти причину неисправности, установить заболевание и др.).

На рисунке 44 приведены этапы разработки экспертных систем.

*Этап идентификации.* Назначение этапа: определение входных и выходных данных, осмысление всего процесса создания экспертной системы, определение требований к ней. Другими словами, опираясь на условие задачи (входные данные), следует ответить на вопрос, что надо сделать и какие ресурсы необходимо задействовать (участники процесса проектирования, их роли, выявление ресурсов и целей).

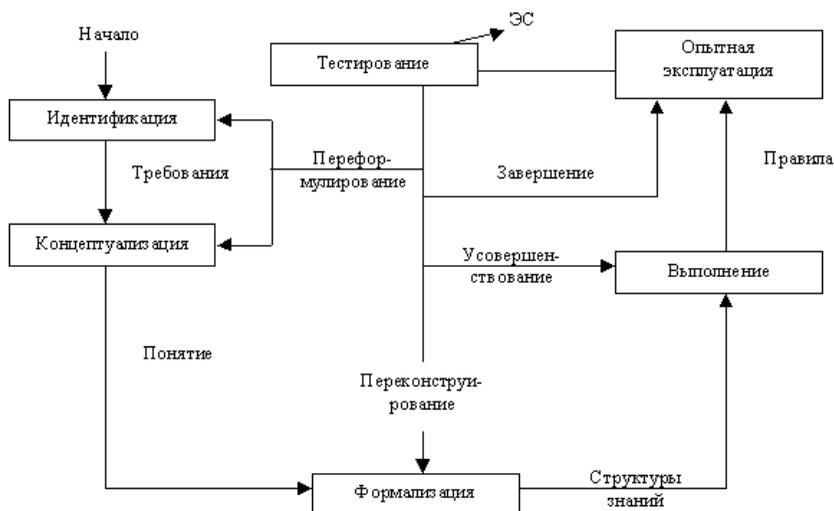


Рис. 44. Этапы разработки экспертных систем

*Этап концептуализации (или этап инфологического проектирования)* – создание информационной модели системы

без привязки к типу ЭВМ и типу системных программных средств; построение инфологической модели базы данных, которая наиболее полно описывает предметную область в терминах пользователя. Этот этап характеризуется определением следующих составляющих:

- типы доступных данных;
- исходные и выводимые данные, подзадачи общей задачи;
- используемые стратегии и гипотезы, виды взаимосвязей между объектами программного обеспечения, типы используемых отношений (иерархия, причина – следствие, часть – целое и т.п.);
- процессы, используемые в ходе решения;
- состав знаний, используемых при решении задачи; типы ограничений, накладываемых на процессы, используемые в ходе решения;
- состав знаний, используемых для обоснования решений.

Определим подходы к процессу построения модели предметной области на этапе концептуализации:

1. Признаковый (или атрибутивный) подход предполагает наличие полученной от экспертов информации в виде троек «объект – атрибут – значение атрибута», а также наличие обучающей информации. Этот подход развивается в рамках направления, получившего название *формирование знаний* или *машинное обучение*.

2. Структурный (или когнитивный) подход заключается в осуществлении элементов предметной области, их взаимосвязей и семантических отношений.

Созданная экспертная система должна обладать следующими свойствами:

- уникальностью, т.е. отсутствием избыточности;
- полнотой, т.е. достаточно полным описанием различных процессов, фактов, явлений рассматриваемой предметной области;
- достоверностью (или валидностью), т.е. соответствием выделенных единиц смысловой информации их реальным наименованиям;
- непротиворечивостью, т.е. отсутствием омонимии.

*Этап формализации.* На данном этапе определяется система программирования, синтаксические и семантические правила построения предложений, осуществляется запись составленных предложений с естественного языка на язык исполнителя.

*Этап выполнения.* Цель этого этапа – создание одного или нескольких вариантов экспертной системы, решающих требуемые задачи. Этот этап характеризуется наполнением базы знаний, тестированием всех предложенных прототипов экспертной системы, выявлением их работоспособности. При создании прототипа необходимо обеспечить проверку адекватности идей, методов и способов представления знаний решаемым задачам.

*Этап тестирования.* В ходе данного этапа производится оценка выбранного средства программирования, способа представления знаний в экспертных системах путем подбора примеров, обеспечивающих проверку всех возможностей разработанной экспертной системы.

*Этап опытной эксплуатации.* На этом этапе проверяется пригодность экспертной системы для конечного

пользователя, которая определяется удобством работы с ней и ее полезностью. Полезность – способность экспертной системы в ходе диалога определять потребности пользователя, выявлять и устранять причины неудач в работе, а также решать поставленные задачи. Удобство работы с экспертной системой заключается:

- в естественности взаимодействия с ней (общение в привычном, не утомляющем пользователя, виде);
- в обладании принципом гибкости (т.е. способности настраиваться на различных пользователей, учете изменений в квалификации одного и того же пользователя);
- устойчивости системы к ошибкам (т.е. не выходить из строя при ошибочных действиях неопытного пользователя).

### **Пример ЭС на правилах**

Пусть нам даны следующие сведения о кандидатах при поступлении на работу (см. рис. 45).

Выбранным считается кандидат, все качества которого удовлетворяют заказчику.

Если кандидат не подходит, то программа должна переключиться на следующего кандидата. При этом не должно быть повторяющихся вопросов к заказчику.

В конце работы программа должна вывести фамилию выбранного кандидата, либо сообщение о том, что подходящей кандидатуры нет.

Если качества нескольких кандидатов идентичны, то в случае успешного завершения поиска выбранным считается первый из них по порядку учетных записей.

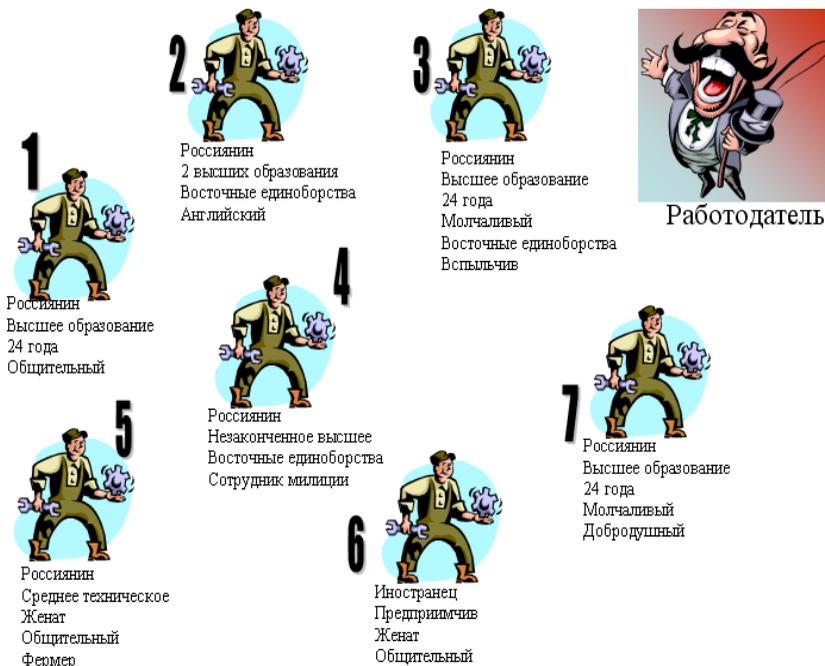


Рис. 45. Сведения о кандидатах

Качество кандидата считается подходящим, если:

1. Если заказчик подтвердил его для предыдущих кандидатов (т.е. оно входит во множество уже подтвержденных качеств).

Иначе:

2. Если на вопрос: «Вам подходит такое качество?» заказчик ответил утвердительно.

Два факта для правила «Проверка свойства»

1. Свойство подходит, если оно уже входит в множество ранее подтвержденных свойств.

2. Свойство подходит, если заказчику задан вопрос и на него дан утвердительный ответ.



## База знаний «Породы собак»

Порода	Характеристика
English Bulldog	short-haired dog height 55 sm low-set tail good natured personality
Beagle	short-haired dog height 55 sm longer ears good natured personality
Dog	short-haired dog low-set tail good natured personality weight 45 kg
...	...

Приведенные утверждения описывают только одно из указанных свойств или характеристик собаки. Поэтому в Прологе при создании базы знаний чаще всего используют продукционные правила (правила типа «если-то»). Каждое такое правило есть условное утверждение. Существуют различные варианты интерпретации этих правил:

- 1) ЕСЛИ <условие P> ТО <заключение C>;
- 2) ЕСЛИ <ситуация S> ТО <действие A>;
- 3) ЕСЛИ <выполняются условия C1 и C2> ТО <не выполняется условие C3>.

Каждая продукция должна обладать следующими свойствами:

- модульность: каждое правило описывает небольшой, относительно независимый фрагмент знаний;

- возможность инкрементного наращивания: добавление новых правил в базу знаний независимо от существующих правил;
- удобство модификации (следствие модульности): старые правила можно изменять и заменять на новые независимо от других правил;
- прозрачность системы как следствие применения правил: способность системы к объяснению принятых решений и полученных результатов.

Правило на языке Пролог для собаки породы English Bulldog будет выглядеть следующим образом:

```
dog_is("English Bulldog") :- it_is("short-haired dog"),
                             positive(has,"height 55 sm"),
                             positive(has,"low-set tail"),
                             positive(has,"good natured
                             personality"),!.
```

Как видно из правила, если собака имеет все перечисленные характеристики, то она имеет породу English Bulldog.

Собака породы Beagle имеет следующие характеристики:

```
dog_is("Beagle") :- it_is("short-haired dog"),
                    positive(has,"height 55 sm"),
                    positive(has,"longer ears"),
                    positive(has,"good natured
                    personality"),!.
```

Собаки приведенных пород имеют сходные и отличительные характеристики. Правило `it_is` производит идентификацию пород собак. Затем правило `positive/2` позволяет

идентифицировать характеристики собаки в каждом отдельном случае. И `it_is` и `positive` используются механизмом вывода. Применение предиката `it_is` позволяет ограничить «пространство поиска» (количество данных, проверяемых при поиске решения).

База данных должна хранить ответы пользователя на вопросы системы пользовательского интерфейса. Эти данные являются утвердительными или отрицательными ответами. Ниже приведена декларация доменов, базы данных, предикатов для выполнения вывода (машина вывода) и для взаимодействия с пользователем (система пользовательского интерфейса).

Механизм вывода имеет правила для управления данными, вводимыми пользователем, для унификации их с продукционными правилами и сохранения отрицательных и положительных ответов.

На рисунке 46 изображено окно программы экспертной системы на этапе исполнения.

Полное описание разработки экспертной системы на языке программирования Пролог можно найти в учебных пособиях Т.Н. Лебедевой [27; 28].

## **2.7. История создания языка программирования Пролог**

Начиная с 70-х годов XX века, специалистов в области программирования, информатики и вычислительной техники, инженерии знаний и искусственного интеллекта все больше привлекает логическое программирование. Свое название Пролог (Prolog) получил от словосочетаний «ПРОграммирование на языке ЛОГики».

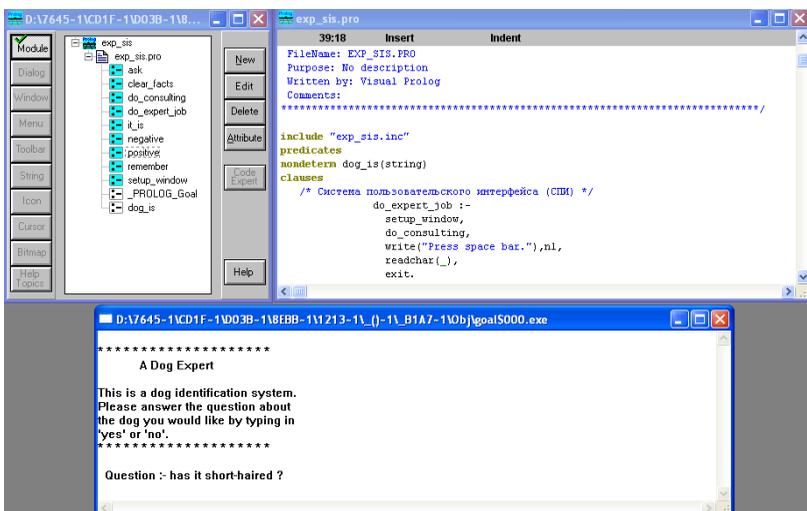


Рис. 46. Экспертная система «Идентификация пород собак»

Язык Пролог разработан в 1973 году Аланом Кольмероэ и его коллегами из Марсельского университета (Франция), в основу разработки которого положены работы Джона Робинсона, посвященные методу резолюций. Они создали программу для доказательства теорем, которая была реализована на языке Фортран и впоследствии получила название Пролог.

В течение многих лет предпринимались попытки к усовершенствованию теории логического программирования. Наиболее существенный вклад в ее развитие внесла работа Р. Ковальского «Логика предикатов как язык программирования». Ковальским и М. ван Эмденом (1976 г.) были предложены два подхода к прочтению текстов логических программ – процедурный и декларативный.

В 1977 г. Д. Уорреном и Ф. Перейра в университете Эдинбурга был создан интерпретатор/компилятор языка Пролог для ЭВМ DEC-10.

В 1980 г. К. Кларком и Ф. Маккейбом в Великобритании разработана версия Пролога для персональных ЭВМ.

В октябре 1981 г. начал разрабатываться японский проект создания ЭВМ пятого поколения. Логическое программирование было положено в основу методологии разработки программных средств. Цель проекта – создание систем обработки информации, базирующихся на знаниях. Главным средством реализации проекта должен был стать язык Пролог.

Начиная с 1981 г. появляется множество коммерческих реализаций Пролога практически для всех типов компьютеров. Наиболее известными являются CProlog, Quintus Prolog, Silogic Knowledge Workbench, Prolog-2, Arity Prolog, Prolog-86, Turbo Prolog и др.

В России наибольшую популярность завоевала система программирования Turbo Prolog. Первая версия этой системы разработана датской компанией Prolog Development Center (PDC) в содружестве с фирмой Borland International в 1986 г.

В 1988 г. разработана значительно более мощная версия Turbo Prolog 2.0, которая включала усовершенствованную интегрированную среду разработки программ, быстрый компилятор и средства низкоуровневого программирования.

В июне 1992 г. появилась версия 3.31, которая работала в среде MS DOS, OS/2, UNIX, MS Windows. Эта версия

является хорошо совместимой с традиционными языками программирования (например, с Си).

В 1996 г. Prolog Development Center (PDC) с участием российских программистов под руководством Виктора Юхтенко выпущена система Visual Prolog 4.0.

В декабре 1997 г. разработан Visual Prolog 5.0, с января 1999 г. распространена версия 5.1. Ознакомиться с особенностями проектирования логических программ в среде Visual Prolog версии 7.0 можно в Internet на сайте <http://www.visual-prolog.com>.

Пролог – это язык программирования, используемый для решения задач, в которых представлены объекты и связи (или отношения) между этими объектами.

Пролог может быть использован в различных приложениях искусственного интеллекта:

- общение с ЭВМ на естественном языке;
- формальные вычисления;
- написание компиляторов;
- базы данных;
- экспертные системы;
- различные области автоматизированного проектирования и т.д.

Пролог существенно отличается от процедурных языков программирования. Отличительными чертами языка Пролог являются способность к сравнению выражений (сопоставлению фактов) и внутренняя реляционная база данных. На основе правил своей внутренней базы данных и аппарат сравнения фактов, программа на Прологе может выдавать ответы на поставленные в логической форме вопросы.

В процедурных языках Basic, Algol, Pascal, C основным методом программирования является создание алгоритма, удовлетворяющее свойствам дискретности, массовости, детерминированности, конечности, результативности. Другими словами, от программиста требуется найти алгоритм решения задачи, который за конечное число шагов приведет к намеченной цели. Если при проектировании решения задачи на алгоритмическом языке программирования мы задаем вопрос «Как найти решение задачи?», то Пролог позволяет ответить на вопрос «Что надо делать?», не заботясь о разбиении решения задачи на элементарную последовательность шагов. При написании логических программ от программиста требуются умения составлять формальные описания ситуаций, используя понятия объектов различных типов и отношений между ними. Чем программист более формализованно подойдет к решению задачи, описанию набора фактов и предложений, тем быстрее будет проведен процесс унификации.

Пролог был принят в качестве базового языка в японской программе создания ЭВМ пятого поколения, ориентированной на исследование методов логического программирования и искусственного интеллекта, а также на разработку нового поколения компьютеров, специально предназначенных для реализации данных методов.

## Контрольные вопросы

1. Перечислите основные направления исследований в области искусственного интеллекта.
2. Дайте характеристику задачам, решаемым с помощью экспертных систем.
3. Перечислите основные разновидности систем извлечения новых знаний.
4. Приведите определение экспертной системы.
5. Каковы функции экспертной системы?
6. Опишите структуру экспертной системы.
7. Что такое база знаний?
8. Что представляет собой семантическая сеть? Приведите ее примеры.
9. Перечислите основные достоинства и недостатки семантических сетей представления знаний.
10. В чем состоит назначение машины логического вывода?
11. Приведите примеры использования экспертных систем в быту.
12. Назовите основное отличие «прямого» вывода в продукционных системах от «обратного».
13. Перечислите достоинства и недостатки продукционной модели знаний.
14. Дайте определение понятию «фрейм».
15. Перечислите основные компоненты (параметры) фрейма.
16. Перечислите достоинства и недостатки фреймовой модели представления знаний.
17. Перечислите компоненты теории нечетких знаний.
18. Назовите сферы применения нечетких знаний.

## Задания для самостоятельного выполнения

- Пусть имеется факт: «причиной неритмичной работы предприятия является старое оборудование, а причиной последнего – отсутствие оборотных средств». На основе данного факта построить семантическую сеть.

- Пусть имеется фраза «Иванов прилетает в Москву из Челябинска рейсом №\_\_ 12.12.2016 г.». Построить семантическую сеть.

- Построить логическую модель представления знаний, описав характеристики персонажей из детских сказок «Колобок», «Репка», «Гуси-лебеди».

- Построить фреймовую модель представления знаний, описав характеристики персонажей из детских сказок «Колобок», «Репка», «Гуси-лебеди».

### 3. КОРПОРАТИВНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

#### 3.1. Определение корпоративной информационной системы

Первые корпоративные информационные системы (КИС) как средства автоматизации деятельности производства и предприятий появились на Западе. Намного позже эти системы с успехом стали распространяться и в России.

В связи с этим вопрос о понятии КИС, подходов к ее построению в последнее время стал столь популярным. Несмотря на то, что понятие корпоративности подразумевает наличие крупной информационной системы, в настоящее время следует также под данным термином понимать и системы любых предприятий, вне зависимости от их масштаба и формы собственности.

Любая фирма, организация или государственное учреждение должны иметь собственную КИС, которая будет представлена в терминах отдельного участка или всего подразделения. КИС дает возможность автоматизации всех сфер деятельности производства, хранения и обработки информации, создания прогнозов на будущее его развитие и самосовершенствование.

Корпоративные информационные системы – это один из видов автоматизированных информационных систем, в частности к автоматизированным системам управления (АСУ). Порой этим термином называют системы, используемые на предприятии или корпорации. Так ли это на самом деле?

АСУ рассматривается как комплекс аппаратных и программных средств, предназначенный для управления различными процессами в рамках технологического процесса, производства, предприятия. Такие системы применяются в различных отраслях промышленности, энергетике, транспорте и т.п.

Термин «автоматизированная» оставляет за человеком оперирование некоторыми функциями наиболее общего, целеполагающего характера или не поддающихся автоматизации. Задача таких систем – планирование и эффективное управление предприятием.

В настоящее время можно встретить различные подходы к определению корпоративных информационных систем: определение с точки зрения стандартов качества, с точки зрения свойств информационных систем или уровня использования. Невозможно сказать, что есть одно наиболее полное и правильное определение. Следовательно, рассмотрим несколько определений.

*Корпоративная информационная система* – информационная система, охватывающая деятельность предприятия (организации, корпорации), происходящие в нем бизнес-процессы и организующая единое информационное пространство.

Такое определение выделяет одно из главных свойств корпоративной информационной системы – способность организовать единое информационное пространство предприятия.

Второй подход определяет КИС как совокупность специализированного программного обеспечения и аппаратной платформы для решения задач предприятия.

В этом определении имеется ограничение функций КИС только технической и программной составляющей, не выделены задачи системы.

Третье определение: КИС – это управленческая идеология, объединяющая бизнес-стратегию предприятия и передовые информационные технологии.

Оно основывается на подходе к информационной системе только с точки зрения менеджмента.

Четвертое определение: КИС – информационная система, охватывающая бизнес компании с целью предоставления оперативной информации для принятия решений.

Такой подход освещает главную цель создания информационной системы на предприятии – это принятие решений.

В конечном счете, можно определить КИС как стратегическую информационную систему, представляющую собой совокупность технических и программных средств, реализующих идеи и методы автоматизации всех функций управления предприятием [26].

Такая информационная система обязательно будет многопользовательской. При этом ее функционирование должно осуществляться в распределенной вычислительной сети. Конечно понятие корпоративности подразумевает наличие довольно крупной, территориально-распределенной информационной системы, однако правомерно присовокупить сюда системы любых предприятий, вне зависимости от их масштаба и формы собственности.

Исходя из определений, можно отметить, что КИС предназначены для обеспечения большинства бизнес-процессов предприятия (или нескольких), сбора и анализа

информации о предприятии и внешней среде с целью решения задач управления предприятием как по вертикали (от первичной информации до поддержки принятия решений высшим руководством), так и по горизонтали (все направления деятельности и технологические операции). Такие системы должны характеризоваться высоким быстродействием и простотой использования.

Рассмотрим роль КИС с точки зрения стандартов качества.

Сегодняшний рынок требует от предприятий достижения общепризнанных стандартов качества, как для производства материальных товаров, так и для сервиса и услуг. Для этой цели разработаны стандарты ISO 9000 (ИСО 9000). основополагающая идея стандартов: построение такой структуры управления процессом производства, которая гарантирует выпуск качественного продукта в любой момент, пока система действует.

Внедрение стандартов ИСО 9000 на предприятии требует серьезного пересмотра бизнес-процессов предприятия. При этом информационные технологии позволяют упростить процесс внедрения стандартов. Отсюда внедрение программных технологий корпоративного уровня и сертифицированной системы качества на многих предприятиях рассматривается как единая проблема.

Внедрение ИСО 9000 на предприятии предполагает использование как минимум трех классов программных продуктов:

- 1) комплексных систем управления предприятием. К ним можно отнести автоматизированные информационные системы поддержки принятия управленческих решений (АИСППР);

- 2) систем электронного документооборота;
- 3) продуктов, позволяющих создавать модели функционирования организации, проводить анализ и оптимизацию ее деятельности. Сюда можно отнести еще и системы нижнего уровня класса: автоматизированные системы управления техническими процессами (АСУТП) и системы автоматизированного проектирования (САПР), продукты интеллектуального анализа данных, а также программы, ориентированные на подготовку и поддержание функционирования систем качества.

Как видно из требований, ИСО 9000 не предполагает построения информационной системы на предприятии. Однако предприятиям, занимающимся внедрением стандартов, будет довольно сложно обходиться без корпоративной информационной системы.

Таким образом, система качества как часть системы управления предприятием сможет эффективно работать и приносить наибольшую выгоду, если ее поддерживают современные информационные системы принятий управленческих решений, разработанные и внедренные на предприятии в строгом соответствии со спецификой его запросов и уровня развития.

В этой связи можно дать следующее определение: КИС – это совокупность информационных систем подразделений предприятия, объединенных общим документооборотом. При этом каждая система выполняет свою часть задач по управлению предприятием решений, а вся система в целом обеспечивает функционирование предприятия в соответствии со стандартом качества ИСО 9000.

### 3.2. Требования к корпоративной информационной системе

Современные способы ведения бизнеса ставят перед предприятиями жесткие требования, корпорации уже не могут существовать без эффективной системы управления. В этом случае эффективно будет организовать управление с помощью современных информационных технологий. При этом требования могут меняться, потоки информации возрастать, что выдвигает требования и к информационным технологиям поддержки деятельности предприятия.

Именно поэтому корпоративные информационные системы постоянно совершенствуются. Рассмотрим требования с точки зрения их исторического появления:

- 1) системность;
- 2) комплексность;
- 3) модульность;
- 4) открытость;
- 5) адаптивность;
- 6) надежность;
- 7) безопасность;
- 8) масштабируемость;
- 9) мобильность;
- 10) простота освоения;
- 11) поддержка внедрения и сопровождение со стороны разработчика.

Поясним каждое требование в отдельности.

Требование «системность» можно рассматривать в совокупности требованием «комплексность». Система разрабатывается для всех уровней предприятия (корпорации)

в целом, включая филиалы, дочерние фирмы, сервисные центры и представительства даже географически удаленные.

Работа предприятия – это некий непрерывный процесс по созданию, обработке, изменению, передаче и хранению бизнес-информации. Рабочее место с этой точки зрения воспринимается как узел, который эту информацию создает или обрабатывает. Множество узлов предприятия связано между собой потоками информации. Это некая информационно-логическая модель предприятия в виде узлов и связей между ними, отражающими потоки информации.

Системность подразумевает, что КИС строит такую информационно-логическую модель, которая охватывает всю деятельность предприятия, логически обосновывает их, выявляет механизмы достижения главной цели – максимизации прибыли в рыночных условиях.

Требование комплексности говорит о том, что эффективно решить такую задачу возможно только при строгом учете максимально возможного обоснованного множества параметров и возможности многокритериального поливариантного анализа, оптимизации и прогнозирования.

Информационно-логическая модель предприятия обычно носит распределенный характер и может быть структурирована на каждом узле (рабочем месте) и в каждом информационном потоке. Некоторые узлы и потоки группируются в подсистемы (например, производственная, финансовая и т.д.). Такое деление на подсистемы выдвигает требование модульности построения КИС. Оно важно также с точки зрения внедрения информационной системы

на предприятии. Можно распараллелить и облегчить процесс инсталляции программных продуктов, ускорить подготовку персонала к использованию системы, ускорить запуск ее в промышленную эксплуатацию. Требование модульности позволяет экономить средства предприятия: при приобретении готовых корпоративных систем можно отказаться от тех компонентов, которые не включены в конкретную информационно-логическую модель предприятия, или без которых можно обойтись на первых этапах внедрения.

Следующее требование открытости важно тем, что при покупке «коробочных» (готовых) решений невозможно говорить о полном соответствии их специфике конкретного предприятия. Даже если корпоративная информационная система создается по специальному заказу, то она не может решить все запросы предприятия. Здесь необходимо руководствоваться принципом: невозможно объять необъятное. Следовательно, чтобы система оставалась работоспособной и востребованной на предприятии, она должна разрабатываться как открытая.

Предыдущее требование связано с требованием адаптивности. Предприятие не может существовать в замкнутом пространстве. Оно реагирует на изменения рынка, требования спроса и предложения. Со временем изменяется структура самого предприятия. Это выдвигает требование адаптивности системы. Данная система должна иметь возможность гибкой настройки под различные виды законодательств, предоставлять возможности использования разноязыковых интерфейсов, предлагать вести учет в нескольких валютах и т.д.

В идеале КИС должна иметь инструменты, средства настройки, с помощью которых программисты или специализированные пользователи получают возможность самостоятельной настройки системы. Например, это есть в системе программ 1С:Предприятие.

Требования безопасности важно, когда КИС уже внедрена на предприятии и становится незаменимым компонентом его деятельности. Если такая система выйдет из строя, застопорив все бизнес-процессы на предприятии, это принесет глобальные потери. Следовательно, требование надежности подразумевает возможность непрерывного функционирования всей системы в целом, даже при выходе из строя некоторых отдельных ее элементов при возникновении непредвиденных или непреодолимых причин.

Требование безопасности включает:

1) защиту данных от потери. Реализовать защиту в системе можно с помощью как организационных решений на местах, так и на уровне технических устройств или на уровне операционной системы. Однако это не значит, что КИС обязательно должна иметь средства резервного копирования и восстановления данных. Подобные вопросы могут быть решены специалистами на уровне операционной среды;

2) сохранение целостности и непротиворечивости данных. КИС призвана следить за всеми изменениями, которые происходят в базе, тем самым обеспечивается работа с различными версиями блоков информации;

3) предотвращение несанкционированного доступа к данным внутри системы. Система должна предотвращать возможности такого доступа. Это решается организационными

мероприятиями и на уровне операционных и прикладных систем. Компоненты информационной системы должны иметь средства администрирования для ограничения доступа к данным, отдельным функциям системы в зависимости от статуса пользователя, а также вести мониторинг действий пользователей в системе;

4) предотвращение несанкционированного доступа к данным извне. Решение данной проблемы реализуется на аппаратной и операционной стороне и требует ряда административно-организационных мероприятий.

Любое предприятие может со временем выявить тенденцию к росту, развитию дочерних фирм, филиалов. Это потребует от КИС увеличения количества рабочих мест – узлов, увеличения объема хранимой и передаваемой информации – информационных потоков.

Требование масштабируемости дает КИС возможность роста и развития предприятия, изменения его информационно-логической модели с условием сохранения технологии управления.

Тенденцию к изменению имеют и программно-аппаратные платформы, используемые на предприятиях. Требование мобильности говорит о том, чтобы, будь то изменение или переезд на другую программно-аппаратную платформу, покупка новой техники или операционной системы, это все не привело к кардинальным изменениям в управленческом процессе, не потребовало капитальных вложений на приобретение дополнительных компонентов системы. Этот процесс должен быть максимально простым и минимально затратным.

Требование простоты в освоении – это наличие интуитивно понятного интерфейса всех прикладных программ, входящих в состав корпоративной информационной системы, наличие сопроводительной документации.

В требование поддержки разработчика входит: обновление версий используемого программного комплекса, сопровождение процесса эксплуатации методической литературой для пользователей, поддержка по «горячей линии», получение информации о новинках в программной линейке разработчика, участие в конференциях, семинарах пользователей и других мероприятиях, проводимых компанией-разработчиком.

Требование сопровождения КИС предполагает оказание услуг специалиста компании-разработчика на объекте заказчика для устранения возникших проблем; техническое обучение непосредственно на предприятии, сопровождение всех изменений в системе методической помощи, практическими рекомендациями, установку новых версий или обновлений отдельных компонентов системы силами заказчика или разработчика.

Данное требование не распространяется на необходимость глобальной реструктуризации системы или новой разработки, возникшей как потребность со стороны предприятия, на котором используется КИС.

Рассмотренные требования выдвигаются к КИС со стороны предприятия. При этом разработчиками КИС также выдвигаются требования, обеспечивающие полноценную функциональность программы. Это могут быть возможные версии сетевой операционной системы, версии

операционных систем клиентских станций, версии систем управления базами данных, необходимость установки комплексов для обеспечения безопасности системы, архивации, резервного копирования и т.п. Полный список требований обычно содержится в сопроводительной документации к корпоративной информационной системе.

### **3.3. Архитектура корпоративных информационных систем**

Существуют различные подходы к определению архитектуры корпоративных информационных систем:

1) подход, рассматривающий архитектуру КИС как вид программных средств, входящих в эту систему и аппаратную платформу, где они установлены;

2) подход, определяющий архитектуру корпоративной информационной системы как информационную систему вообще;

3) подход, основывающийся на рассмотрении архитектуры как набора уровней КИС.

Первый подход выделяет следующие виды специализированного программного обеспечения, входящего в состав любой КИС:

- система управления корпоративной базой данных;
- 1) система автоматизации документооборота;
- 2) система управления электронными документами;
- 3) специальные программные средства;
- 4) системы поддержки принятия решений.

Второй подход, определяющий архитектуру корпоративной информационной системы, выделяет двухуровневую клиент-серверную архитектуру, трехуровневую (многоуровневую) и распределенную архитектуру.

Третий подход к архитектуре КИС выделяет в ней несколько уровней: информационно-логический, прикладной, системный, аппаратный и транспортный. Рассмотрим каждый уровень более подробно.

Информационно-логический уровень – это совокупность узлов создания, потребления, обработки информации и потоков данных между этими узлами. Он может быть представлен в виде модели, на основании которой разрабатываются структуры баз данных, системные соглашения и организационные правила для обеспечения взаимодействия компонентов прикладного программного обеспечения. Модель строится на основании бизнес-логики компании с учетом всех бизнес-процессов, протекающих в ней.

Прикладной уровень – совокупность прикладных программ и программных комплексов, которые реализуют функционирование информационно-логической модели. Это системы электронного документооборота, системы контроля над исполнением управленческих решений, системы сетевого планирования, автоматизированные системы управления техническими процессами, системы автоматизированного проектирования, бухгалтерские системы, офисные пакеты, системы управления финансами, кадрами, логистикой и т.д.

Системный уровень представлен операционными системами и сетевыми средствами, на основании которых строится вся жизнедеятельность КИС.

Аппаратный уровень – это все средства вычислительной техники, включенные в КИС.

Транспортный уровень представляет собой все активное и пассивное сетевое оборудование, сетевые протоколы и технологии, задействованные для поддержания деятельности КИС.

### **3.4. Классификация корпоративных информационных систем**

Выделяются несколько признаков классификации КИС:

- функциональные возможности;
- масштабы предприятия;
- стоимость проекта внедрения;
- используемая программно-аппаратная платформа;
- наличие отраслевых решений.

Первый подход делит корпоративные информационные системы на два больших класса: финансово-управленческие и производственные системы.

Финансово-управленческие системы включают подклассы локальных и малых интегрированных систем. Такие системы предназначены для учета по одному или нескольким направлениям (например, бухгалтерия, склад, учет кадров и т.д.). Системами этой группы могут пользоваться любые предприятия с целью управления финансовыми потоками и автоматизации учетных функций.

Системы такого класса по многим критериям выступают как универсальные, разработчики предлагают готовые решения отраслевых проблем. В этом случае можно

воспользоваться «коробочным решением». Например, система программ фирмы «1С»: 1С:Бухгалтерия, 1С:Зарплата и кадры, и др.

Производственные системы включают подклассы средних и крупных интегрированных систем. Эти системы предназначены для управления и планирования производственного процесса.

Производственные системы часто ориентированы на одну или несколько отраслей и/или типов производства: серийное сборочное (электроника, машиностроение), малосерийное и опытное (авиация, тяжелое машиностроение), дискретное (химия, металлургия, упаковка), непрерывное (газо- и нефтедобыча).

На организацию производственной системы влияет тип организации самого производственного процесса. Например, циклическое повторное или производство на заказ и др. Примеры систем фирм 1С, Галактика, Ваан, Oracle и др.

Все эти спецификации отражаются на наборе функций производственной системы, в существовании бизнес-моделей данного типа производства. Такие системы по многим параметрам значительно более жесткие, чем финансово-управленческие. Ядро системы – воплощенные в ней рекомендации по управлению производством.

Финансово-управленческие системы также могут быть использованы на небольших производственных предприятиях, где процесс производства несложен. Например, 1С:Управление небольшой фирмой.

Следующий подход к классификации представлен на рис. 47.

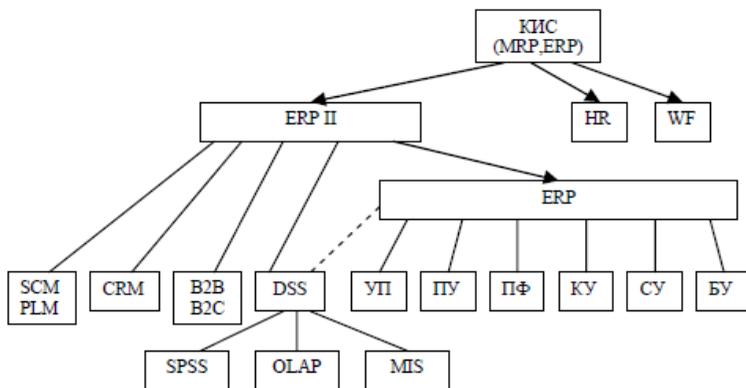


Рис. 47. Классификация КИС

Здесь КИС делятся на следующие системы:

- ERP II – расширение ERP-системы;
- HR – управление персоналом, может рассматриваться как самостоятельная задача, так и как входящая в состав ERP;
- WF – электронный документооборот (электронная почта).

ERP II состоит из:

- 1) SCM – управление цепочками поставок;
- 2) PLM – управление жизненным циклом продукции;
- 3) CRM – управление отношениями с клиентами (WF – частный случай CRM);
- 4) B2B – электронная торговая площадка («онлайновый бизнес»);
- 5) DSS – поддержка принятия управленческих решений:
  - a) SPSS – статистический анализ данных;
  - b) OLAP – анализ многомерных данных;

- с) MIS – автоматизированное рабочее место (АРМ) руководителя;
- б) ERP, включающей: УП – управление производством; ПУ – первичный учет; ПФ – планирование и бюджетирование и др.

### 3.5. Типы корпоративных информационных систем

Как уже было отмечено, КИС относится к АСУ и представляется в виде взаимодействующих автоматизированных систем разного уровня:

1. Уровень предприятия – уровень систем класса ERP и MRP. Реализуется автоматизированными системами управления предприятием (АСУП), в настоящее время все чаще используется термин «корпоративные информационные системы».

2. Уровень управления производством – уровень систем класса Manufacturing Execution System (MES).

3. Уровень управления технологическими процессами – уровень Distributed Control System (DCS). Реализуется автоматизированными системами управления технологическими процессами (АСУТП).

Уровень управления производством реализуется сопряжением между АСУП и АСУТП. Рассмотрим более подробно только автоматизированные системы уровня предприятия – КИС – исходя из периодов их развития (рис. 48). Система любого типа, представленная на рис. 48, включает в себя системы более ранних типов. Это значит, что системы всех типов сосуществуют и в настоящее время.

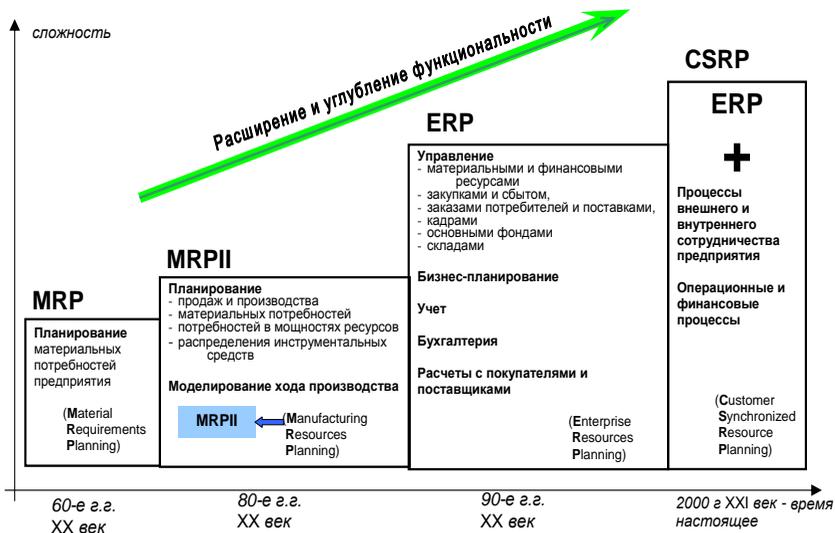


Рис. 48. Периоды развития ИС

### 3.5.1. Системы стандарта MRP

В начале 60-х гг. XX века стали популярными идеи использования возможностей компьютера с целью планирования деятельности предприятия, планирования производственных процессов. Появился первый стандарт управления бизнесом – MPS – Master Planning Scheduling, объемно-календарное планирование (рис. 49).

С течением времени вопросы управления запасами становились сложнее. И как результат появилась методология планирования производства – MRP – Material Requirement Planning, планирование потребности в материалах.

Следует различать два термина: 1) MRP-программа – компьютерная программа, которая реализует алгоритм на основе MRP-методологии и 2) собственно MRP-методология.



Рис. 49. MPS-стандарт

Процесс работы MRP-программы представлен на рис. 50.



Рис. 50. MRP-программа

Использование такой MRP-системы для планирования производственных потребностей позволяет оптимизировать время поступления каждого материала, снизить складские издержки и облегчить ведение производственного учета.

### *3.5.2. Системы стандарта MRP II*

Потребности предприятий для производства товаров и услуг росли во времени и MRP-системы уже из не удовлетворяли. И в конце 70-х гг. XX века заговорили о воспроизведении замкнутого цикла (Closed Loop, замкнутый цикл) в таких программах. Необходимо было ввести в рассмотрение более широкий спектр факторов при проведении планирования путем введения дополнительных функций.

Так к набору стандартных функций программы добавили дополнительные. Например, контроль соответствия количества произведенной продукции количеству использованных в процессе сборки комплектующих, составление регулярных отчетов о задержках заказов, об объемах и динамике продаж продукции, о поставщиках и т.д.

Смысл состоит в том, что созданные в процессе ее работы отчеты анализируются и учитываются на дальнейших этапах планирования, изменяя при необходимости программу производства и план заказов. Система дает обратную связь и планирование становится гибким по отношению к внешним факторам.

Систему назвали MRP II-системой, Manufacturing Resource Planning System, система планирования ресурсов предприятия. В 80-х гг. XX века для внесения различий в принцип работы системы ввели дополнение II или MRP второго поколения.

Американским обществом по управлению производством и запасами, American Production and Inventory Control Society (APICS) разработало стандарт MRP II, который до сих пор поддерживается.

Работа MRP II-системы отражена на рис. 51.

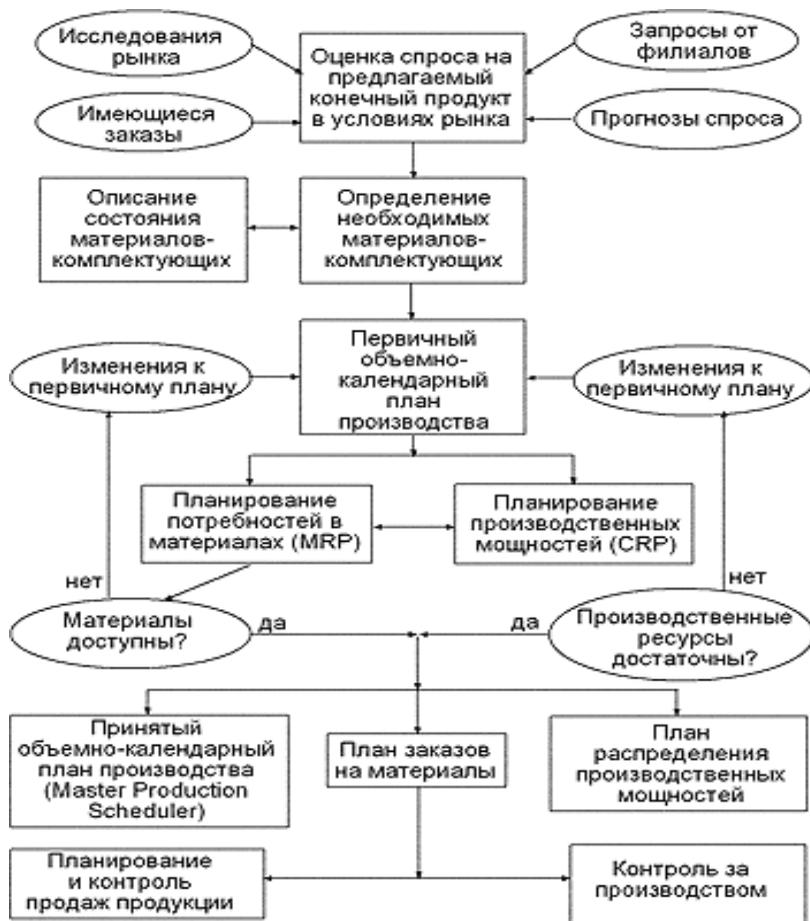


Рис. 51. MRP II-система

Результаты использования КИС стандарта MRP II:

1) получение оперативной информации о текущих результатах деятельности предприятия, как в целом, так и с полной детализацией по отдельным заказам, видам ресурсов, выполнению планов;

2) долгосрочное, оперативное и детальное планирование деятельности предприятия с возможностью корректировки плановых данных на основе оперативной информации;

3) решение задач оптимизации производственных и материальных потоков;

4) реальное сокращение материальных ресурсов на складах;

5) планирование и контроль за всем циклом производства с возможностью влияния на него в целях достижения оптимальной эффективности в использовании производственных мощностей, всех видов ресурсов и удовлетворения потребностей заказчиков;

6) автоматизация работ договорного отдела с полным контролем за платежами, отгрузкой продукции и сроками выполнения договорных обязательств;

7) финансовое отражение деятельности предприятия в целом;

8) значительное сокращение непроизводственных затрат;

9) защита инвестиций, произведенных в информационные технологии;

10) возможность поэтапного внедрения системы с учетом инвестиционной политики конкретного предприятия.

MRP II класса активно используются крупными и средними предприятиями.

### 3.5.3. Системы ERP

Предприятия постоянно развиваются и расширяют зоны своего действия. И как следствие к середине 90-х гг. XX века появились системы класса ERP.

Термин используется в двух значениях.

1. ERP-система – это информационная система для идентификации и планирования всех ресурсов предприятия, которые необходимы для осуществления продаж, производства, закупок и учета в процессе выполнения клиентских заказов.

2. ERP как методология – методология эффективного планирования и управления всеми ресурсами предприятия, которые необходимы для осуществления продаж, производства, закупок и учета при исполнении заказов клиентов в сферах производства, дистрибьюции и оказания услуг.

Это не только информационная система, но и методология управления, которую реализует и поддерживает такая информационная система.

Рассмотрим отличия ERP- и MRP II-систем. Основой для них является производство. При этом ERP используются на больших многофункциональных и территориально распределенных производственных корпорациях. MRP II предназначены для средних предприятий, не требующих всей мощности ERP.

Это можно выразить с помощью формулы:

ERP = MRP II + реализация всех типов производства + интегрирование планирования ресурсов по различным направлениям деятельности компании + многозвенное планирование.

Концепция ERP – это некая надстройка над MRP II. По сути такая система не вносит изменения в механизм планирования производственных ресурсов, она решает дополнительные задачи, которые появились в связи с усложнением структуры компании.

ERP до сих пор не оформлена в виде стандарта, потому что специалисты не сходятся в определении критериев принадлежности системы к такому классу.

Производственный процесс ERP-системы может быть отражен на рис. 52.

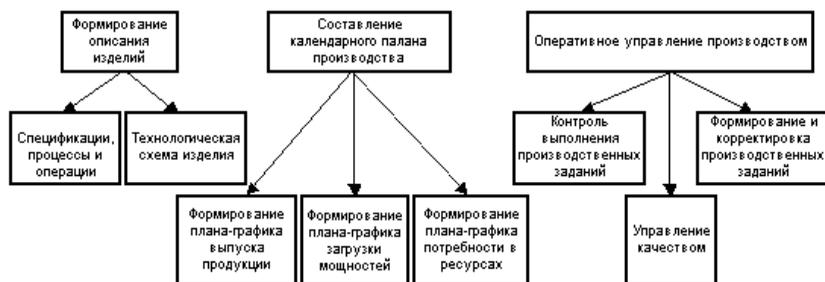


Рис. 52. Производственный процесс в рамках ERP-системы

Управление материальными потоками представлено на рис. 53.



Рис. 53. Управление материальными потоками

Финансовый раздел на рис. 54.



Рис. 54. Финансовый раздел

### 3.5.4. Системы класса CRM

CRM, Customer Relations Management, означает управление отношениями с клиентами. Системы такого класса появились в результате поиска предприятиями возможности установления и налаживания связей с клиентами. В настоящее время такие системы представляют собой целую стратегию применения управленческих и информационных технологий для аккумуляции знаний о клиентах и выстраивания с ними взаимовыгодных отношений. Все это должно способствовать увеличению прибыли.

CRM определяют в широком смысле как стратегию «отличительного» ведения бизнеса. В узком смысле – это информационные технологии, которые позволяют формализовать и автоматизировать различные аспекты взаимодействия с клиентами подразделений маркетинга, продаж и сервисного сопровождения на основе автоматических/автоматизированных процессов (в том числе сбытовых) и единого информационного пространства организации.

CRM можно классифицировать по основным категориям:

- SFA (Sales Force Automation) – автоматизация деятельности торговых представителей.
- MA (Marketing Automation) – автоматизация деятельности маркетинга.
- CSA, CSS (Customer Service Automation, Customer Service Support) – автоматизация службы поддержки и обслуживания клиентов.
- Call/Contact Center Management – центры обработки вызовов, контакт-центры.
- Field Service Management – управление территориально удаленными подразделениями или пользователями.
- PRM (Partner Relationship Management) – управление взаимоотношениями с партнерами (не поставщиками, а элементами товаропроводящей сети, разделяющими риски).
- Help Desk – техническая поддержка пользователей.

Широкое распространение интернет-технологий порождает новый уровень развития стратегии CRM. Такое взаимодействие может быть выражено формулой:

$$\text{CRM} + \text{интернет} = \text{eCRM}.$$

Можно отметить, что такие технологии позволяют определить дальнейшее развитие КИС. Объединение ERP-систем с системами класса CRM называется интеллектуальными системами планирования ресурсов предприятия (IRP, Intelligence Resource Planning), которые выражаются следующей формулой:

$$\text{ERP} + \text{CRM} = \text{IRP}.$$

### 3.5.5. Системы CSRP

Системы класса CRM интегрируют с системами управления предприятием (MRP II и ERP), однако даже такое решение может не дать ожидаемого эффекта. Причиной является то, что обычно вычисление себестоимости продукции выполняется методом прямых расходов (direct cost), который учитывает затраты на оборудование, материалы и комплектующие, рабочую силу, технологический процесс, а затраты на сервис, логистику и маркетинг очень часто рассматриваются как накладные расходы.

Более современной концепцией управления ресурсами предприятия является CSRP, Customer Synchronized Resource Planning, планирование ресурсов, синхронизированное с клиентом. Система призвана захватить почти весь жизненный цикл товара. Это позволит на порядок точнее управлять стоимостью товара, учитывая производство, продвижение и обслуживание товара данного типа, и учитывать все элементы его функционального жизненного цикла, а не только производства, как во всех стандартных системах предыдущих поколений.

Сущность концепции CSRP заключается в следующем: при планировании и управлении компанией можно и нужно учитывать не только основные производственные и материальные ресурсы предприятия, но и все те, которые обычно рассматриваются как «вспомогательные» или «накладные». Это могут быть: ресурсы, потребляемые во время маркетинговой и «текущей» работы с клиентом, послепродажного обслуживания реализованных товаров, используемые для перевалочных и обслуживающих операций,

а также внутрицеховые расходы. Учет абсолютно всех использованных ресурсов имеет решающее значение для повышения конкурентоспособности предприятия в отраслях, где жизненный цикл товара невелик, и требуется оперативно реагировать на изменение желаний потребителя.

Следствие такой системы – реализация задачи тонкого управления производственными графиками в условиях ограниченных мощностей или APS-задачи, *Advanced Planning and Scheduling*, расширенного управления производственными графиками. Системы типа APS позволяют решать такие задачи, как «проталкивание» срочного заказа в производственные графики, распределение заданий с учетом приоритетов и ограничений, перепланирование с использованием полноценного графического интерфейса. Благодаря принципиально новой «математике», расчет типовых задач MRP осуществляется значительно быстрее, чем раньше.

### *3.5.6. Системы ERP II*

На Западе ERP считается пройденным этапом. Однако для российских предприятий это не так. Консалтинговая компания Gartner Group заявила о завершении эпохи ERP-систем в 1999 г. Ее заменяет методология управления внутренними ресурсами и внешними связями предприятия ERP II, *Enterprise Resource and Relationship Processing*.

Положение ERP II-систем представлено на рис. 55.

ERP II представляет собой бизнес-стратегию, которая формируется как набор специфичных для отрасли приложений, которые, в свою очередь, позволяют осуществлять внутренние и внешние бизнес-процессы, совместные операционные и финансовые инициативы, и дают возможность оптимизировать их.

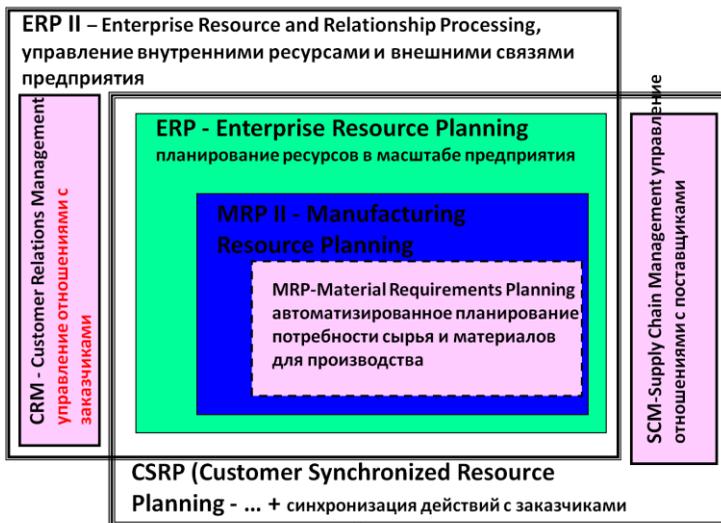


Рис. 55. Взаимосвязь различных видов КИС

Ключевые сферы таких систем – бухгалтерский учет, купля-продажа, ввод заказов и калькуляция себестоимости. Основное отличие заключается в использовании информации. ERP II интегрируется и в непромышленные области. Управленческая информация, ранее хранимая и применяемая только внутри предприятия, теперь должна быть доступной (разумеется, с разумными ограничениями) для информационных систем клиентов и партнеров.

Таким образом, ERP II выступает результатом развития методологии и технологии ERP в направлении более тесного взаимодействия предприятия с его клиентами и контрагентами. При этом управленческая информация компании не только используется для внутренних целей, но и служит для развития отношений сотрудничества с другими организациями.

Система ERP II имеет свои технологические особенности, она может быть реализована на основе интернет-ориентированной архитектуры. Следовательно, традиционная клиент-серверная архитектура начинает уступать место веб-клиентам и распределенным компонентным технологиям.

### 3.6. Примеры корпоративных информационных систем

По данным компании TAdviser на сентябрь 2016 года самыми популярными решениями КИС являются программы компаний 1С, Microsoft, SAP и Галактика (рис. 56) [1].

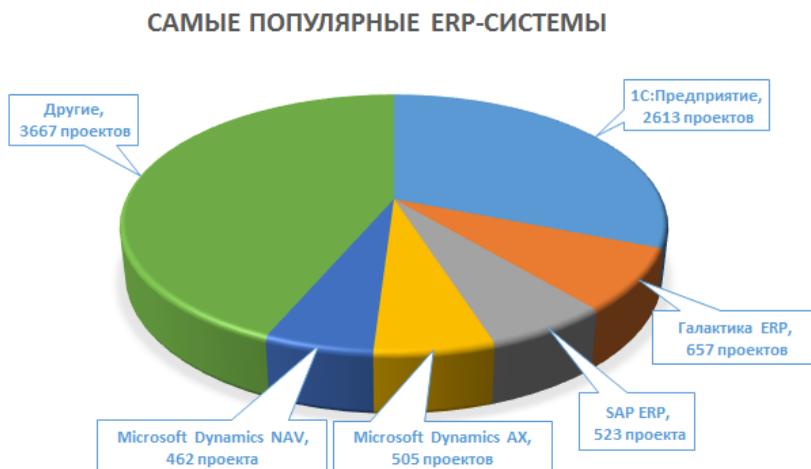


Рис. 56. Популярные КИС

Таким образом, на российском рынке корпоративных информационных систем присутствуют как системы

зарубежных производителей, так и российских. Однако в связи с политикой импортозамещения все большее число компаний переходят на продукты российских разработчиков.

Наиболее распространенными отечественными КИС являются: «Галактика», «Парус», «БОСС», NS2000; «БЭСТ-ПРО», IBS Trade House, «Аккорд», «Альфа», «Флагман», «Vrsystem», «Супер-Менеджер», «Ресурс», «Эверест» и др.

Аналитики компаний-разработчиков КИС говорят о том, что ERP-рынок и рынок приложений для бизнеса как в России, так и на постсоветском пространстве еще далек от насыщения. По их данным, в Западной Европе и Америке компании выбирают уже вторую и третью систему, в то время как российские предприятия только собираются внедрять свою первую полноценную комплексную систему управления.

Этим обусловлен интерес многих разработчиков ERP-систем: огромная территория с неосвоенной базой потенциальных заказчиков. Российский рынок остается очень динамичным и благоприятным для расширения бизнеса. Однако вопрос с выбором комплексной КИС для предприятий среднего уровня всегда принять сложнее, так как таким компаниям необходима быстрая отдача от внедрения.

Рассмотрим примеры программ, относящихся к различным видам корпоративных информационных систем.

### **HansaWorld Enterprise**

HansaWorld Enterprise – это многоплатформенная профессиональная финансовая система управления предприятием, обладающая встроенными возможностями электронной коммерции, и создания корпоративного портала

(комплексная автоматизация торговли, производства, оказания услуг, бухгалтерии, CRM, ERP и др.) (рис. 57) [42].



Рис. 57. Главное окно HansaWorld

Система включает более 45 интегрированных модулей, отличительными чертами HansaWorld Enterprise являются многоплатформенность, мультивалютность, работа в глобальных сетях, решение для гостиниц, поддержка работы с вариантами товаров, производство, интегрированный веб-магазин.

### Масоному

Масоному – это интегрированные решения по управлению бизнесом, которые разрабатывались с учетом опыта практического использования в крупнейших компаниях-клиентах по всему миру и предполагают следующие возможности:

- управление взаимоотношениями с клиентами (CRM),
- управление продажами, закупками и складами,
- управление проектами,
- планирование ресурсов,
- бюджетирование,
- финансовый и управленческий учет,
- управление персоналом и расширенные возможности по созданию отчетов.

Интерфейс продукта представлена на рис. 58.

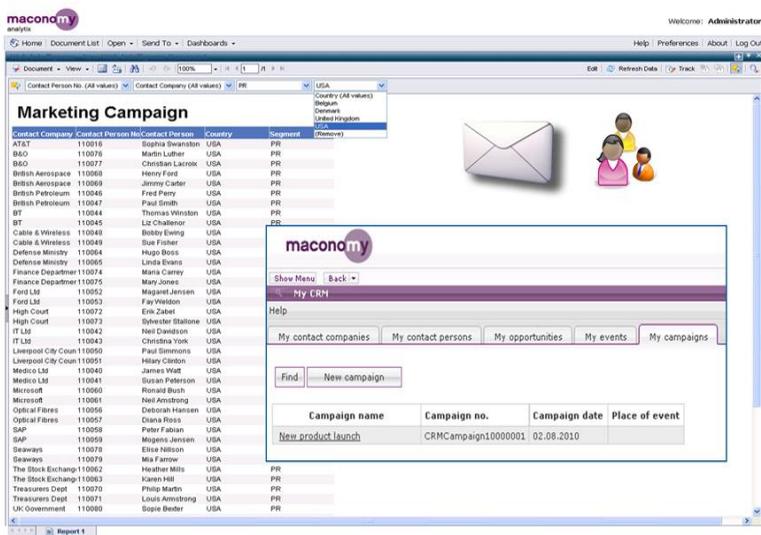


Рис. 58. Интерфейс Mazonomy

## КОМПАС

КОМПАС позиционируется производителями как полнофункциональная ERP-система. Программа позволяет [46]:

- собирать и анализировать информацию о состоянии компании (финансовом, товарном, кадровом и т.п.);

- осуществлять бюджетирование и финансовое планирование;
- взаимодействовать с клиентами и поставщиками на основе современных клиент-ориентированных технологий;
- находить «узкие» места;
- планировать бизнес на перспективу;
- вести бухгалтерский и налоговый учет;
- управлять производством по стандарту MRP-II;
- организовывать полноценное управление персоналом и расчет заработной платы;
- автоматизировать систему менеджмента качества.

Интерфейс программы представлен на рис. 59.

The screenshot shows a software window titled "Личная карточка с данными о стаже" (Personal card with service data). The window contains the following information:

- Header:** "ЛИЧНАЯ КАРТОЧКА" (PERSONAL CARD) with ID "00003".
- Personal Data:**
  - Пол (м.ж.): ж (Female)
  - Фамилия: Иванова
  - Имя и отчество: Ольга Владимировна
  - Дата приема: 01.06.1999
  - Дата увольнения: (empty)
- Navigation:** Tabs for "Анкета", "Банк", "Образование", "Семья", "Зарплата", "ИФНС", "ВУС", "Документы", "Фото".
- Main Sections:**
  - Основные данные:**
    - Рождение:
      - Дата: 10.06.1969
      - Место: Санкт-Петербург
  - Паспорт:**
    - Код страны: 643
    - Гражданство: РОССИЯ
    - Код документа: 01
    - Серия: ХХIII-АК N 242234
    - Дата выдачи: 20.01.1976
    - Кем выдан: 12 о/м г. Ленинграда
  - Адрес:**
    - Почтов. адрес: С.-Пб, ул. Малая Посадская, д. 12, кв. 39
    - Телефоны: 233-28-48
- Buttons:** "OK", "Сохранить", "Отмена", "Справка", and navigation arrows.

Рис. 59. Интерфейс программы КОМПАС

На сегодняшний момент компания-разработчик готова представить новую облачную ERP-систему «КОМПАС-CLOUD».

## Парус-Предприятие 8

Система класса ERP «Парус-Предприятие 8» представляет собой комплексную систему управления предприятием. В стандартный набор системы входят средства, позволяющие автоматизировать бухгалтерский учет, основные торговые процессы и складской учет, расчет заработной платы, кадровый учет [47] (рис. 60).

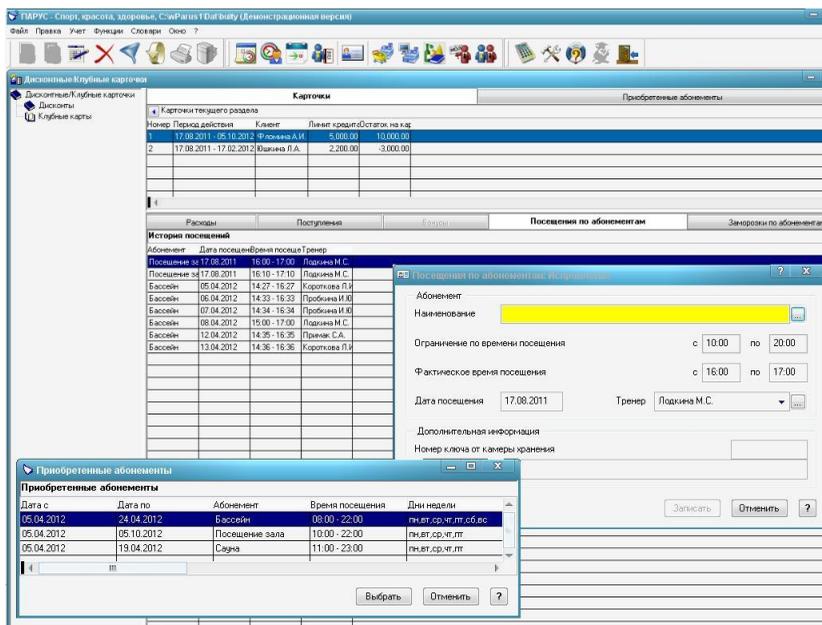


Рис. 60. Интерфейс «Парс-Предприятие»

## 1С:Управление производственным предприятием

1С:Управление производственным предприятием 8 является комплексным прикладным решением, охватывающим основные контуры управления и учета на производственном предприятии. Решение позволяет организовать комплексную информационную систему, соответствующую корпоративным, российским и международным стандартам и обеспечивающую финансово-хозяйственную деятельность предприятия.

Программа создает единое информационное пространство для отображения финансово-хозяйственной деятельности предприятия, охватывая основные бизнес-процессы, разграничивает доступ к хранимым сведениям, а также возможности тех или иных действий в зависимости от статуса работников [43]. Интерфейс продукта представлен на рисунке 61.

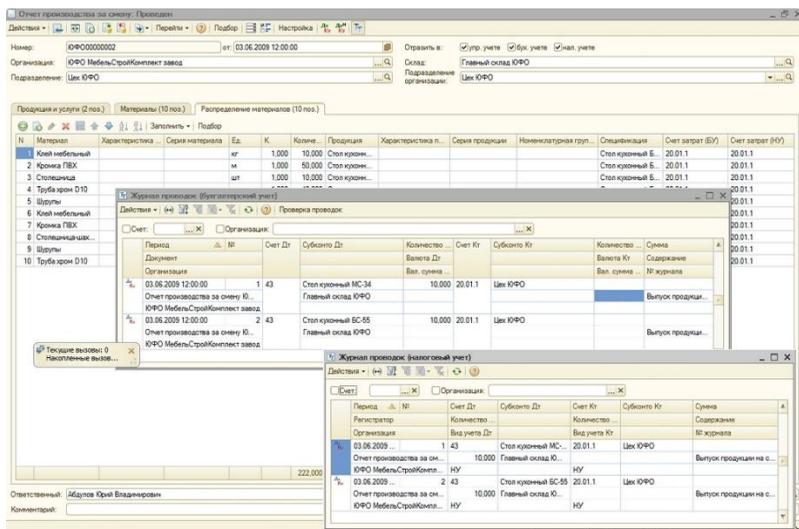


Рис. 61. Программа фирмы «1С»

Более подробное описание возможностей данного программного продукта, а также пример разработки фрагмента корпоративной информационной системы средствами технологической платформы фирмы «1С» представлено в учебно-методическом пособии «Корпоративные информационные системы» [39].

### Галактика ERP

Галактика ERP – это полномасштабная управленческая информационная система, разработанная специально для крупных и средних предприятий России и стран СНГ. Отличием от западных систем является отсутствие избыточного функционала [45].

Система является ядром комплекса бизнес-решений Галактика Business Suite, предназначенного выполнять в едином информационном пространстве типовые и специализированные задачи управления предприятием, холдингом, группой компаний в условиях современной экономики.

Основные функциональные возможности:

- построение системы учета и формирование различных видов отчетности;
- управление материальными и финансовыми потоками (логистика);
- финансовое планирование и оперативный финансовый менеджмент, управленческий учет;
- производственное планирование и управление производством, контроллинг;
- управление персоналом и кадровой политикой;
- управление техническим обслуживанием и ремонтами оборудования;

- управление качеством продукции;
- управление взаимоотношениями с клиентами;
- управление недвижимостью.

Пример интерфейса программы представлена на рис. 62.

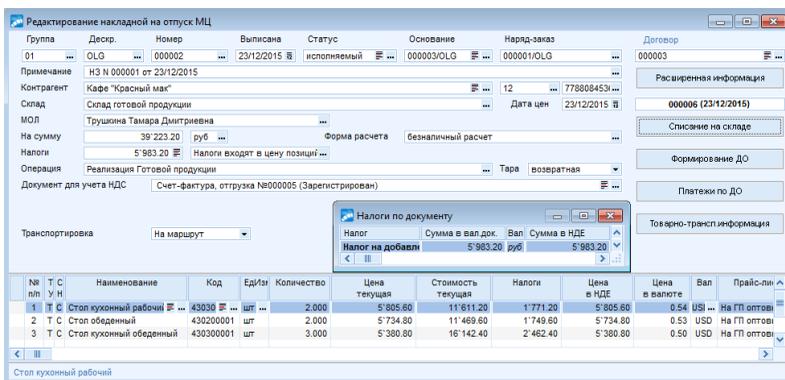


Рис. 62. Интерфейс программы Галактика

## Система Ericor iScala

Система управления предприятием Ericor iScala предназначена для подразделений и зависимых предприятий корпораций международного уровня, расположенных в разных странах, и для больших компаний местного или регионального уровня, для которых характерно наличие большого числа операций между подразделениями. В нее входят интегрированные системы управления ресурсами предприятия, снабжением, производством, обслуживанием на месте установки, проектами, фондом заработной платы и пр.

Как указывают разработчики, Ericor iScala предназначена для подразделений и филиалов международных

корпораций по всему миру, а также для крупных местных или региональных компаний, имеющих значительный торговый оборот между странами. Интерфейс программного продукта представлен на рис. 63.

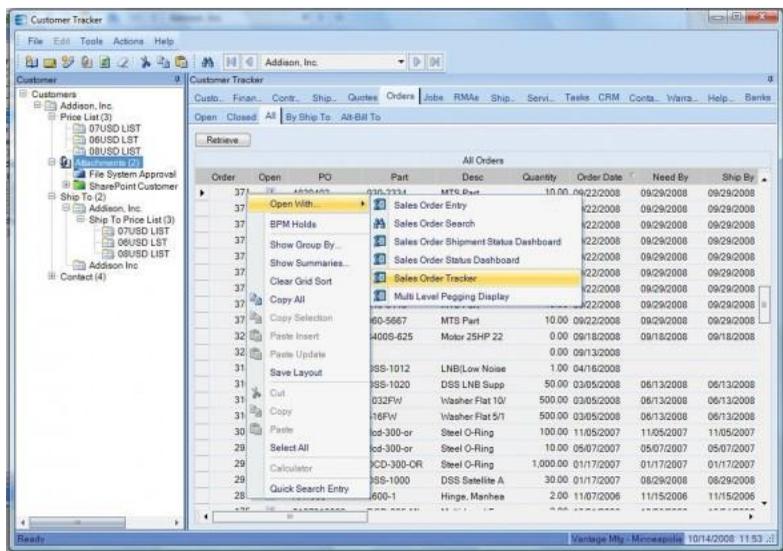


Рис. 63. Интерфейс Epicor iScala

## Microsoft Dynamics

Компания Microsoft предлагает два решения:

- Microsoft Dynamics NAV – многофункциональная система управления ресурсами предприятия для малого и среднего бизнеса.
- Microsoft Dynamics AX – это бизнес-решение для глобальных компаний, которые поддерживают отраслевые и операционные бизнес-процессы и нуждаются в полноценных функциональных возможностях ERP для управления финансами и персоналом.

Microsoft Dynamics NAV – это глобальное решение для планирования ресурсов предприятия (ERP), позволяющее компаниям малого и среднего размера лучше управлять своими финансами, а также упростить логистическую цепочку, производственные и рабочие процессы. Оно быстро внедряется и удобно в использовании, а также обеспечивает полноценную поддержку растущей компании [44].

Интерфейс программы представлен на рисунке 64.



Рис. 64. Программа Microsoft Dynamics NAV

Microsoft Dynamics AX объединяет с помощью своей платформы различные отделы предприятия, позволяя решать задачи компании, принимать более эффективные решения, быстрее преобразовывать бизнес-процессы. Поддерживает гибридную облачную среду.

Внешний вид программы представлен на рис. 65.

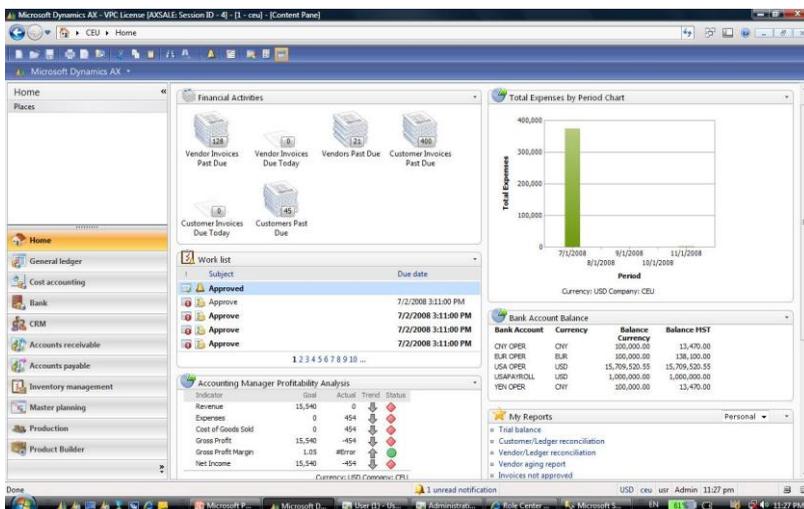


Рис. 65. Программа Microsoft Dynamics AX

## SAP ERP

Компания SAP разрабатывает целое семейство решений, предлагающее широкую функциональность, полную интеграцию, неограниченную масштабируемость и простое взаимодействие в рамках сетевых инфраструктур ведения бизнеса.

Программы SAP Business Suite включают различные направления:

- Управление ресурсами предприятия – SAP ERP.
- Управление взаимоотношениями с клиентами – SAP Customer Relationship Management.
- Управление жизненным циклом продукта – SAP Product Lifecycle Management.
- Управление взаимоотношениями с поставщиками – SAP Supplier Relationship Management.

- Управление логистической сетью – SAP Supply Chain Management и другие.

Пользовательский интерфейс программы SAP представлен на рис. 66.

Icon	Icon name	Comment	Length	Printabl	Internal
	ICON_SYSTEM_COMP_MENU	Enterprise menu	2		68 F_COMM
	ICON_SPACE	Blank	2		5F B_SPCE
	ICON_SYSTEM_USER_MENU	User menu	2		6C F_USRM
	ICON_SYSTEM_FAVORITES	User favorites	2		6D F_FAVO
	ICON_MESSAGE_INFORMATION	Information message	4		19 M_INFO
	ICON_MESSAGE_WARNING	Warning	4		1A M_WARN
	ICON_MESSAGE_ERROR	Error message	4		1B M_ERRO
	ICON_MESSAGE_QUESTION	Question	4		1C M_QUES
	ICON_MESSAGE_CRITICAL	Critical message	4		1D M_CRIT
	ICON_POLICY	Police data view / attachment	2		JK POLICY
	ICON_PAYMENT	Payment view / definition	2		JL PAYMEN

Рис. 66. Программы SAP

### Контрольные вопросы

1. Что такое корпоративная информационная система? Дайте определения с точки зрения различных подходов.

2. Как понимается корпоративная информационная система с точки зрения стандартов качества?

3. Какие классы программных продуктов предполагает внедрение ИСО 9000 на предприятии?

4. Перечислите требования к корпоративным информационным системам.

5. Дайте пояснения требованиям к корпоративным информационным системам.
6. Назовите требования, которые выдвигают корпоративные информационные системы относительно предприятий, на которых они внедряются.
7. Каковы подходы к определению архитектуры корпоративных информационных систем?
8. Перечислите и поясните уровни третьего подхода к архитектуре корпоративных информационных систем.
9. Выделите признаки классификации корпоративных информационных систем.
10. Назовите классы корпоративных информационных систем при классификации их по функциональным возможностям.
11. Какие типы корпоративных информационных систем существуют?
12. Чем обусловлено появление именно таких типов корпоративных информационных систем?
13. Каковы особенности корпоративных информационных систем стандарта MRP?
14. Опишите логику работы компьютерной программы, реализующей MRP-систему.
15. Каковы особенности корпоративных информационных систем стандарта MRP II?
16. Являются ли системы класса MRP и MRP II разными типами корпоративных информационных систем? Ответ поясните.
17. Каковы особенности корпоративных информационных систем стандарта ERP?
18. Чем отличаются системы стандарта ERP от MRP II?

19. Каковы особенности корпоративных информационных систем стандарта CRM?

20. Перечислите основания для классификации CRM-систем.

21. Каковы особенности корпоративных информационных систем стандарта CSRP?

22. Каковы особенности корпоративных информационных систем стандарта ERP II?

23. В чем принципиальные отличия систем класса ERP и ERP II?

24. Приведите примеры корпоративных информационных систем различных типов.

### **Задания для самостоятельного выполнения**

1. Для каждого типа корпоративных информационных систем заполните следующую таблицу (см. табл. 7).

Таблица 7

### **Сравнение типов корпоративных информационных систем**

№ п/п	Тип	Время появления	Основные особенности	Отличия от предыдущего типа	Есть ли методология	Есть ли стандарт	Примеры программных продуктов
1	MRP						
2	MRP II						
3	ERP						
4	ERP II						

2. Выберите корпоративную информационную систему и представьте подробный анализ по схеме:

- Название программы.
- Разработчик.
- Тип корпоративной информационной системы.
- Перечень функциональных возможностей.
- Технические требования программы.
- Print Screen'ы окон программы.
- Примерная стоимость.
- Оценка сроков внедрения.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. ERP. Обзор TAdviser [Электронный ресурс]. – Режим доступа: <http://www.tadviser.ru/index.php/ERP>.
2. Microsoft Access – реляционная СУБД [Электронный ресурс]. – Режим доступа: <https://products.office.com/ru-ru/access>.
3. Абдеев, Р.Ф. Философия информационной цивилизации / Р.Ф. Абдеев. – М.: ВЛАДОС, 1994. – 336 с.: ил.
4. Аккоф, Р. От данных к мудрости [Электронный ресурс] / Р. Аккоф. – Режим доступа: <http://www.nsu.ru/xmlui/bitstream/handle/nsu/9064/01.pdf>.
5. Анисимов, В.В. Продукционная модель представления знаний [Электронный ресурс] / В.В. Анисимов. – Режим доступа: <https://sites.google.com/site/anisimovkhv/learning/knowledge/lecture/tema4>.
6. Анисимов, В.В. Фреймы. Фреймовая модель представления знаний [Электронный ресурс] / В.В. Анисимов. – Режим доступа: <https://sites.google.com/site/anisimovkhv/learning/knowledge/lecture/tema6>.
7. Архитектура информационных систем [Текст]: учеб. для студ. учреждений высш. проф. образования / Б.Я. Советов, А.И. Водяхо, В.А. Дубенецкий, В.В. Цехановский. – М.: Издательский центр «Академия», 2012. – 288 с. – (Сер. Бакалавриат).
8. Боканов, А.А. Понятие информации в современной экономической науке / А.А. Боканов // Вестник Военного университета. – 2010. – №1 (21). – С. 120–126.

9. Васильев, А.А. Теория нечетких множеств [Электронный ресурс] / А.А. Васильев. – Режим доступа: <http://nurtakar.chat.ru/fuzzy.htm>.
10. Винер, Н. Кибернетика или Управление и связь в животном и машине / Н. Винер. – М.: Советское радио, 1968. – 201 с.
11. Вывод в логических моделях. Метод правил вывода [Электронный ресурс]. – Режим доступа: <http://www.aiportal.ru/articles/knowledge-models/modus-ponens.html>.
12. Гаврилова, Т.А. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский. – СПб.: Питер, 2000. – 384 с.].
13. Герасимов, А.С. Курс математической логики и теории вычислимости: учебное пособие / А.С. Герасимов. – СПб.: Издательство «ЛЕМА», 2011. – 284 с.
14. Готт, В.С. Социальная роль информатики / В.С. Готт, Э.П. Семенюк, А.Д. Урсул. – М.: Знание, 1987. – 241 с.
15. Гусева, Е.Н. Информатика [Текст]: учеб. пособие / Е.Н. Гусева, И.Ю. Ефимова, Р.И. Коробков, К.В. Коробкова, И.Н. Мовчан, Л.А. Савельева. – М.: ФЛИНТА, 2011. – 260 с.
16. Дейт, К.Дж. Введение в системы баз данных [Текст] / К.Дж. Дейт. – М.: Издательский дом «Вильямс», 2016. – 1 328 с.
17. Дубровин, А.Д. Интеллектуальные информационные системы: учеб. пособ. для студ. фак-та МИСИТ Московского государственного университета культуры и искусств / А.Д. Дубровин. – М.: МГУКИ, 2008. – Ч. 1.
18. Иванов, В. Продукционная модель экспертных систем [Электронный ресурс] / В. Иванов. – Режим доступа:

<https://libtime.ru/expertsystems/produkcionnaya-model-ekspertnyh-sistem.html>.

19. Избачков, Ю.С. Информационные системы [Текст]: учеб. для вузов / Ю.С. Избачков, В.Н. Петров, А.А. Васильев, И.С. Телина. – СПб.: Питер, 2011. – 544 с.: ил.
20. Информатика [Текст]: учеб. / под ред. проф. Н.В. Макаровой. – М.: Финансы и статистика, 2000. – 768 с.: ил.
21. Информатика. Систематический курс [Текст]: учеб. для 11 класса гуманитарного профиля / С.А. Бешенков, Н.В. Кузьмина, Е.А. Ракитина. – М.: Бином. Лаборатория Знаний, 2007. – 198 с.
22. Искусственный интеллект: в 3-х кн. Кн. 2. Модели и методы: Справочник / под ред. Д.А. Поспелова – М.: Радио и связь, 1990. – 304 с.
23. Каширин, Д.И. Модели представления знаний в системах искусственного интеллекта / Д.И. Каширин, И.Ю. Каширин // Вестник РГРТУ. – 2010. – №31. – С. 36–43.
24. Кузнецов, С.Д. Объектно-ориентированные базы данных – основные концепции, организация и управление: краткий обзор [Электронный ресурс] / С.Д. Кузнецов. – Режим доступа: [http://citforum.ru/database/articles/art\\_24.shtml](http://citforum.ru/database/articles/art_24.shtml).
25. Куликовский, Л.Ф. Теоретические основы информационных процессов / Л.Ф. Куликовский, В.В. Мотов. – М.: Высш. школа, 1987. – 248 с.
26. Лебедева, Т.Н. Корпоративные информационные системы: учеб. пособие // Т.Н. Лебедева, Л.С. Носова. – Челябинск: Центр оперативной полиграфии «Violit-print», 2013. – 83 с.

27. Лебедева, Т.Н. Функциональное и логическое программирование. Ч. 1. Пролог: учеб. пособие для студ. высш. учеб. заведений / Т.Н. Лебедева. – Челябинск: ООО «Полиграф-Мастер», 2008. – 134 с.
28. Лебедева, Т.Н. Функционально-логическое программирование. Язык Пролог: учеб.-метод. пособие для студ. высших учебных заведений / ЧОУ ВПО Юж.-Урал. ин-т управления и экономики; Т.Н. Лебедева. – Челябинск: Центр оперативной полиграфии «Violit-print», 2014. – 110 с.
29. Лидовский, В.В. Теория информации: учебное пособие / В.В. Лидовский. – М.: Компания Спутник+, 2004. – 111 с.
30. Логическая модель представления знаний [Электронный ресурс]. – Режим доступа: <http://www.aiportal.ru/articles/knowledge-models/logical-model.html>.
31. Логические модели [Электронный ресурс]. – Режим доступа: [http://info-tehnologii.ru/predstav\\_znan/model/log\\_model/index.html](http://info-tehnologii.ru/predstav_znan/model/log_model/index.html).
32. Макарова, Н.В. Информатика [Текст]: учеб. для вузов / Н.В. Макарова, В.Б. Волков. – СПб.: Питер, 2015. – 576 с.: ил.
33. Малязина, А.А. Основные категории теории фреймов в трактовке И. Гофмана // Электронный научный журнал «APRIORI. СЕРИЯ: ГУМАНИТАРНЫЕ НАУКИ» WWW.APRIORI-JOURNAL.RU – 2015. – № 3.
34. Матвейкин, В.Г. Информационный менеджмент: курс лекций / В.Г. Матвейкин, Б.С. Дмитриевский, К.А. Садов. – Тамбов: Изд-во Тамб. гос. техн. ун-та, 2009. – 44 с.
35. Минский, М. Фреймы для представления знаний / М. Минский. – М.: Энергия, 1979. – 151 с.

36. Моисеев, Н.Н. Современный рационализм / Н.Н. Моисеев. – М.: МГВП КОКС, 1995. – 376 с.
37. Молдованова, О.В. Информационные системы и базы данных [Электронный ресурс]: учеб. пособие / О.В. Молдованова. – Электрон. текстовые данные. – Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2014. – 178 с. – Режим доступа: <http://www.iprbookshop.ru/45470>. – ЭБС «IPRbooks», по паролю.
38. Николаев, А.Б. Интеллектуальные системы: учебное пособие / А.Б. Николаев, А.В. Остроух. – М.: МАДИ, 2012. – 271 с., ил.
39. Носова, Л.С. Корпоративные информационные системы: учеб.-метод. пособие // Л.С. Носова. – Челябинск: Центр оперативной полиграфии «Violitprint», 2014. – 126 с.
40. Ожегов, С.И. Толковый словарь русского языка / С.И. Ожегов, Н.Ю. Шведова. – М.: Азъ, 1992. – 892 с.
41. Основные направления в области искусственного интеллекта [Электронный ресурс]. – Режим доступа: <http://www.aiportal.ru/articles/other/direction-researches.html>.
42. Официальный сайт 1С [Электронный ресурс]. – Режим доступа: <http://www.1c.ru/>.
43. Официальный сайт HansaWorld [Электронный ресурс]. – Режим доступа: <http://www.hansaworld.com/products/enterprise>.
44. Официальный сайт Microsoft Dynamics NAV [Электронный ресурс]. – Режим доступа: <https://www.microsoft.com/ru-ru/dynamics/erp-nav-overview.aspx>.

45. Официальный сайт Галактика ERP [Электронный ресурс]. – Режим доступа: <http://galaktika-express.ru/galaktika-erp-opisanie-avtomatizirovannoj-sistemy-upravleniya.html>.
46. Официальный сайт КОМПАС [Электронный ресурс]. – Режим доступа: <http://www.compas.ru/about/about.php>.
47. Официальный сайт ПАРУС [Электронный ресурс]. – Режим доступа: <http://www.parus.com/solutions/corporate/system/348/>.
48. Персианов, В.В. Информационные системы [Текст]: учеб.-метод. пособие / В.В. Персианов, Е.И. Логвинова. – М.–Берлин: Директ-Медия, 2016. – 191 с.
49. Представление и использование знаний / под ред. Х. Уэно, М. Исидзука. – М.: Мир, 1989. – 220 с.
50. Психологический словарь / под общей науч. ред. П.С. Гуревича. – М.: ОЛМА Медиа Групп, ОЛМА ПРЕСС Образование, 2007. – 800 с.
51. Разумовский, О.С. Инварианты и фреймы как объективные феномены и когнитивные конструкции [Электронный ресурс]. – Режим доступа: [www.philosophy.nsc.ru/journals/philscience/800/05Razum.htm](http://www.philosophy.nsc.ru/journals/philscience/800/05Razum.htm).
52. Рузаков, А.А. Управление данными [Текст]: учеб. пособие / А.А. Рузаков. – Челябинск: Изд-во Челяб. гос. пед. ун-та, 2015. – 132 с.
53. Саак, А.Э. Информационные технологии управления [Текст]: учеб. для вузов / А.Э. Саак, Е.В. Пахомов, В.Н. Тюшняков. – СПб.: Питер, 2013. – 320 с.
54. Семакин, И. Г. Информатика и ИКТ. Базовый уровень: учеб. для 10–11 классов / И.Г. Семакин, Е.К. Хеннер. – М.: БИНОМ. Лаборатория знаний, 2012.

55. Словарь иностранных слов. – М.: ЛОКИД, 2005. – 657 с.
56. Смолин, Д.В. Введение в искусственный интеллект: конспект лекций. – М.: ФИЗМАТЛИТ, 2004. – 208 с.
57. Толковый словарь по искусственному интеллекту / авторы-составители: А.Н. Аверкин, М.Г. Гаазе-Рапопорт, Д.А. Поспелов. – М.: Радио и связь, 1992.
58. Урсул, А.Д. Информация. Методологические аспекты / А.Д. Урсул. – М.: Наука, 1971. – 296 с.
59. Федеральный закон «Об информации, информационных технологиях и о защите информации» [Электронный ресурс]. – Режим доступа: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_165971/?frame=1](http://www.consultant.ru/document/cons_doc_LAW_165971/?frame=1).
60. Фреймовая модель представления знаний [Электронный ресурс]. – Режим доступа: <http://itteach.ru/predstavlenie-znaniy/freymovaya-model-predstavleniya-znaniy>.
61. Хабаров, С.П. 6. Продукции. 6.1 Продукционные модели [Электронный ресурс]. – Режим доступа: <http://www.habarov.spb.ru/bz/bz06.htm>.
62. Шефер, О.Р. Инновационные технологии визуализации данных в обучении [Текст] / О.Р. Шефер, Н.В. Лапикова, Т.Н. Лебедева, Л.С. Носова // Дистанционное и виртуальное обучение. – 2017. – № 2(116). – С. 4-11.

Учебное издание

**Лебедева Татьяна Николаевна  
Носова Людмила Сергеевна  
Рузаков Андрей Александрович**

**ИНФОРМАЦИОННЫЕ СИСТЕМЫ И БАЗЫ ЗНАНИЙ**  
Учебно-методическое пособие

ISBN 978-5-906908-60-5

Работа рекомендована РИСом ЮУрГППУ  
Протокол № 3/12 (пункт 6), 2017 г.

Редактор Е.М. Сапегина  
Технический редактор А.Г. Петрова  
Эксперт Н.В. Гилязева

Издательство ЮУрГППУ  
454080, г. Челябинск, пр. Ленина, 69

---

Объем 5,3 уч.-изд. л	Тираж 100 экз.
Формат 60x84/16	Подписано в печать 05.05.2017
Бумага типографская	Заказ №

Отпечатано с готового оригинал-макета  
в типографии ЮУрГППУ  
454080, г. Челябинск, пр. Ленина, 69