





**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНО-  
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»**  
(ФГБОУ ВО «ЮУрГГПУ»)

**ФИЗИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ  
КАФЕДРА ИНФОРМАТИКИ, ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И  
МЕТОДИКИ ОБУЧЕНИЯ ИНФОРМАТИКЕ**

**Разработка автоматизированной системы «Конструктор практико-ориентированных  
задач по программированию»  
Выпускная квалификационная работа  
по направлению 09.03.02 Информационные системы и технологии  
Направленность программы бакалавриата  
«Информационные технологии в образовании»**

Проверка на объем заимствований:  
\_\_\_\_\_ % авторского текста

Работа \_\_\_\_\_ к защите  
рекомендована/не рекомендована

« \_\_\_ » \_\_\_\_\_ 20\_\_ г.  
зав. кафедрой И, ИТ и МОИ

\_\_\_\_\_ Рузаков А.А.

Выполнила:  
Студентка группы ОФ-413/095-4-1  
Смирнова Марина Игоревна

Научный руководитель:  
Магистр педагогического образования,  
старший преподаватель кафедры И, ИТ и  
МОИ ЮУрГГПУ

\_\_\_\_\_ Боровская Елена Владимировна

**Челябинск  
2019**

## Содержание

Введение.....	4
Глава 1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И НАЗНАЧЕНИЕ ПРОЕКТИРУЕМОЙ СИСТЕМЫ .....	6
1.1 Характеристика объекта автоматизации.....	6
1.1.1 Описание МАОУ «Лицей №67 г. Челябинска» .....	6
1.1.2 Описание структуры организации заказчика системы и основных бизнес-процессов.....	8
1.1.3 Место решаемой проблемы.....	11
1.1.4 Описание средств автоматизации работы МАОУ «Лицей №67 г. Челябинска» .....	13
1.2 Описание проектируемой системы.....	17
1.2.1 Назначение системы .....	17
1.2.2 Цели создания системы .....	17
1.2.3 Требования к структуре и функционированию системы.....	17
1.2.4 Требования к функциям, выполняемым системой .....	18
1.3 Анализ аналогичных программных продуктов .....	19
1.3.1 Web-портал «Генератор практических заданий».....	19
1.3.2 Web-портал «Генератор примеров и словарных слов» .....	21
1.3.3 Web-портал «ЯКласс».....	22
Вывод по главе 1 .....	25
ГЛАВА 2. ПРОЕКТИРОВАНИЕ .....	26
2.1 Выбор инструментальных средств создания программного продукта и определение архитектуры информационной среды.....	26
2.2 Модель данных .....	28
2.3 Описание подхода к реализации основных функций .....	32
Вывод по главе 2 .....	42
Глава 3. Документирование готового программного продукта .....	43
3.1 Испытание системы.....	43
3.2.1 Аннотация .....	45
3.2.2 Область применения .....	46
3.2.3 Краткое описание возможностей.....	46
3.2.4 Уровень подготовки пользователя .....	46
3.2.5 Приложение к разрабатываемой ИС .....	46
3.2.6 Список обозначений и сокращений .....	47

3.2.7 Назначение .....	47
3.2.8 Условия применения.....	47
3.2.9 Состав и содержание носителя данных .....	48
3.2.10 Порядок загрузки данных и программ.....	48
3.2.11 Главное окно .....	48
3.2.12 Генератор .....	49
3.2.13 Источник .....	50
3.2.14 Тема.....	51
3.2.15 Тип задачи.....	52
3.2.16 Содержание.....	53
3.2.17 Критерии задачи.....	54
3.2.18 Задача.....	55
3.2.19 Переход на нужную страницу.....	56
3.2.20 Задачи .....	57
3.2.21 Просмотр задачи.....	58
3.2.22 Справочники .....	58
3.3 Техничко-экономическое обоснование программного продукта.....	60
Вывод по главе 3 .....	64
Заключение .....	65
Библиографический список .....	67

## Введение

Для человека крайне важна способность применять обобщенные знания и умения для решения конкретных ситуаций и проблем, возникающих в реальной действительности. По мнению известных психологов и методистов, формировать способность решения проблем помогают специальным образом подобранные задачи. Такие задачи называются практико-ориентированными.

Практико-ориентированные задачи – это задачи из окружающей действительности, связанные с формированием практических навыков, необходимых в повседневной жизни.

Практика показывает, что ученики с интересом решают и воспринимают задачи практического содержания. Они с увлечением наблюдают, как из практической задачи возникает теоретическая и как чисто теоретической задаче можно придать практическую форму.

Цель этих задач – формирование умений действовать в социально-значимой ситуации. Они базируются на знаниях и умениях, но требуют умения применять накопленные знания в практической деятельности. Назначением практико-ориентированных задач является возможность «окунуть» в решение «жизненной» задачи. Важными отличиями практико-ориентированных задач от стандартных являются:

- значимость получаемого результата, что обеспечивает познавательную мотивацию учащегося;
- условие задачи сформулировано как сюжет, ситуация или проблема, для ее решения необходимо использовать знания из различных разделов основного предмета, из другого предмета или из жизни, на которые не явного указания в тексте задачи;
- информация и данные могут быть представлены в различной форме, такие как рисунки, таблицы, схемы и т.д., что потребует распознавания объектов;
- указание области применения результата, полученного при решении задачи.

Для удобства построения таких задач используются генераторы задач. Анализ существующего ПО показал, что такие генераторы применяются в области математики и на данный момент нет доступных генераторов для задач по программированию. Но программирование требует решения практико-ориентированных задач, для закрепления материала. Таким образом были сформированы следующие цели и задачи.

**Цель работы:** разработать автоматизированную информационную систему «Конструктор практико-ориентированных задач по программированию»

**Для достижения поставленной цели необходимо решить следующие задачи:**

- описать структуру и бизнес–процессы организации-заказчика;
- выявить объект автоматизации;
- подготовить и утвердить техническое задание;
- провести анализ аналогичных программных продуктов;
- спроектировать и реализовать систему;
- внедрить разрабатываемую информационную систему в МАОУ «Лицей № 67 г. Челябинска»;
- провести функциональные и нефункциональные тесты;
- разработать и прописать руководство пользователя;
- выполнить технико-экономическое обоснование разработки системы.

**Практическая значимость:** разработанная программа повысит эффективность труда учителя информатики за счет снижения времени на составление контрольных работ.

# **Глава 1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И НАЗНАЧЕНИЕ ПРОЕКТИРУЕМОЙ СИСТЕМЫ**

## **1.1 Характеристика объекта автоматизации**

### **1.1.1 Описание МАОУ «Лицей №67 г. Челябинска»**

Общая информация:

- Местоположение: 454080, г. Челябинск, ул. Сони Кривой, д.40,
- Контактный телефон: +7(351)263-86-86,
- E-mail: chelmou-67@yandex.ru
- График работы: с 8-30 до 17-00 (с понедельника по субботу)
- Учредителем является МАОУ «Лицей № 67 г. Челябинска» от

имени муниципального образования «Город Челябинск» является Администрация города Челябинска в лице Комитета по делам образования города Челябинска.

- Юридический адрес учредителя: 454080, г. Челябинск, ул. Володарского, 14

В своей деятельности школа руководствуется действующим законодательством Российской Федерации, муниципальными правовыми актами города Челябинска и уполномоченных должностных лиц Администрации города Челябинска, принятыми в пределах их компетенции, установленной муниципальными правовыми актами Администрации города Челябинска, Уставом школы.

Образовательная организация осуществляет образовательный процесс в соответствии с уровнями общеобразовательных программ: начальное (1-4 классы), основное (5-9 классы), среднее (10-11 классы) общее образование.

Лицей № 67 использует и совершенствует методики образовательного процесса и образовательных технологий, в том числе дистанционных образовательных технологий. Школа разрабатывает и утверждает образовательные программы и учебные планы в соответствии с федеральным государственным образовательным стандартом общего образования.

Нормативный срок обучения осуществляется так же в соответствии с уровнями общеобразовательных программ: 4, 5 и 2 года соответственно.

Форма обучения: Дневная, очная.

Лицей осуществляет следующие виды деятельности:

1) образовательная деятельность по основным общеобразовательным программам – образование дошкольное, образование начальное общее, образование основное общее, образование среднее общее;

2) образовательная деятельность по дополнительным общеобразовательным программам – образование дополнительное детей и взрослых;

3) образовательная деятельность по программам профессионального обучения – обучение профессиональное;

4) коррекционно-развивающая, компенсирующая и логопедическая помощь обучающимся;

5) психолого-педагогическое консультирование обучающихся, их родителей (законных представителей) и педагогических работников - ОКВЭД;

6) научно-методическое обеспечение образовательной деятельности

7) деятельность предприятий общественного питания по прочим видам организации питания – предоставление питания;

8) деятельность полиграфическая и копирование носителей информации;

9) торговля розничная прочая вне магазинов, палаток, рынков;

10) деятельность издательская;

11) деятельность по обработке данных, предоставление услуг по размещению информации, деятельность порталов в информационно-коммуникационной сети Интернет;

12) аренда и управление собственным или арендованным нежилым недвижимым имуществом– предоставление в аренду и (или) безвозмездное пользование муниципального имущества, находящегося в оперативном управлении МАОУ «Лицей № 67 г. Челябинска»;



13) прокат и аренда предметов личного пользования и хозяйственно-бытового назначения;

14) аренда и лизинг прочих машин и оборудования и материальных средств;

15) аренда интеллектуальной собственности и подобной продукции, кроме авторских прав;

16) деятельность в области медицины прочая, первичная медико-санитарная помощь, включенная в базовую программу обязательного медицинского страхования;

17) предоставление прочих социальных услуг без обеспечения проживания – присмотр и уход за детьми, организация отдыха детей;

18) деятельность творческая, деятельность в области искусства и организации развлечений, организация проведения общественно-значимых мероприятий в сфере образования, науки и молодежной политики;

19) деятельность библиотек, архивов, музеев и прочих объектов культуры;

20) деятельность в области спорта, отдыха и развлечений, пропаганда физической культуры, спорта и здорового образа жизни;

21) деятельность профессиональных союзов;

22) деятельность по предоставлению прочих персональных услуг.

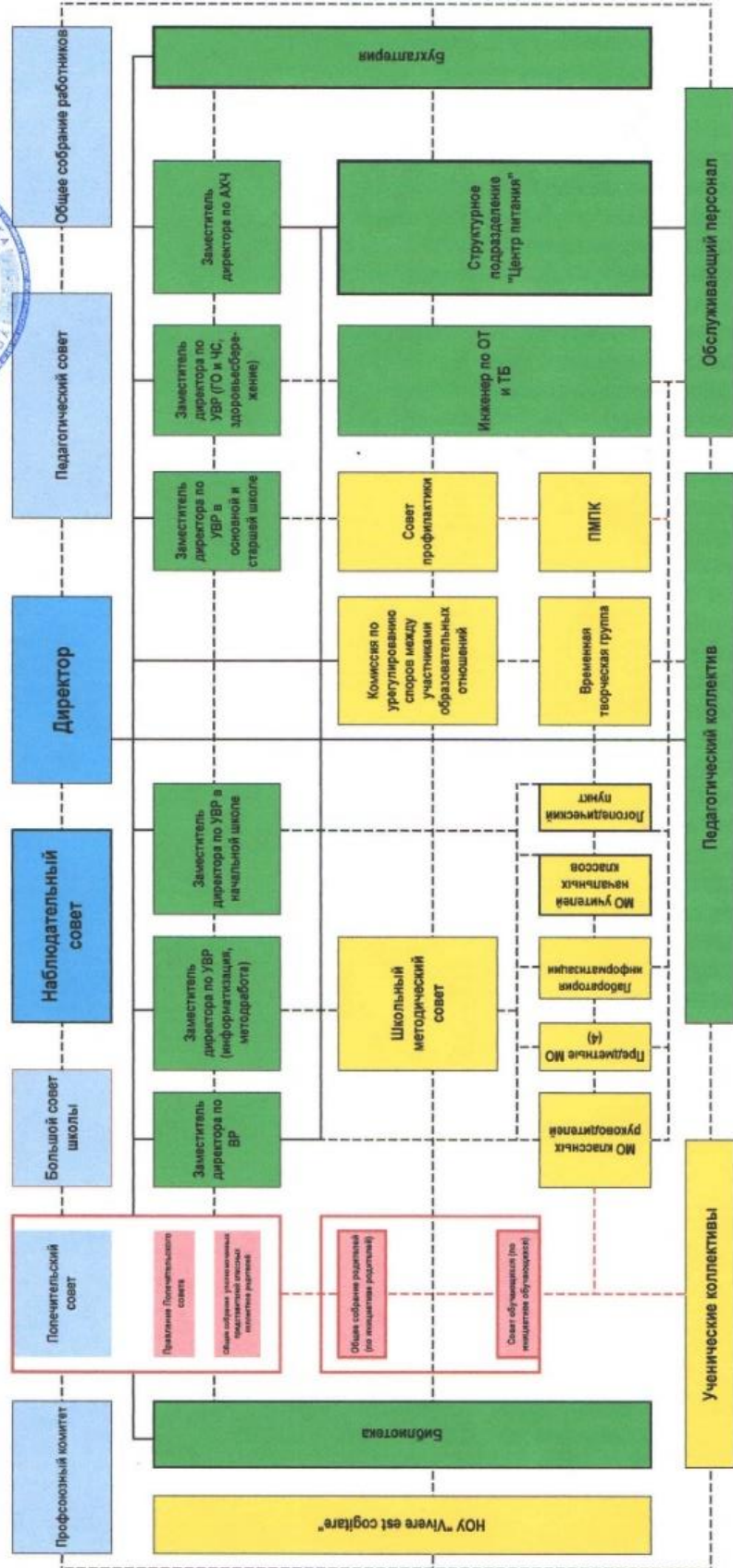
### **1.1.2 Описание структуры организации заказчика системы и основных бизнес-процессов**

Организационная структура управления - совокупность специализированных функциональных подразделений, взаимосвязанных в процессе обоснования, выработки, принятия и реализации управленческих решений.

В данном случае организационная структура, именно управления, Муниципального автономного общеобразовательного учреждения «Лицей № 67 г. Челябинска» (Рисунок 1).



**ОРГАНИЗАЦИОННАЯ СТРУКТУРА УПРАВЛЕНИЯ  
МАОУ "СОШ № 67 г. Челябинска"**



— прямое подчинение  
- - - функциональное подчинение

Рис. 1. Организационная структура управления МАОУ «Лицей №67 г. Челябинска».

«Органы государственно-общественного управления» состоит из:

- Наблюдательный совет;
- Большой Совет;
- Попечительский совет.

Органы самоуправления предприятия:

- Педагогический совет;
- Общее собрание трудового коллектива.

В него входят такие люди, как директор, бухгалтер, заведующие и заместители.

Орган государственно-общественного управления и орган самоуправления являются основой структуры органов управления МАОУ Лицей №67.

На базе лицея существуют следующие лаборатории:

- Лаборатория физики;
- Лаборатория информатики;
- Лаборатория химии;
- Лаборатория биологии.

На базе предметного методического совета осуществляются практические занятия по предмету информатика. Для предоставления практических знаний учитель разрабатывает большое количество задач. Для оптимального сочетания теоретического и практического материала, демонстраций возможностей IT технологий в бытовой, повседневной и профессиональной деятельности применяют практико-ориентированные задачи. Использование данного типа задач производится с целью формирования умений действовать в ситуациях, встречающихся в повседневной жизни. Научить учащихся работать с информацией, то есть добывать, объяснять, отобрать, критически оценить, найти собственное

решение, развить свои точки зрения, чувства, убеждения и желания в поисковой деятельности учащихся. Практико-ориентированные задачи являются элементом «живых» задач, в содержании понятия которых интегрированы так же и понятия учебная задача и познавательная задача. При этом «живая» задача должна иметь личностно-значимый характер, входить в сферу потребностей личного опыта учащегося, реконструировать личный опыт, обогащая его новым, более глубоким знанием.

Создание практико-ориентированных задач требует большого количества однотипных действий. Что занимает время учителя и замедляет процесс обучения. Для автоматизации процесса составления задач существуют генераторы. Сейчас представлено большое количество генераторов для математиков, но нет генераторов для учителей информатики.

### **1.1.3 Место решаемой проблемы**

Для описания функциональной модели формирования контрольных работ была выбрана методология IDEF0 так как она является наиболее удобной для отображения бизнес-процессов. Методология IDEF0 нашла широкое применение благодаря простому отображению информации. Главным компонентом модели являются диаграммы. На них отображаются функции системы в виде прямоугольников, а также связи между ними и внешней средой посредством стрелок. Использование такого малого количества графических примитив позволяют быстро объяснить правила и принципы построения диаграмм IDEF0 людям, незнакомым с данной методологией. Это достоинство позволяет подключить и активизировать деятельность заказчика по описанию бизнес-процессов с использованием формального и наглядного графического языка.

Контекстная диаграмма с описанием процессов формирования контрольных работ представлена на рисунке 2.

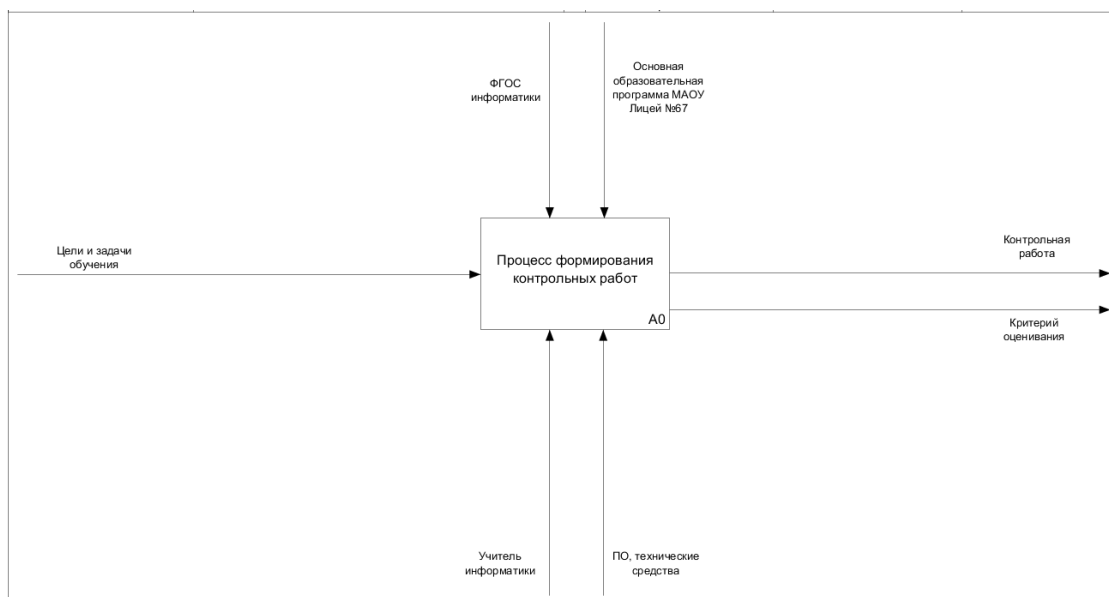


Рис. 2 Бизнес-процесс формирования контрольных работ

Далее, на рисунке, представлена декомпозиция контекстной диаграммы.

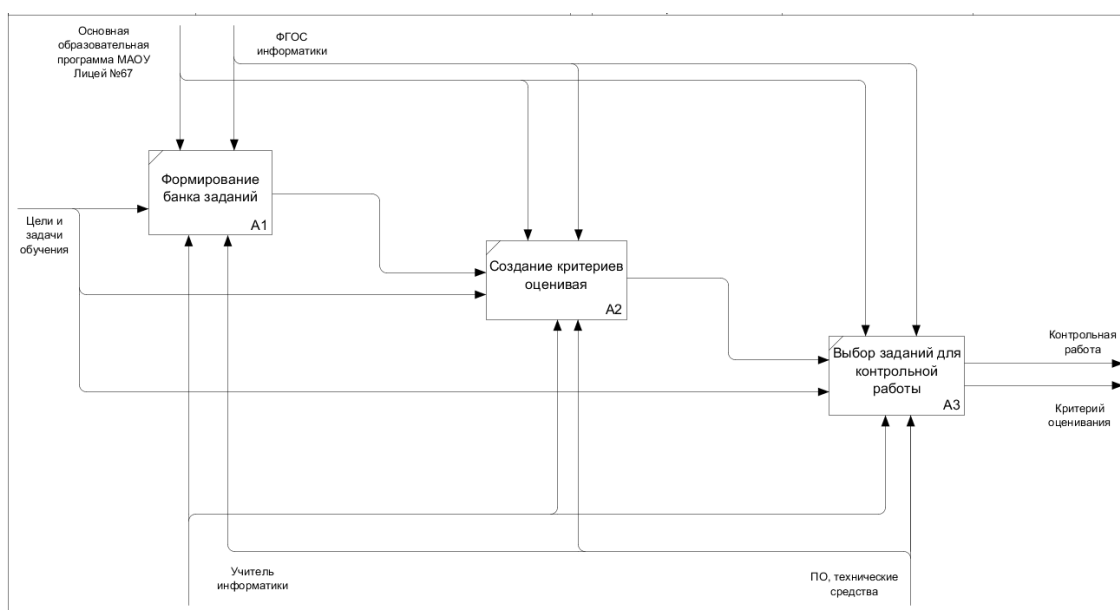


Рис. 3 Декомпозиция контекстной диаграммы

После составления проблем данного проекта, были сформированы следующие требования для системы «Конструктор практико-ориентированных задач по программированию»:

- возможность пополнения базы данных;
- возможность генерации практико-ориентированных задач;

- возможность выгрузить сформированную контрольную в документ Word.

#### **1.1.4 Описание средств автоматизации работы МАОУ «Лицей №67 г. Челябинска»**

Компьютерная сеть – сеть взаимосвязанных ПК для обмена информации, коллективного использования ресурсов: аппаратных, программных, и информационных (баз данных).

Локальная сеть лицея №67 – это сложный организационно-технологический комплекс, созданный для взаимодействия информационных ресурсов школы. Оборудована локальная сеть посредством коммутаторов и сервера. Так же она включает в себя средства защиты информации, доступ к глобальным сетям, и средства работы (пк, рабочие станции и т.д.).

Учреждение постоянно закупает новые компьютеры. Ученики занимаются как на ПК, так и на ноутбуках, моноблоках, интерактивной доске.

В Учреждении имеется порядка 100 рабочих компьютеров. Все они работают в соответствии с нормами, техникой безопасности, и уставе о лицензионном обеспечении. В учреждении имеется компьютерный класс, с кондиционером и вентиляцией.

Основные характеристики компьютеров в компьютерном классе:

Моноблоки (10 экземпляров).

- Процессор Intel(R) Celeron(R) CPU N2840 2.16GHz
- Оперативная память 2гб
- Встроенное графическое ядро Intel Graphics
- ОС Windows 7

Персональные компьютеры (10 экземпляров+1 учительский).

- Процессор Intel(R) Celeron(R) CPU B430 1.80GHz
- Оперативная память 3гб
- Встроенное графическое ядро Intel Graphics
- ОС Windows 7

В компьютерном классе, помимо ПК и моноблоков имеются ноутбуки. Они аналогично персональным компьютерам находятся в аудитории и служат для работы учеников во время занятий.

- Процессор Intel(R) Celeron(R) CPU B800 1.50GHz
- Оперативная память 2гб
- Встроенное графическое ядро Intel(R) HD Graphics Family
- ОС Windows 7

Информационная организация компьютеров школы имеет стандартный способ защиты. Реализованная система имеет 2х пользователей. При включении компьютера пользователю предлагается выбрать пользователя, всего их два: «Учитель» и «Ученик». Пользователь «учитель» защищён паролем, который знают лишь преподаватели. Таким образом «ученикам» закрывается доступ к ПО, не относящемуся к учебному процессу.

Базовый перечень лицензионного программного обеспечения, используемого в школе.

Тип программного обеспечения	Используемое ПО
Операционная система	Microsoft Windows или эквивалент по функциональным характеристикам (7, 8, 8.1, 10)
Офисные приложения	Microsoft Office, MS front Page
Антивирусное программное обеспечение	AVP Kaspersky и аналогичные по функциональности бесплатные продукты
ПО для файловых архивов	Стандартный проводник от Windows, а также Total Commander
Языки программирования	Pascal abc, Scratch, Исполнители

Все эти компьютеры лицея, и остальные ПК в школе подключены в локальную сеть. В лицее реализована поддержка сети Wi-Fi сетей в компьютерном и соседнем классах.

Структура сети имеет форму «звезды» - базовая топология компьютерной сети, в которой все компьютеры сети подключены к центральному узлу, образуя физический сегмент сети. Весь обмен информацией идет исключительно через центральный компьютер, на который таким способом возлагается большая нагрузка, поэтому ничем другим, кроме сети, он заниматься не может. Центральный компьютер является самым мощным, и именно на него возлагаются все функции по управлению сетью. Конфликты в сети с топологией звезды невозможны в принципе, так как управление полностью централизовано. Краткая схема локальной сети представлена ниже (Рис. 4).

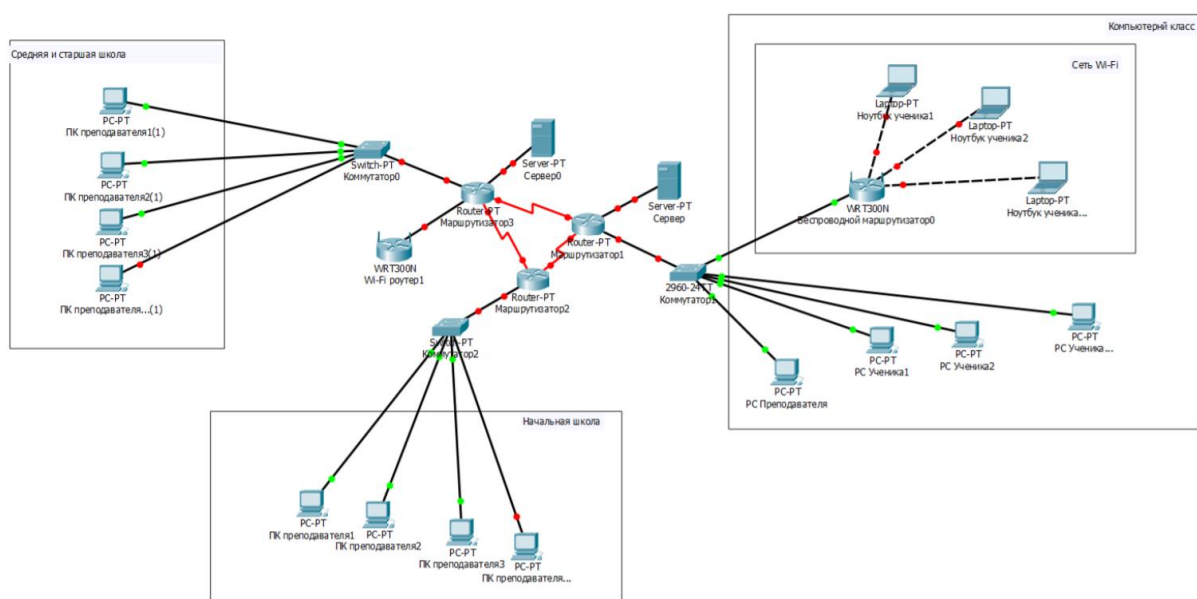


Рис. 4. Локальная сеть MAOU «Лицей №67 г. Челябинска».

В учреждении установлена Система Криптографической Защиты Информации (далее СКЗИ). СКЗИ-это ПО или комплекс, с помощью которого происходит шифрование данных и передача их по сети интернет. Благодаря этому комплексу, данные, которые хранятся в блоке СКЗИ, не могут быть скорректированы и удалены без расшифрования. Производство и работа с этим модулем находится под контролем ФСБ РФ.



Сам блок СКЗИ – это небольшой корпусный элемент, размещающийся на материнской плате. Он имеет интерфейсный выход и разъём для антенны (работа под сетью ГЛОНАСС).

Основные функции СКЗИ:

- работа с цифровой подписью;
- защита от корректировки защищённых (записанных) данных;
- согласование записанных данных с временным штампом, что исключает корректировку времени записи;
- шифрование данных;
- идентификация подписи, которая применяется к выдаче зашифрованных данных.

В школе отсутствует централизованный сервер, что приводит к отсутствию централизованного управления доступа к информации, таким как материалы уроков, задания для практических работ и т.п.

Сайт школы <http://chel67.ru> написан на WordPress и уже устарел. У WordPress есть свои достоинства, в виде простоты и удобства использования, но также множество недостатков:

- достаточно весома́я нагрузка на сервер;
- порою́ низкая скорость загрузки сайта;
- ограниченный набор базовых функций;
- дублирование страниц и изображений;
- большое количество некачественных тем и дополнений с ошибками;
- относительное недоверие поисковых систем к сайтам на бесплатных движках.

## **1.2 Описание проектируемой системы**

### **1.2.1 Назначение системы**

Автоматизированная информационная система «Конструктор практико-ориентированных задач по программированию» (далее – ИС) должна иметь следующий вид: десктопное приложение, предназначенное для:

- хранения данных для создания практико-ориентированных задач;
- автоматизации работы по формированию контрольных работ по информатике;
- хранения готовых практико-ориентированных задач.

### **1.2.2 Цели создания системы**

Основными целями создания системы являются:

- автоматизировать работу учителя информатики в формировании контрольных работ;
- уменьшение возникновения ошибок при составлении задач и контрольных работ.

### **1.2.3 Требования к структуре и функционированию системы**

ИС создается как десктопное приложение, реализуется на персональном компьютере. Приложение рассчитано на одного пользователя. Система должна быстро работать и иметь отказоустойчивость.

Система должна отвечать всем современным требованиям, иметь лаконичный и удобный интерфейс.

Система должна быть рассчитана на функционирование в операционной системе минимум Microsoft Windows 7, имеющей объем ОЗУ равный или более 2 ГБ, процессором с тактовой частотой 1,5 ГГц или большей. Рекомендуется использовать операционную систему Microsoft Windows 10, процессор с частотой 2 ГГц или выше и объем ОЗУ больше 2 ГБ.

Целью создания приложения является возможность повысить эффективность труда учителя информатики за счет снижения времени на составление контрольных работ.

#### 1.2.4 Требования к функциям, выполняемым системой

Согласно поставленной задаче, разрабатываемая система должна состоять из следующих подсистем, разделенных по функциональному признаку:

- подсистема хранения данных;
- подсистема генерации задач;
- подсистема формирования контрольных работ.

Для подсистемы хранения данных входными данными является текст из БД, выходными – измененный текст из БД. Для подсистемы генерации задач входными данными так же является текст из БД, выходными – готовая к выводу информация. Для подсистемы формирования контрольных работ входными данными является текст из БД, выходными – документ Word.

Основной функцией является генерация практико-ориентированных задач.

Структура практико-ориентированной задачи включает:

- стимул;
- задачу формулировку;
- источник информации;
- инструмент проверки.

Стимул моделирует практико-ориентированную жизненную ситуацию, мотивируя при этом учащегося на выполнение задания.

Задачная формулировка – конкретный вопрос или указание на ту деятельность, которую учащийся должен совершить. Каждый вопрос, включенный в состав задачной формулировки, должен быть нацелен на оценку достижения конкретных планируемых результатов.

Источник информации – содержит информацию, необходимую для успешной деятельности учащегося по выполнению задания.

Инструмент проверки – определяет количество баллов за каждый этап деятельности и общий итог в зависимости от сложности учебного материала, дополнительных видов деятельности.

Процесс создания практико-ориентированного задания можно разбить на несколько последовательных этапов:

- выбор дидактических единиц, планируемых результатов освоения учебной программы и универсальных учебных действий для оценки;
- создание мотивирующей ситуации в соответствии с выбранными дидактическими единицами;
- создание задачной формулировки с включением источника информации и формы отчета в соответствии с проверяемыми планируемыми результатами и конкретными действиями;
- уточнение задачной формулировки в соответствии с проверяемыми универсальными учебными действиями;
- подготовка инструмента проверки и шкалы оценивания.

Практико-ориентированные задачи предполагают возможность создания задачи по одному стимулу, так как к одному стимулу может относиться несколько источников. Программный продукт должен учитывать это.

Практико-ориентированные задачи включают в себя большое количество данных, для удобства создания задач, приложение должно иметь такой программный инструмент, как мастер. Мастер помогает пользователю быстро и наглядно осуществить нужные настройки приложения. В нашем случае, поэтапный выбор данных для генерации задачи поможет пользователю быстро выбрать нужные данные для задачи.

Содержание и критерии оценивания имеют стандартный вид для каждого типа и темы программы. Программный продукт должен хранить шаблоны содержаний и критериев оценивания, которые пользователь сможет изменить при создании задачи.

### **1.3 Анализ аналогичных программных продуктов**

#### **1.3.1 Web-портал «Генератор практических заданий»**

Этот генератор практических заданий позволяет создавать и сразу распечатывать задания для детей разного возраста по математике, чтению и

письму по заданным параметрам. На рисунке 5 представлен пример генерации заданий на данном ресурсе.

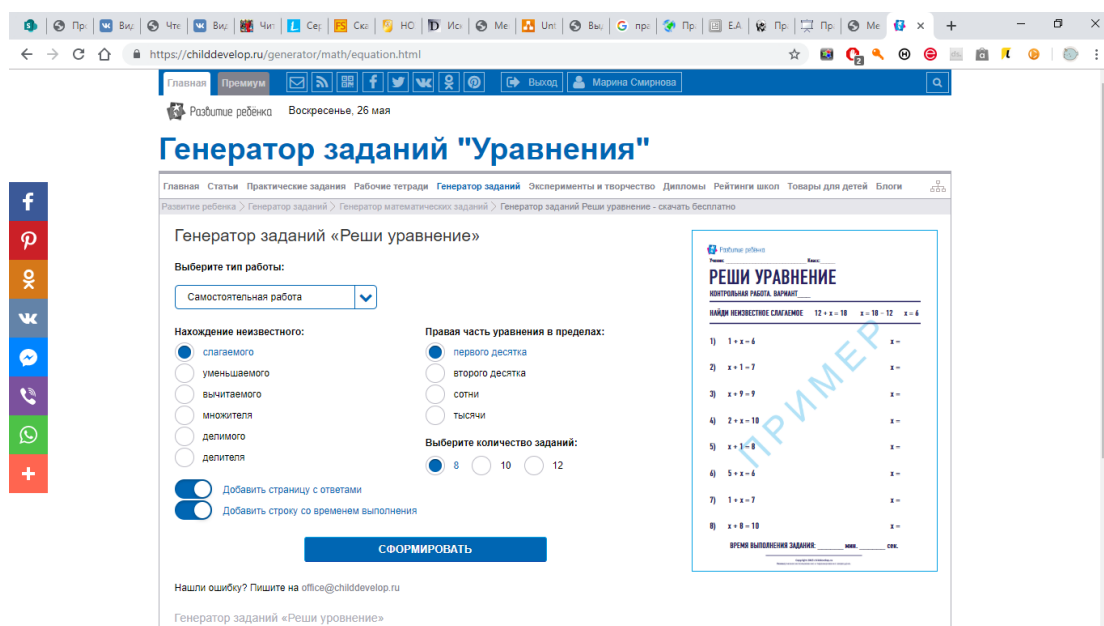


Рис. 5 Генератор практических заданий

Генератор практических заданий создан для:

- 1) Учителя дошкольных и начальных классов для которых. Они могут создать задания для ребенка и не беспокоится, что ответы на них он спишет в решебнике.
- 2) Родителям, которые хотят, чтоб их дети лучше закрепляли полученные знания. Очень пригодится, если они решили отдать своего ребенка в школу с углубленным изучением предметов.
- 3) Учителям средних и старших классов. Для учителей генератор практических заданий будет настоящей находкой и каждый урок будет разнообразным и интересным для детей

На портале представлено множество видов генераторов.

Возможности web-портала:

- Выбрать тип работы;
- Выбрать тему работы;
- Выбрать количество заданий;
- Обозначить время выполнения;
- Добавление страницы с ответами.

При формировании контрольной работы дается возможность загрузить PDF файл с примерами.

Недостатки

- Нет возможности пополнить БД
- Нет генерации задач по программированию

### 1.3.2 Web-портал «Генератор примеров и словарных слов»

Вы можете сгенерировать примеры любой сложности, а затем распечатать их или решать в интерактивном режиме. Ниже на рисунке 6 представлен пример генерации проверочной работы на данном ресурсе.

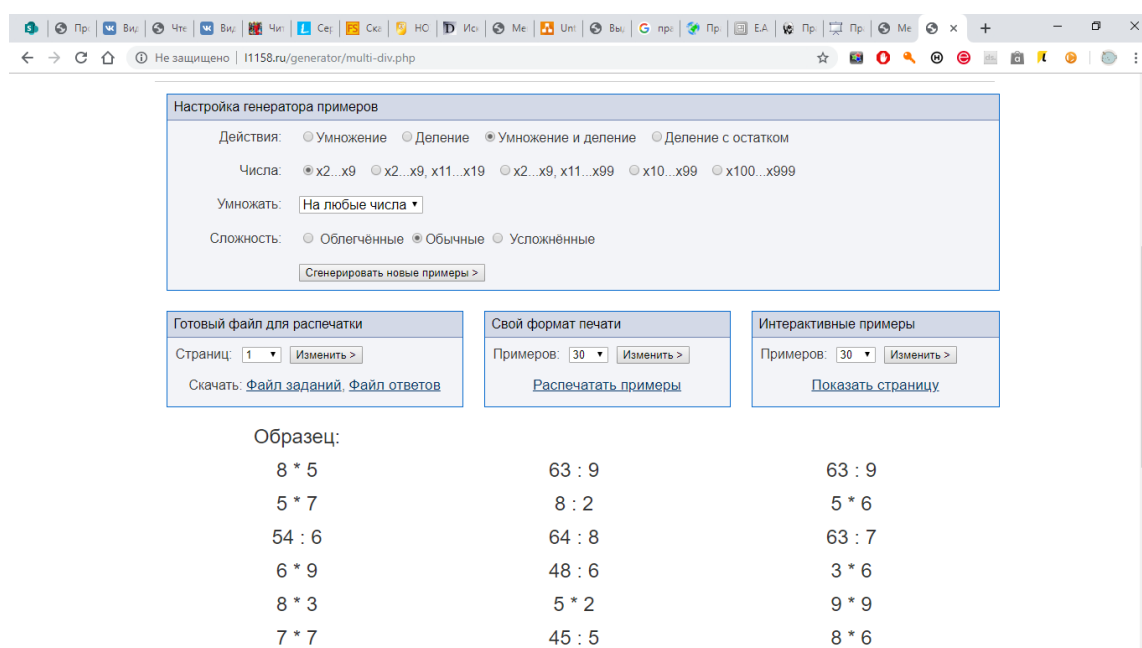


Рис. 6 Генератор примеров и словарных слов

Возможности данного ресурса:

- Готовый файл для распечатки - файл для Word'a, который можно распечатать.
- Есть вариант примеров на умножение и деление для ученика - с местами для ответов, и вариант для родителей - с указанными ответами.
- Задайте количество страниц, нажмите «изменить», и скачивайте готовый файл.
- Свой формат печати - для тех, кто хочет скопировать примеры в другой файл, или перенести в другой формат.

- Можно установить свой шрифт, количество примеров, количество столбиков.
- Есть вариант примеров на умножение и деление для ученика - с местами для ответов, и вариант для родителей - с указанными ответами.
- Интерактивные примеры - для тех, кто занимается с планшетом, смартфоном, компьютером, и другим устройством подключённым к интернету. Задайте нужный шрифт, количество примеров и столбцов и проверяйте себя on-line. Если нажать на пример - на экране вместо него появится результат, и ученик сразу сможет проверить себя.

Недостатки:

- Нет возможности пополнить БД
- Нет генерации задач по программированию

### **1.3.3 Web-портал «ЯКласс»**

ЯКласс — образовательный онлайн-ресурс, начавший работу в 2013 году. Технология сайта позволяет проводить электронные тестирования и генерировать задания, уникальные для каждого ученика. Ниже на рисунке 7 представлен пример создания проверочной работы на данном ресурсе.

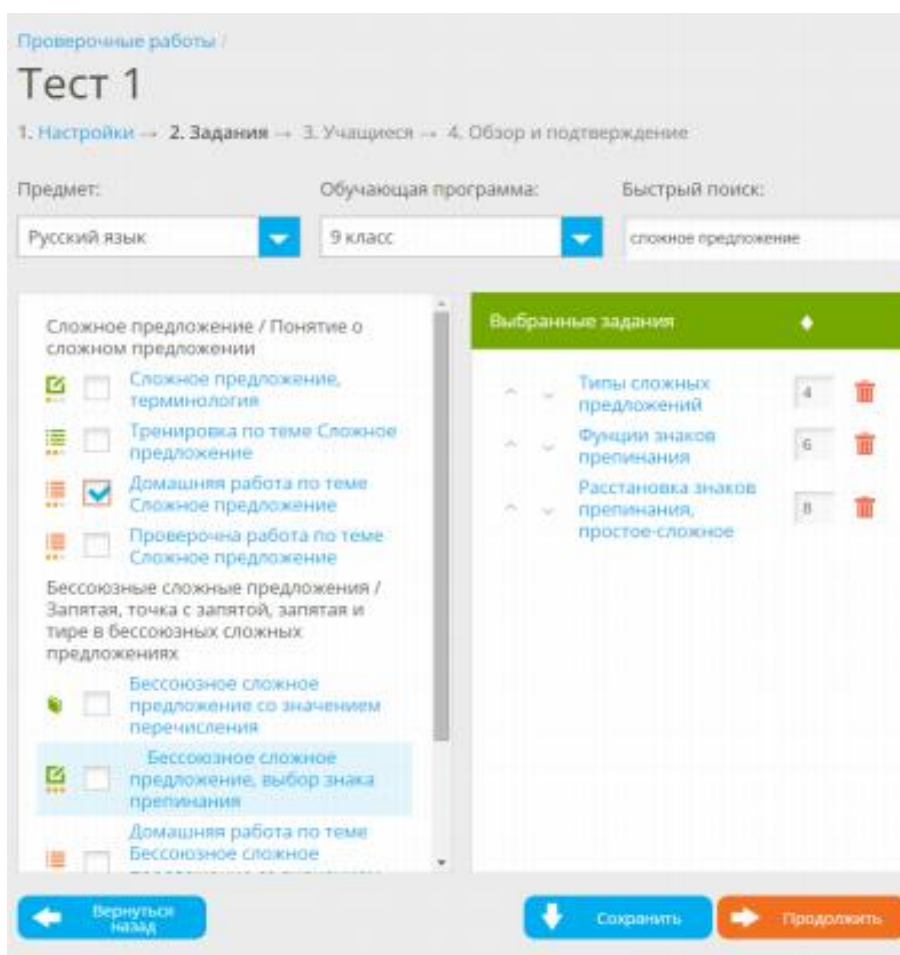


Рис. 7 ЯКласс

Возможности ресурса ЯКласс:

- выбор готовых материалов из базы ЯКласс;
- возможность фильтровать по предметам, обучающей программе и с помощью быстрого поиска;
- создать и добавить в работу собственное задание;
- проверочную работу учитель может задать учащимся, зарегистрированным на ЯКласс;
- распечатать задания для учеников, у которых нет доступа к компьютеру;
- исправляйте задания или отправляйте комментарии.

Недостатки:

- Учитель обязан быть зарегистрирован в системе ЯКласс;



- Использование услуги формирования проверочных работ платная, месячная оплата стоит 299 рублей.

Анализ аналогичных программных продуктов показал, что сейчас в открытом доступе множество генераторов, но большая часть представлена для определенных предметов. Чаще всего пользователь не может изменять базу данных генераторов. Генераторы с возможностью добавления своих задач являются платными, что подходит не каждому пользователю.

В основном генераторы предназначены для заданий по математике, их невозможно перестроить под свои нужды, например для задач по программированию. Программирование в свою очередь требует решения множества задач. Но все их приходится создавать собственноручно. Ручная подготовка практико-ориентированных задач представляет сложность в первую очередь из-за объема необходимой однообразной работы и возникающих при этом ошибок и опечаток.

Существующие генераторы можно назвать образцом для создания собственного приложения. Схожий функционал «завязанный» на специфике практико-ориентированных задач можно реализовать в собственной разработке.

## **Вывод по главе 1**

В главе 1 была рассмотрена организационная структура МАОУ «Лицей №67 г. Челябинска», а также описан будущий функционал «конструктора практико-ориентированных задач по программированию». Подведены основные задачи проекта, и пред распределён план работы.

Была описана и проанализирована характеристика объекта автоматизации, проведен комплексный анализ представления данных. Учитывая рабочее пространство объекта автоматизации были учтены все возможные поправки системы.

Описаны цели создания информационной системы. Приложение «Конструктор практико-ориентированных задач по программированию» создается с целью повысить эффективность труда учителя информатики за счет снижения времени на составление контрольных работ.

Выделены подсистемы: формирование банка заданий, создание критериев оценивания, выбор заданий для контрольной работы.

## ГЛАВА 2. ПРОЕКТИРОВАНИЕ

### 2.1 Выбор инструментальных средств создания программного продукта и определение архитектуры информационной среды

Для разработки автоматизированной системы «Конструктор практико-ориентированных задач по программированию» был выбран объектно-ориентированный язык программирования C#.

C# позволяет быстро и легко разрабатывать приложения. При создании C#, его авторы учитывали достижения многих языков программирования, таких как: C++, C, Java и другие. Благодаря тому, что язык разрабатывался с нуля, создателям удалось избавиться от неудобных и неприятных особенностей любого из предшествующих языков. C# является простым, удобным и современным языком, по мощности не уступающим C++, но существенно повышающий продуктивность разработок [1].

В качестве платформы для разработки пользовательского интерфейса была выбрана платформа Windows Presentation Foundation (WPF).

Windows Presentation Foundation – это графическая система в составе .Net Framework 3.0 и более поздних версий. Спроектирована под влиянием технологий HTML и Flash и использует аппаратное ускорение.

WPF сочетает в себе удобство верстки страниц и окон, благодаря собственному языку верстки XAML [3]. В основе WPF лежит мощная инфраструктура, которая основана на DirectX – API-интерфейсе графики с аппаратным ускорением, который используется в современных компьютерных играх. Это дает возможность применения развитых графических эффектов, не платя за это производительностью.

Вместо рисования пикселей в WPF вы имеете дело с примитивами — базовыми фигурами, блоками текста и прочими графическими ингредиентами. Кроме того, доступны такие новые средства, как действительно прозрачные элементы управления, возможность укладывания друг на друга множества уровней с разной степенью прозрачности.

В качестве системы управления базами данных была выбрана SQLite. SQLite - это легко встраиваемая в приложения база данных. SQLite базируется на файлах, она предоставляет довольно широкий набор инструментов для работы с ней, по сравнению с сетевыми СУБД. При работе с SQLite обращения происходят напрямую к файлам, вместо портов и сокетов в сетевых СУБД [7]. Поэтому SQLite очень быстрая и мощная благодаря технологиям обслуживающих библиотек.

Движок и интерфейс базы находятся в пределах одной библиотеки, все данные можно хранить в одном файле. SQLite считается одной из самых быстрых БД. Используя высоко эффективную инфраструктуру, SQLite может работать в крошечном объеме выделяемой для неё памяти, гораздо меньшем, чем в любых других системах БД. Это делает SQLite очень удобным инструментом с возможностью использования практически в любых задачах возлагаемых на базу данных.

Архитектура информационной системы – концепция, определяющая модель, структуру, выполняемые функции и взаимосвязь компонентов информационной системы.

Данное приложение содержит централизованную архитектуру. При использовании такой технологии БД, СУБД и программа располагаются на одном компьютере (рис. 1). Для такого способа организации не требуется поддержки сети, все сводится к автономной работе [5].

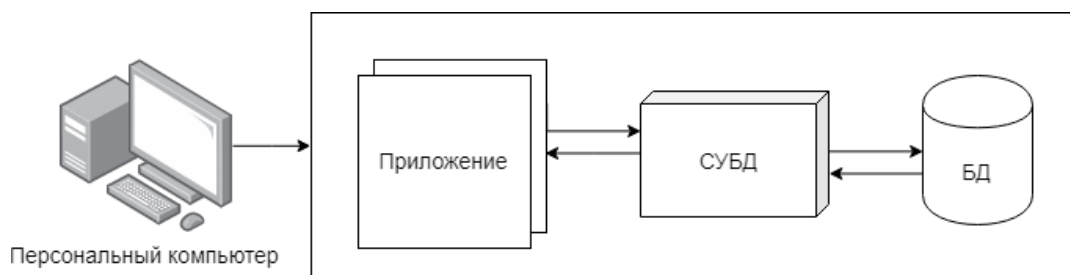


Рис. 1 – Центральная архитектура приложения

Работа такого приложения построена следующим образом:

- база данных в виде файла находится на жестком диске компьютера, на том же компьютере установлены СУБД и приложение для работы с БД;

- пользователь запускает приложение, используя интерфейс приложения, он инициирует обращение к БД на выборку информации;
- обращение к БД идет через СУБД, которая инкапсулирует внутри себя сведения о физической структуре БД, СУБД инициирует обращение к данным, обеспечивая выполнение запросов пользователя, результат возвращает в приложение;
- приложение отображает результат выполнения запросов, используя пользовательский интерфейс.

## 2.2 Модель данных

Для создания приложения нам потребуется база данных, содержащая таблицы с основной информацией, имя базы данных: «Generator». Схема сущностей БД ИС «Конструктор практико-ориентированных задач по программированию» представлена на рисунке 2.

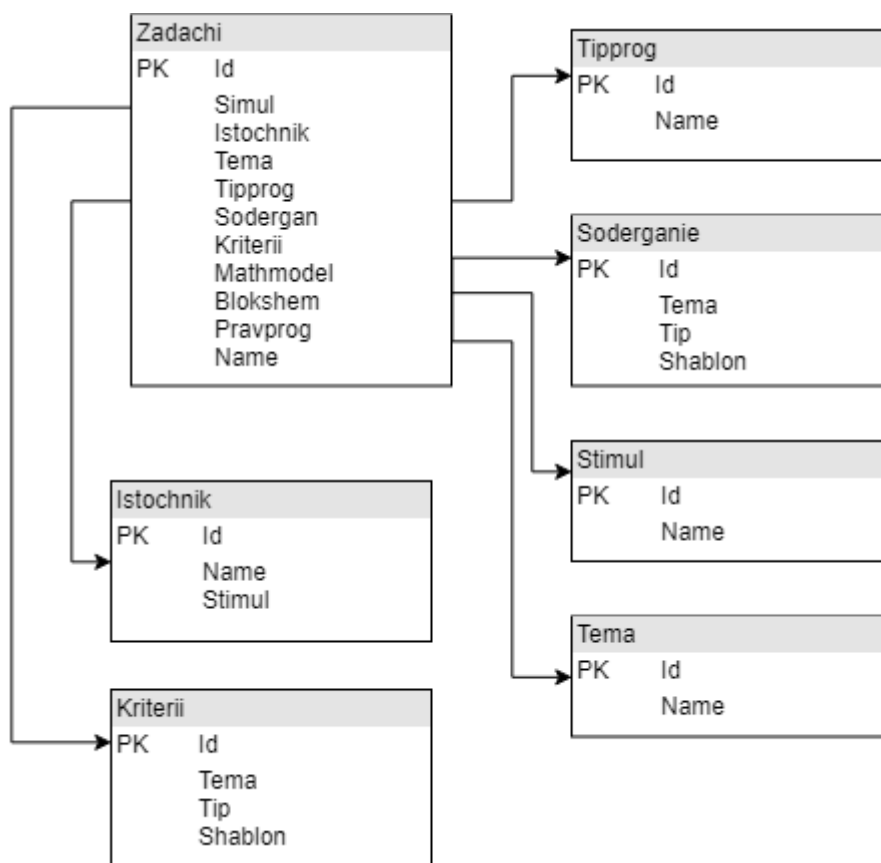


Рис. 2 – Схема сущностей БД ИС «Конструктор практико-ориентированных задач по программированию»

## База данных Generator

Данная база данных предназначена для хранения необходимой для программы информации и содержит семь таблиц.

Для обеспечения нормальной работы БД, устранения избыточности и других аномалий необходимо произвести нормализацию БД и привести ее, к третьей нормальной форме (Н.Ф.).

Переменная отношения находится в Н.Ф. тогда и только тогда, когда для каждой из её функциональных зависимостей  $X \rightarrow A$  выполняется хотя бы одно из следующих условий:

- $X$  содержит  $A$  (то есть  $X \rightarrow A$  — тривиальная функциональная зависимость);
- $X$  — суперключ;
- $A$  — ключевой атрибут (то есть  $A$  входит в состав потенциального ключа).

Проанализировав структуру отношений на предмет выполнения условий нормальных форм, заключим, что все таблицы БД приведены к 3-ей Н.Ф

SQLite поддерживает динамическое типизирование данных. Возможные типы значений: INTEGER, REAL, TEXT и BLOB. Так же поддерживается специальное значение NULL.

Каждое значение в любом поле любой записи может быть любого из этих типов, независимо от типа, указанного при объявлении полей таблицы. Указанный при объявлении поля тип хранится для справки в его исходном написании, используется в качестве основы для выбора предпочтений при выполнении неявных преобразований типов на основании схожести этого названия типа на что-либо, знакомое SQLite.

В таблицах 1.1-1.7 представлены данные о таблицах БД.

Таблица 1.1

Таблица «Zadachi»

Id	INTEGER
----	---------

Stimul	INTEGER
Istochnik	INTEGER
Tema	INTEGER
Tipprog	INTEGER
Soderjan	INTEGER
Kriterii	INTEGER
Mathmodel	TEXT
Blokshem	BLOB
Pravprog	TEXT
Name	TEXT

Таблица 1.2

Таблица «Istochnik»

Id	INTEGER
Name	TEXT
Stimul	INTEGER

Таблица 1.3

Таблица «Kriterii»

Id	INTEGER
Tema	INTEGER
Tip	INTEGER
Shablon	TEXT

Таблица 1.4

Таблица «Soderganie»

Id	INTEGER
Tema	INTEGER
Tip	INTEGER
Shablon	TEXT

Таблица 1.5

Таблица «Stimul»

Id	INTEGER
Name	TEXT

Таблица 1.6

Таблица «Тема»

Id	INTEGER
Name	TEXT

Таблица 1.7

Таблица «Tipprog»

Id	INTEGER
Name	TEXT

Связи между таблицами выполняется посредством внешнего ключа или FOREIGN KEY.

Внешний ключ или FOREIGN KEY – ограничение уровня таблицы в реляционных базах данных, в том числе и в базе данных SQLite. Внешние ключи определяют правила, по которым будут связаны таблицы в базах данных SQLite.

В SQLite внешний ключ объявляется при помощи конструкции FOREIGN KEY, а таблица, на которую ссылается внешний ключ указывается после ключевого слова REFERENC. Указывается таблица и столбец, на который идет ссылка.

В данной базе данных реализована связь «один ко многим» при помощи внешнего ключа. Связь «один ко многим» означает ситуацию, когда строка одной таблицы соответствует нескольким строкам другой таблицы. Это наиболее распространенный тип отношений.



### 2.3 Описание подхода к реализации основных функций

Главными задачами ИС «Конструктор практико-ориентированных задач по программированию» являются генератор задач и возможность формирования контрольных работ из построенных задач.

Генератор было решено выполнить в виде мастера.

Мастер – это инструмент, помогающий пользователю быстро и наглядно осуществить те или иные настройки в компьютерной программе, интерактивная функция в графическом интерфейсе пользователя, которая представляет собой последовательно сменяющиеся друг друга диалоговые окна для выполнения определенной задачи, которую можно разбить на этапы.

В данном программном продукте было решено разделить генерацию задачи на шаги. Было решено отделить выбор каждой компоненты задачи. Для удобства пользователя выбранные компоненты на каждом последующем шаге отображаются в левой части страницы.

Большинство данных представлено в виде списков, выводимых в ListBox. Ниже представлен код, позволяющий выводить данные из БД в ListBox. На примере списка стимулов.

```
void fill_Listbox()
{
    SQLiteConnection sqliteCon = new SQLiteConnection(dbConnectionString);
    try
    {
        sqliteCon.Open();
        string Query = "select * from Stimul ";
        SQLiteCommand createCommand = new SQLiteCommand(Query,
sqliteCon);
        SQLiteDataReader dr = createCommand.ExecuteReader();
        while (dr.Read())
        {
            string Name = dr.GetString(1);
```

```

stimullistbox.Items.Add(Name);
}
sqliteCon.Close();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

```

Источник отображается в видео таблицы данных. В справочники источника мы можем добавить нужный нам источник, привязав его к определенному стимулу. Источник будет записан в текстовый файл. Который в последующем будет выводиться в DataGridView. Сохранение и вывод реализованы по средствам сериализации.

Код сохранения источника представлен ниже.

```

private void Stroki_Click(object sender, RoutedEventArgs e)
{
    int value = rnd.Next();
    nameistochnik.Text = Convert.ToString(value)+".xml";
    XmlSerializer serializer = new
    XmlSerializer(typeof(ObservableCollection<Infistochnik>));
    StreamWriter writer = new StreamWriter(nameistochnik.Text);
    serializer.Serialize(writer, tableistochnik.ItemsSource);
    writer.Close();
    SQLiteConnection sqliteCon = new SQLiteConnection(dbConnectionString);
    try
    {
        sqliteCon.Open();
        string Query = "INSERT INTO Istochnik (Name, Stimul) values
        (""+this.nameistochnik.Text+"","" + this.idstimul.Text + "")";
    }
}

```

```

        SQLiteCommand createCommand = new SQLiteCommand(Query,
sqliteCon);
        createCommand.ExecuteNonQuery();
        MessageBox.Show("Сохранено");
        sqliteCon.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

Так же ниже представлен код вывода данных из текстового файла в Datagrid.

```

        SQLiteConnection sqliteCon = new SQLiteConnection(dbConnectionString);
        try
        {
            sqliteCon.Open();
            string Query = "select * from Istochnik where Stimul=" + this.stimulid.Text
+ """;
            SQLiteCommand createCommand = new SQLiteCommand(Query,
sqliteCon);
            SQLiteDataReader dr = createCommand.ExecuteReader();
            while (dr.Read())
            {
                string Sid = dr.GetInt32(0).ToString();
                string SName = dr.GetString(1);
                istochnikid.Text = Sid;
                istochnikname.Text = SName;
            }
            sqliteCon.Close();

```

```

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    XmlSerializer serializer = new
    XmlSerializer(typeof(ObservableCollection<Infistochnik>));
    StreamReader reader = new StreamReader(istochnikname.Text);
    ObservableCollection<Infistochnik> Infistochniks =
    (ObservableCollection<Infistochnik>)serializer.Deserialize(reader);
    reader.Close();
    this.istochnikdatagrid.ItemsSource = Infistochniks;
    this.datagrid1.ItemsSource = Infistochniks;

```

Страница генератора выводит информацию из БД, по каждому отдельному компоненту задачи. После выбора всех компонентов, оператору дается возможность сохранить полученную задачу в БД. Код, позволяющий сохранить задачу в БД представлен ниже.

```

private void Save_Click(object sender, RoutedEventArgs e)
{
    gridnew.Visibility = Visibility.Visible;
    gridconech.Visibility = Visibility.Hidden;
    var bmp = blockshema.Source as BitmapImage;
    int height = bmp.PixelHeight;
    int width = bmp.PixelWidth;
    int stride = width * ((bmp.Format.BitsPerPixel + 7) / 8);
    byte[] bits = new byte[height * stride];
    bmp.CopyPixels(bits, stride, 0);
    string zadachaname = tip4.Text + " , " + tema4.Text + " , " + soderjanie3.Text;
    SQLiteConnection sqliteCon = new SQLiteConnection(dbConnectionString);
    try

```

```

    {
        sqliteCon.Open();
        string Query = "INSERT INTO Zadachi
(Stimul,Istochnik,Tipprog,Tema,Soderjan,Kriterii,Mathmodel,Blokshem,Pravprog,
Name) values ('" + this.stimulid.Text + "','" + this.istochnikid.Text + "','" +
this.tipid.Text + "','" + this.temaid.Text + "','" + this.soderjanie.Text + "','" +
this.kriter.Text + "','" + this.matemmodell1.Text + "','" + bits + "','" +
this.pravzadach1.Text + "','" + zadachaname + "')";
        SQLiteCommand createCommand = new SQLiteCommand(Query,
sqliteCon);
        createCommand.ExecuteNonQuery();
        sqliteCon.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

На странице с готовыми задачами пользователь может собрать необходимые ему задачи в контрольную работу. Для этого на странице представлен ListBox со списком выводимых задач, который можно отфильтровать по теме и типу задачи. Когда пользователь выберет нужные ему задачи, он может нажать на кнопку «Сформировать». После ему откроется 2 Word документа. Один будет содержать сами задачи, а второй критерии к этим задачам и дополнительные материалы. Код, позволяющий отобразить документы представлен ниже.

```

        private string TemplateFileName =
@"C:\Users\Марина\source\repos\sprav\sprav\bin\Debug\zadac.docx";
        private string TemplateFileName1 =
@"C:\Users\Марина\source\repos\sprav\sprav\bin\Debug\kriter.docx";

```

```

private void Open_Click(object sender, RoutedEventArgs e)
{
    var wordApp = new Word.Application();
    wordApp.Visible = false;
    int z = 1;
    var wordDocument = wordApp.Documents.Open(TemplateFileName);
    var wordDocument1 = wordApp.Documents.Open(TemplateFileName1);
    for (int i = 0; i < zadachilist.SelectedItems.Count/2; i++)
    {
        foreach (object item in zadachilist.SelectedItems)
        {
            SQLiteConnection sqlLiteCon = new SQLiteConnection(dbConnectionString);
            try
            {
                sqlLiteCon.Open();
                string Query = "select * from Zadachi where Name='" + item.ToString() + "'";
                SQLiteCommand createCommand = new SQLiteCommand(Query,
sqlLiteCon);
                SQLiteDataReader dr = createCommand.ExecuteReader();
                while (dr.Read())
                {
                    int SId = dr.GetInt32(0);
                    SStimul1 = dr.GetInt32(1);
                    SIstochnik1 = dr.GetInt32(2);
                    STema1 = dr.GetInt32(3);
                    STipprog1 = dr.GetInt32(4);
                    SSoderganie1 = dr.GetString(5);
                    SKriterii1 = dr.GetString(6);
                    SMathmodell1 = dr.GetString(7);
                    SPravprog1 = dr.GetString(9);
                }
            }
            catch { }
        }
    }
}

```

```

}
dr.Close();
Query = "select * from Stimul where Id=" + SStimul1 + "";
createCommand = new SQLiteCommand(Query, sqliteCon);
dr = createCommand.ExecuteReader();
while (dr.Read())
{
SStimul2 = dr.GetString(1);
}
dr.Close();
Query = "select * from Tema where Id=" + STema1 + "";
createCommand = new SQLiteCommand(Query, sqliteCon);
dr = createCommand.ExecuteReader();
while (dr.Read())
{
STema2 = dr.GetString(1);
}
dr.Close();
Query = "select * from Tipprog where Id=" + STipprog1 + "";
createCommand = new SQLiteCommand(Query, sqliteCon);
dr = createCommand.ExecuteReader();
while (dr.Read())
{
STipprog2 = dr.GetString(1);
}
dr.Close();
Query = "select * from Istochnik where Id=" + SIstochnik + "";
createCommand = new SQLiteCommand(Query, sqliteCon);
dr = createCommand.ExecuteReader();
while (dr.Read())

```

```

    {
        Istochnikname1 = dr.GetString(1);
        XmlSerializer serializer = new
        XmlSerializer(typeof(ObservableCollection<Infistochnik>));
        StreamReader reader = new StreamReader(Istochnikname1);
        ObservableCollection<Infistochnik> Infistochniks =
        (ObservableCollection<Infistochnik>)serializer.Deserialize(reader);
        reader.Close();
        this.datagrid.ItemsSource = Infistochniks;
    }
    sqliteCon.Close();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
try
{
    ReplaceWordStub("{n}", Convert.ToString(z), wordDocument);
    ReplaceWordStub("{tema}", STema2, wordDocument);
    ReplaceWordStub("{tipprog}", STipprog2, wordDocument);
    ReplaceWordStub("{stimul}", SStimul2, wordDocument);
    ReplaceWordStub("{istochnik}", Istochnikname1, wordDocument);
    ReplaceWordStub("{soderganie}", SSoderganie1, wordDocument);
    ReplaceWordStub("{dop}", "Задача {n}\rТема:
{tema}\r{tipprog}\r{stimul}\r{istochnik}\r{soderganie}\r{dop}\r",
wordDocument);
}
catch
{

```



```

MessageBox.Show("Ошибка");
}
try
{
ReplaceWordStub("{n}", Convert.ToString(z), wordDocument1);
ReplaceWordStub("{kriterii}", SKriterii1, wordDocument1);
ReplaceWordStub("{mathmodel}", SMathmodel1, wordDocument1);
ReplaceWordStub("{pravprog}", SPravprog1, wordDocument1);
ReplaceWordStub("{blockshem}", "", wordDocument1);
ReplaceWordStub("{dop1}", "", wordDocument1);
{n}\r{kriterii}\r{mathmodel}\r{pravprog}\r{blockshem}\r{dop1}\r",
wordDocument1);
}
catch
{
MessageBox.Show("Ошибка");
}
z++;
}
ReplaceWordStub("Задача {n}", "", wordDocument);
ReplaceWordStub("Тема: {tema}", "", wordDocument);
ReplaceWordStub("{tipprog}", "", wordDocument);
ReplaceWordStub("{stimul}", "", wordDocument);
ReplaceWordStub("{istochnik}", "", wordDocument);
ReplaceWordStub("{soderhanie}", "", wordDocument);
ReplaceWordStub("{dop}", "", wordDocument);
ReplaceWordStub("Задача {n}", "", wordDocument1);
ReplaceWordStub("{kriterii}", "", wordDocument1);
ReplaceWordStub("{mathmodel}", "", wordDocument1);
ReplaceWordStub("{pravprog}", "", wordDocument1);

```

"Задача

```
ReplaceWordStub("{blockshem}", "", wordDocument1);
ReplaceWordStub("{dop1}", "", wordDocument1);
wordDocument.SaveAs(@"C:\Users\Марина\source\repos\sprav\sprav\bin\
Debug\_zadachi.docx");
wordApp.Visible = true;
wordDocument1.SaveAs(@"C:\Users\Марина\source\repos\sprav\sprav\bin\
Debug\_kriterii.docx");
wordApp.Visible = true;
}
}
```

## **Вывод по главе 2**

Во второй главе были рассмотрены инструментальные средства реализации автоматизированной системы. В качестве языка программирования был выбран язык C#, в качестве платформы для разработки пользовательского интерфейса была выбрана платформа WPF, для хранения данных используется система управления базой данных SQLite.

Представлено описание модели данных, приведены типы данных для каждого поля таблицы.

Рассмотрен код на языке C# с помощью которого выполняются функции данного программного продукта.

## **Глава 3. Документирование готового программного продукта**

### **3.1 Испытание системы**

Испытание информационной системы и тестирование программного продукта очень похожи, но на практике это совсем не так.

Информационная система – это не только используемые в ее составе программные компоненты, но и аппаратное и организационное обеспечение. Необходимо учесть, что каждые программные компоненты входящие в состав системы, не должны конфликтовать с техническим сопровождением предприятия. Аппаратное и организационное обеспечение на предприятии не должно вызывать проблем в работе информационной системы.

Все виды испытаний информационной системы делятся на функциональные и нефункциональные тесты.

Функциональное тестирование призвано показать (доказать), что автоматизированные рабочие места информационной системы предоставляют пользователям ровно ту функциональность, которую они от нее ожидают. Система выполняет свои функции корректно.

Нефункциональное тестирование подтверждает или опровергает соответствие таких свойств информационной системы, как производительность, надежность, эргономичность и т.д. заданным на этапе ее проектирования параметрам. Система выполняет свои функции в срок, в должном объеме и с приемлемым качеством, и пользоваться ею удобно [2].

Виды функционального тестирования:

- Компонентное тестирование
- Интеграционное тестирование
- Тестирование прототипа

Виды нефункционального тестирования:

- Нагрузочное тестирование
- Стрессовое тестирование
- Объемное тестирование

- Тестирование стабильности
- Тестирование надежности
- Тестирование эргономики решений

Такое количество тестов излишне для внедрения АИС «Конструктор практико-ориентированных задач по программированию». При тестировании ПП на предприятии практики, затронуты лишь несколько сфер их вышеуказанной классификации нефункциональных тестов. Функциональное тестирование проведено полностью.

ИС «Конструктор практико-ориентированных задач по программированию».

ПП данной информационной системы разрабатывался под одного пользователя, поэтому внедрять в систему авторизацию не имеет смысла.

Администратором будет являться человек, работающий с данным продуктом, в случае данного проекта это учитель информатики. Он работает с данной системой, как с основным помощником в составлении контрольных и самостоятельных работ.

Тестирование данной системы состоит из нескольких этапов:

- Согласование технических возможностей – минимизация конфликтов, которые могут возникнуть в следствии несоответствия технического обеспечения на персональном компьютере учителя информатики, с требованиями к работе ИС «Конструктор практико-ориентированных задач по программированию».

- Запуск программы – папка с базой данных и файлом расширения «\*.EXE», с виртуального носителя. Возможно, копировать на любое место на жестком диске компьютера. После завершения копирования, необходимо открыть файл с расширение «\*.EXE». Проверить правильность загрузки начальной формы и удостовериться в отсутствии ошибок.

Далее этапы можно условно разбить на поэтапную проверку каждой функции программного продукта. Основных функций три: Генератор задач, Формирование контрольной и Обновление справочников.

Генератор представляет собой мастера, сменяемые между собой страницы, которые перенимают данные с предыдущих страниц. На страницах должны корректно отображаться данные. При сохранении не должно выдавать ошибки. При нажатии на кнопку «Сохранить» должны переходить на страницу выбора дальнейших действий.

На странице формирования контрольных должны корректно отображаться данные. При фильтровании задач, данные со списком задач должны меняться. При выбранных задачах и нажатой кнопке «Сформировать» должны открываться 2 текстовых документа. В них должны корректно отображаться данные выбранных задач.

Страницы со справочниками должны корректно отображать данные, взятые из базы данных. Функции добавления и изменения должны работать без ошибок.

### *3.2 Руководство пользователя*

#### **3.2.1 Аннотация**

Данный документ представляет собой дополнительную документацию к проекту «Конструктор практико-ориентированных задач по программированию». Конкретно данное приложение к проекту является руководством пользователя.

Интерфейс программного продукта (ПП) автоматизированной системы «Конструктор практико-ориентированных задач по программированию» выполнен в простом и лаконичном стиле для минимизации недопонимания функций со стороны оператора.

Пользовательский интерфейс ПП, служит информационной поддержкой деятельности оператора, предоставляет функции.

Документ подготовлен в соответствии с РД 50-34.698-90 - в части структуры и содержания документов, и в соответствии с ГОСТ 34.201-89 – в части наименования и обозначения документов.

### **3.2.2 Область применения**

Программный продукт применяется непосредственно в рамках работы учителя информатики. Выполняет ряд операций связанных с построением задач и контрольных работ.

### **3.2.3 Краткое описание возможностей**

Автоматизированная Информационная Система (АИС) Конструктор практико-ориентированных задач по программированию выполняет следующие функции:

- генерация практико-ориентированных задач по программированию;
- формирование контрольных работ по информатике.

### **3.2.4 Уровень подготовки пользователя**

Для эксплуатации ИС Конструктор практико-ориентированных задач по программированию определены следующие пользователи:

- оператор (он же учитель информатики).

Оператор обязан:

- ознакомиться с руководством пользователя перед эксплуатацией программного продукта;
- иметь общие сведения о система и ее назначении;
- иметь общее представление о интерфейсе программного продукта информационной системы;
- ознакомиться с содержанием общих данных БД, и его архитектурой;
- осуществлять поддержку справочников и их обновлением (ввод, изменение данных и т.д.);
- иметь средние навыки пользования персональным компьютером.

### **3.2.5 Приложение к разрабатываемой ИС**

Руководство пользователя ИС Конструктор практико-ориентированных задач по программированию.

### **3.2.6 Список обозначений и сокращений**

- БД – База данных;
- ПП – программный продукт;
- ИС – Информационная Система;
- АИС – автоматизированная Информационная Система.

### **3.2.7 Назначение**

АИС предназначена для упрощения формирования контрольных работ по информатике.

АИС Конструктор практико-ориентированных задач по программированию выполняет следующие функции:

- обеспечить упрощение построения практико-ориентированных задач по программированию;
- обеспечить упрощение формирование контрольных и самостоятельных работ по информатике.

А также ряд возможностей:

- генерация практико-ориентированных задач по программированию;
- формирование контрольных и самостоятельных работ.

### **3.2.8 Условия применения.**

АИС эксплуатируется только при соблюдении предписываемых требованиях к системе, которые касаются технических, системных и прикладных сфер.

Системные требования:

- операционная система: Windows 7 SP4(только 64бит), Windows 8.1, Windows 10;
- оперативная память: от 128 мегабайт;
- необходимо наличие пакета Microsoft Office, версии 2007 и выше.

Особенно необходимы наличие программ из данного пакета: Microsoft Word 2007.



### 3.2.9 Состав и содержание носителя данных

На носителе данных размещены следующие файлы:

- Generator.exe – программный продукт;
- Generator.bd – база данных ИС, необходимая для функционирования системы. Приложение работает без ошибок, только при наличии данного файла в одном каталоге с «Generator.exe»;
- Руководство пользователя.docx – данный документ является поддержкой ИС. Текстовый файл, являющийся приложением системы;
- Квалификационная работа.docx – приложение с подробным описанием системы, тонкостями и общими требованиями.

### 3.2.10 Порядок загрузки данных и программ

Два файла необходимо разместить на локальном диске системы:

- Generator.exe – программный продукт;
- Generator.bd – база данных ИС, необходимая для функционирования системы.

Приложение будет работать без ошибок, только при размещении обоих файлов в одном каталоге.

### 3.2.11 Главное окно

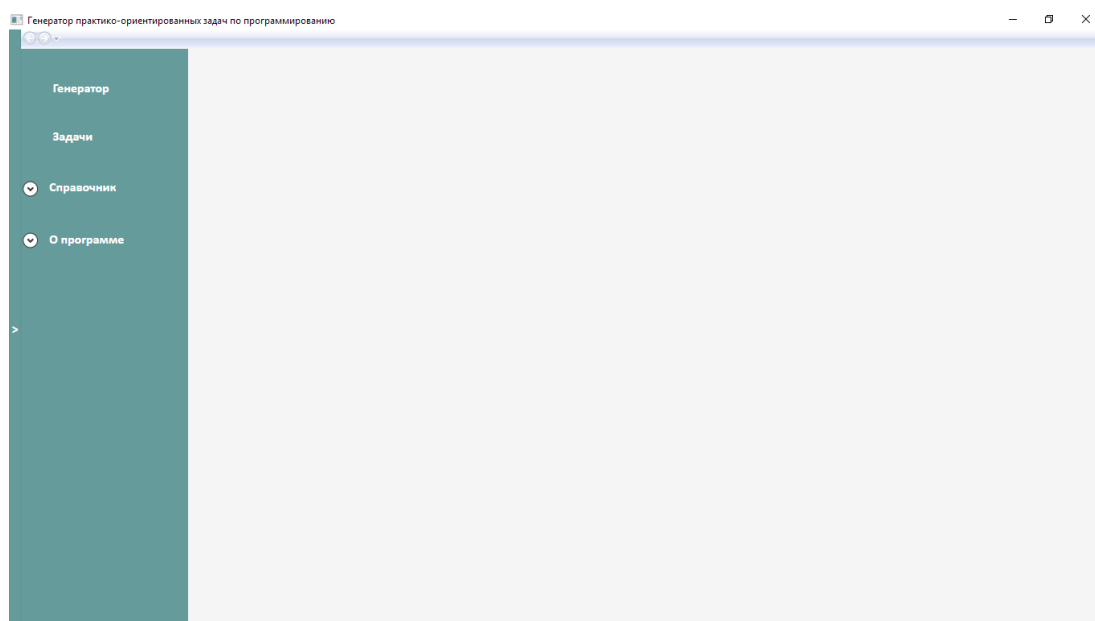


Рис. 3 – Главное окно

На рисунке 3, изображена главная форма ПП. На ней имеется меню программы, состоящее из: «Главная страница», «Генератор», «Задачи», «Справочники», «О программе».

### 3.2.12 Генератор

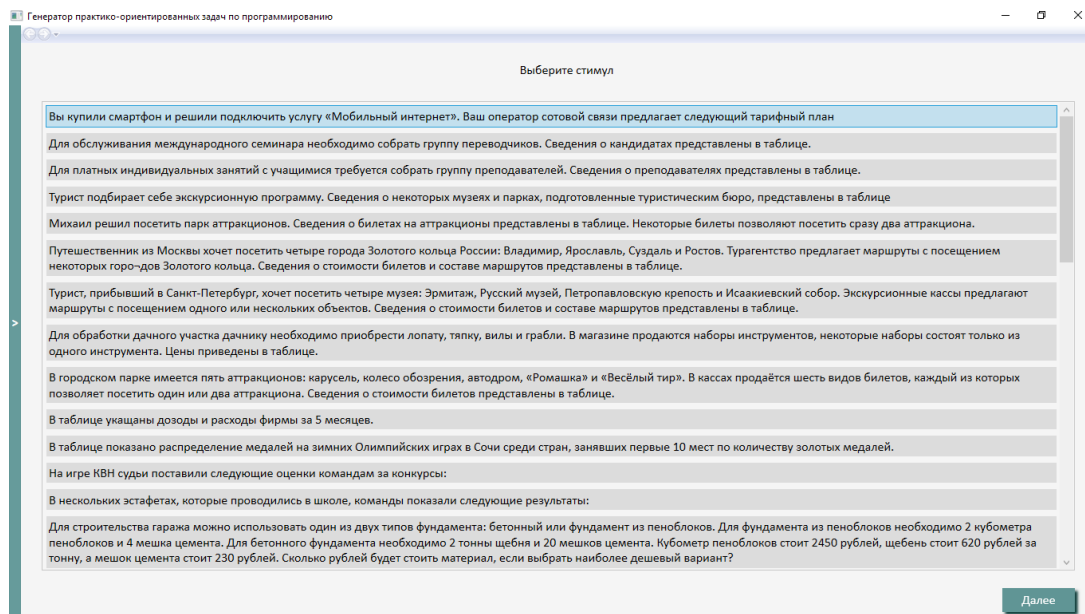


Рис. 4 – Стимулы задач

На данной странице (рис. 4) располагается генератор практико-ориентированных задач по программированию.

Генератор представляет собой мастера, который начинается с выбора стимула для задачи

Описание опций данной страницы:

- **ListBox**, при запуске страницы загружаются данные из БД, таблица Stimul. Возможно выбрать только один стимул.
- Кнопка «Далее», сохраняет выбранный стимул и отправляет к выбору источника, которые выгружаются из БД, относительно выбранного стимула.

### 3.2.13 Источник

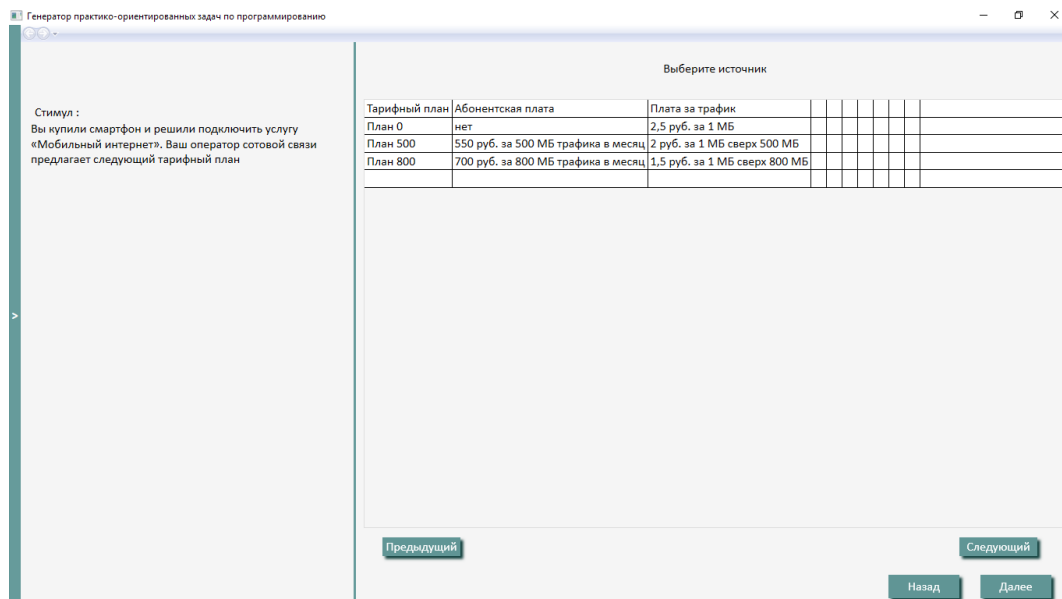


Рис. 5 – Источник задачи

На данной странице (рис. 5) располагается источник, которые выгружаются из БД, таблица Istochnik, в зависимости от выбранного ранее стимула. В левой части страницы можно увидеть ранее выбранный стимул.

Описание опций данной страницы:

- DataGrid, при запуске странице загружается источник из БД, в зависимости от выбранного стимула.
- Кнопки «Следующий» и «Предыдущий» дают возможность выбрать подходящий источник, если их более одного.
- Кнопка «Далее», сохраняет выбранный источник и отправляет к выбору темы задачи.
- Кнопка «Назад», возможность вернуться на предыдущую страницу, с целью изменения выбора предыдущего компонента задачи.

### 3.2.14 Тема

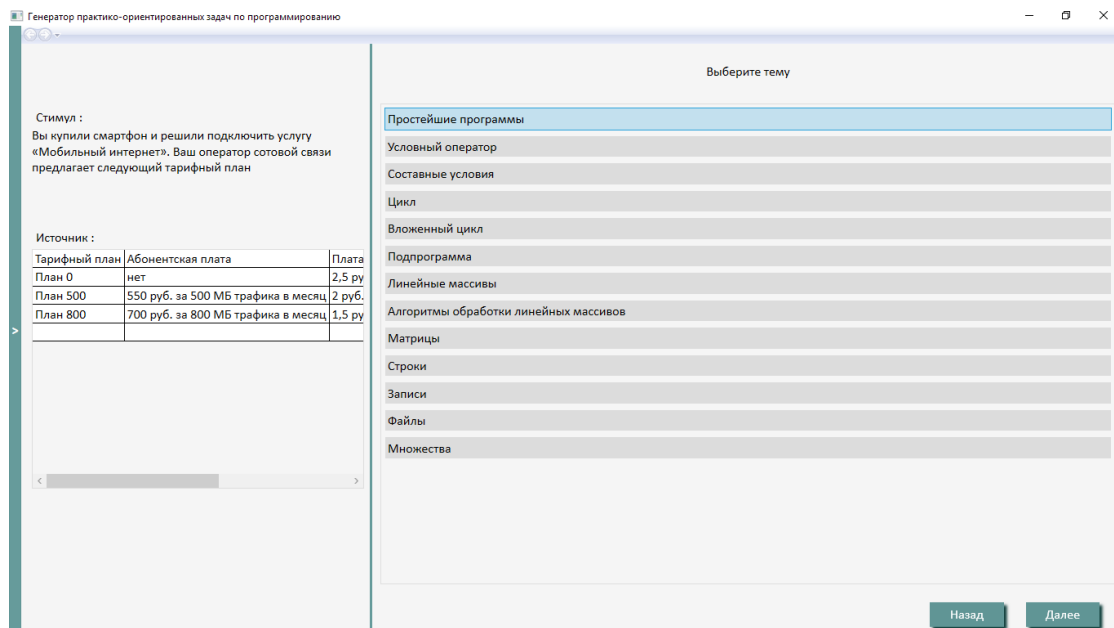


Рис. 6 – Темы задач

На данной странице (рис. 6) располагается список тем задач, полученных из БД. В левой части страницы можно увидеть ранее выбранные стимул и источник.

Описание опций данной страницы:

- ListBox, при запуске страницы загружаются данные их БД, таблица Тема. Возможно выбрать только одну тему.
- Кнопка «Далее», сохраняет выбранную тему и отправляет к выбору типа программы.
- Кнопка «Назад», возможность вернуться на предыдущую страницу, с целью изменения выбора предыдущего компонента задачи.

### 3.2.15 Тип задачи

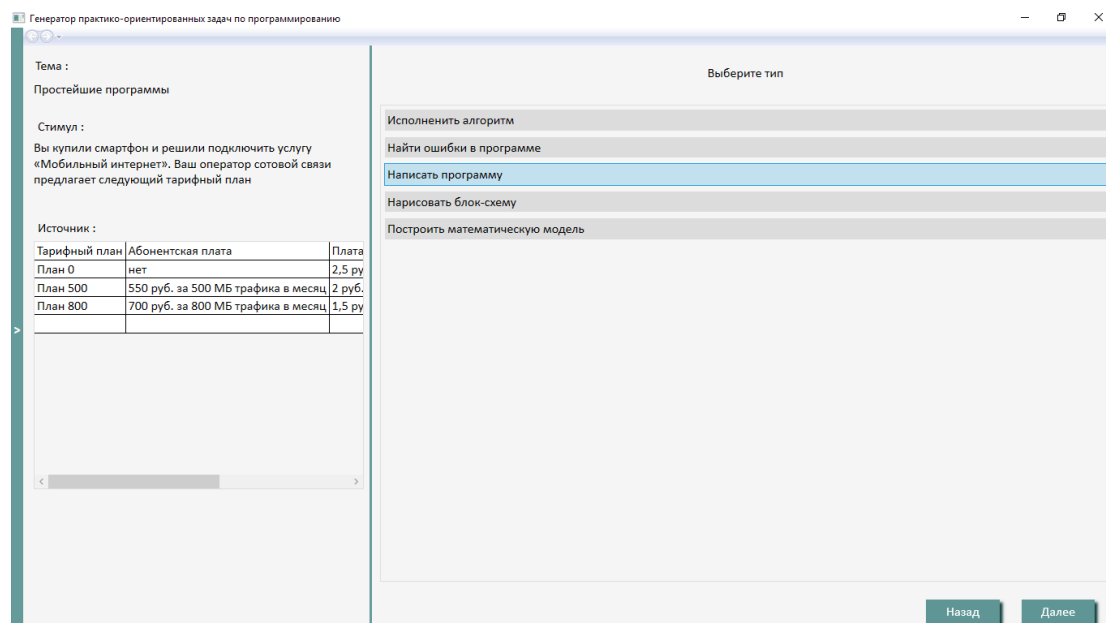


Рис. 7 – Типы задач

На данной странице (рис. 7) располагается список типов программ, полученных из БД. В левой части страницы можно увидеть ранее выбранные тему, стимул и источник задачи.

Описание опций данной страницы:

- ListBox, при запуске страницы загружаются данные их БД, таблица Тирпрог. Возможно выбрать только один тип программы.
- Кнопка «Далее», сохраняет выбранный тип программы и отправляет к введению содержания, шаблон которого отображается в зависимости от выбранных темы и типа программы.
- Кнопка «Назад», возможность вернуться на предыдущую страницу, с целью изменения выбора предыдущего компонента задачи.

### 3.2.16 Содержание

Генератор практико-ориентированных задач по программированию

Тема :  
Простейшие программы

Тип задачи :  
Написать программу

Стимул :  
Вы купили смартфон и решили подключить услугу «Мобильный интернет». Ваш оператор сотовой связи предлагает следующий тарифный план

Источник :

Тарифный план	Абонентская плата	Плата
План 0	нет	2,5 руб
План 500	550 руб. за 500 МБ трафика в месяц	2 руб.
План 800	700 руб. за 800 МБ трафика в месяц	1,5 руб.

Введите содержание

Напишите программу, которая ....  
Входные данные: ...  
Выходные данные: ...  
Пример работы программы: ...  
Входные данные: ...  
Выходные данные: ...

Назад Далее

Рис. 8 – Содержание задачи

На данной странице (рис. 8) располагается шаблон содержания задачи, полученный из БД. В левой части страницы можно увидеть ранее выбранные тему, тип программы, стимул и источник задачи.

Описание опций данной страницы:

- TextBox, при запуске страницы загружаются данные из БД, таблица Soderganie, в зависимости от выбранных темы и типа программы. Содержание возможно менять, на усмотрение оператора.
- Кнопка «Далее», сохраняет содержание и отправляет к введению критериев задачи, шаблон которого отображается в зависимости от выбранных темы и типа программы.
- Кнопка «Назад», возможность вернуться на предыдущую страницу, с целью изменения выбора предыдущего компонента задачи.

### 3.2.17 Критерии задачи

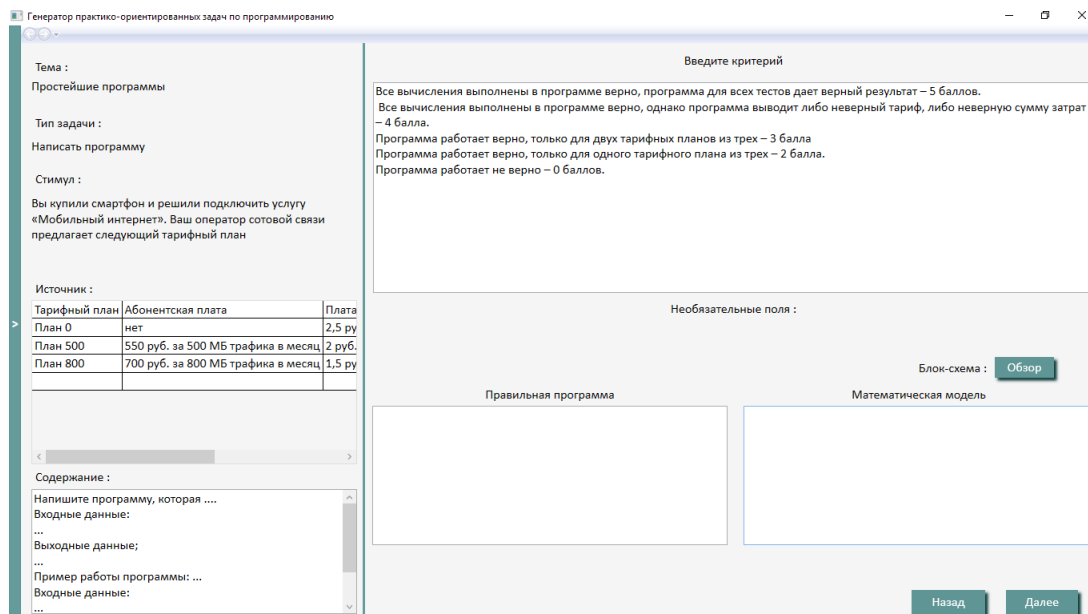


Рис. 9 – Критерии задачи

На данной странице (рис. 9) располагается шаблон критериев задачи, полученный из БД. В левой части страницы можно увидеть ранее выбранные тему, тип программы, стимул, источник и содержание задачи.

Так же располагаются необязательные компоненты, такие как: математическая модель, правильная программа и возможность загрузки изображения блок-схемы.

Описание опций данной страницы:

- **TextBox Критерии**, при запуске страницы загружаются данный из БД, таблица Kriterii в зависимости от выбранных темы и типа программы. Критерии возможно менять, на усмотрение оператора.
- **Кнопка «Обзор»**, позволяет пользователю загрузить изображение блок-схемы, нужной для задачи.
- **TextBox Математическая модель**, в случае, если математическая модель нужна в генерируемой задаче, пользователь может вписать нужную ему математическую модель.
- **TextBox Правильная программа**, в случае, если правильная программа нужна в генерируемой задаче, пользователь может вписать правильную программу.

- Кнопка «Далее», сохраняет всю информацию со страницы и отправляет к просмотру готовой задачи.
- Кнопка «Назад», возможность вернуться на предыдущую страницу, с целью изменения выбора предыдущего компонента задачи.

### 3.2.18 Задача

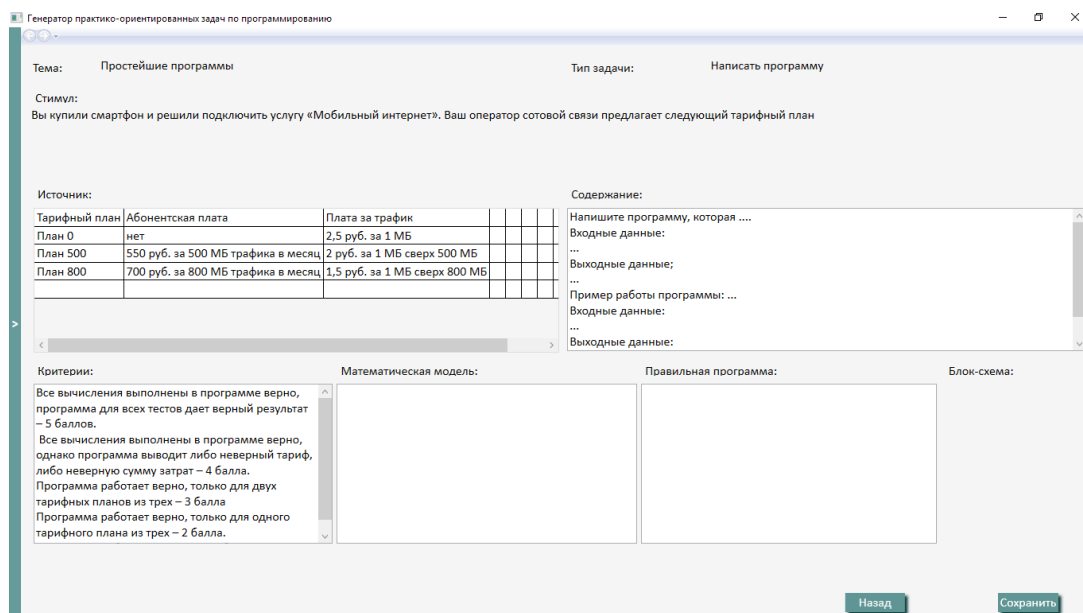


Рис. 10 – Собранная задача

На данной странице (рис. 10) отображаются все выбранные ранее компоненты задачи.

Описание опций данной страницы:

- TextBox, Datagrid, Image, просмотр ранее выбранных компонентов.
- Кнопка «Назад», возможность вернуться на предыдущую страницу, с целью изменения выбора предыдущего компонента задачи.
- Кнопка «Сохранить», сохраняет в БД, таблица Zadachi, построенную ранее задачу. Переносит на страницу выбора дальнейших действий.



### 3.2.19 Переход на нужную страницу

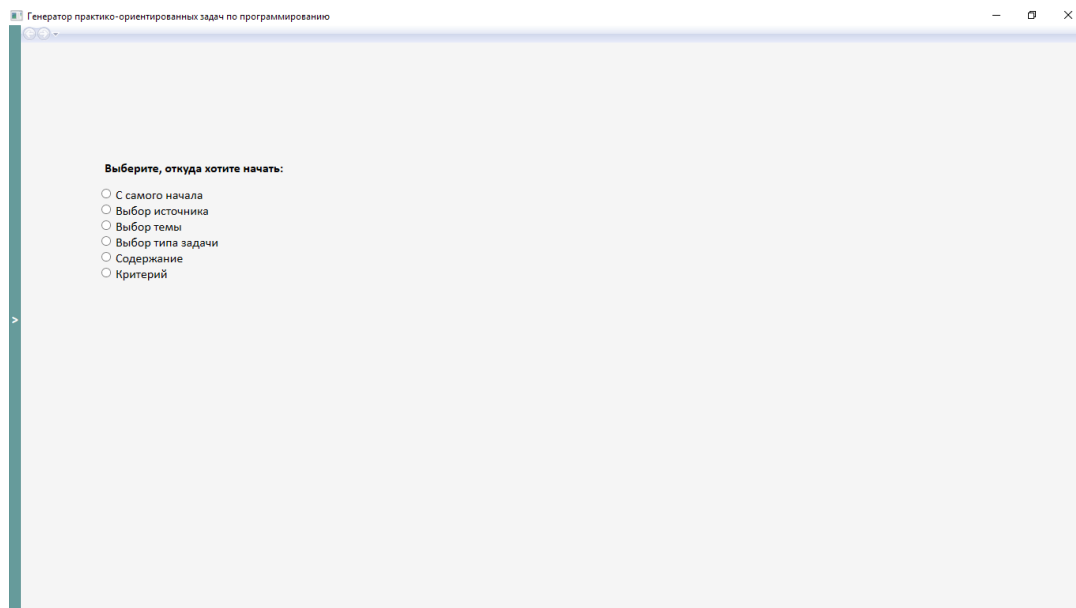


Рис. 11 – Переход на определенную страницу

На данной странице (рис. 11) отображается список ранее показанных страниц, на которые можно перейти для построения новой задачи и все ранее выбранные данные будут сохранены.

Описание опций данной страницы:

- RadioButton, содержит список страниц генератора. Выбрав определённый RadioButton, вы перейдете на нужную страницу.
- После того, как все задачи собраны, можно перейти на страницу «Задачи»

## 3.2.20 Задачи

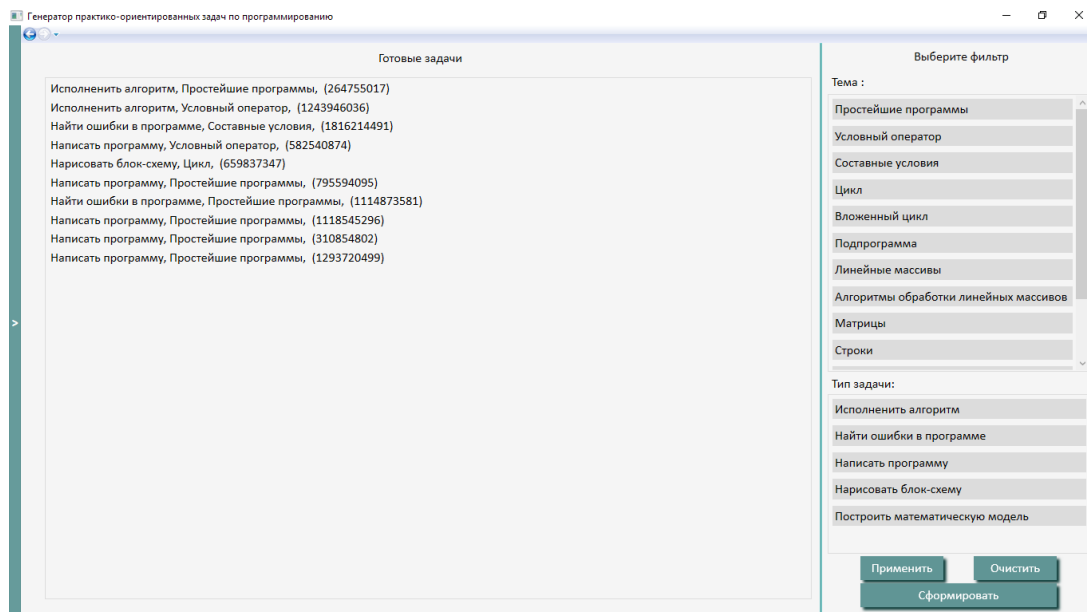


Рис. 12 – Список готовых задач

На данной странице (рис. 12) располагается список готовых задач, которые можно отфильтровать по теме и типу программы. Задачи можно просмотреть или собрать в Word файл.

Описание опций данной страницы:

- **ListBox Задачи**, при запуске страницы загружаются данные из БД, таблица *Zadachi*. При двойном клике на задачу открывается страница с просмотром содержимого задачи.
- **ListBox Тема, Тип программы**, при запуске страницы загружаются данные из БД, таблицы *Tema* и *Tipprog*. Выбрать возможно только один вариант
- Кнопка «Применить», при выборе из списка нужных тем и тапа программы, происходит фильтрация задач в **ListBox Задачи**.
- Кнопка «Очистить», возвращает **ListBox** в начальное значение.
- Кнопка «Сформировать», формирует 2 Word документа, по выбранным в **ListBox Задачи, задачам**. Первый документ содержит Тему, Тип, Стимул, Источник, Содержание задач. Второй документ содержит Критерии, Математическую модель, Правильную программу, Блок-схему задач.

### 3.2.21 Просмотр задачи

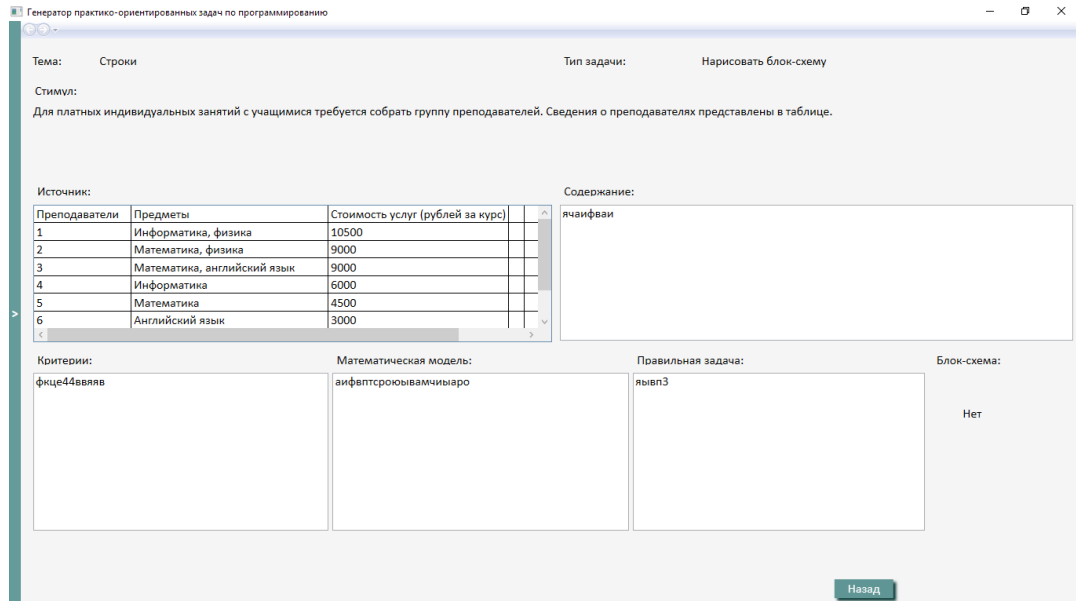


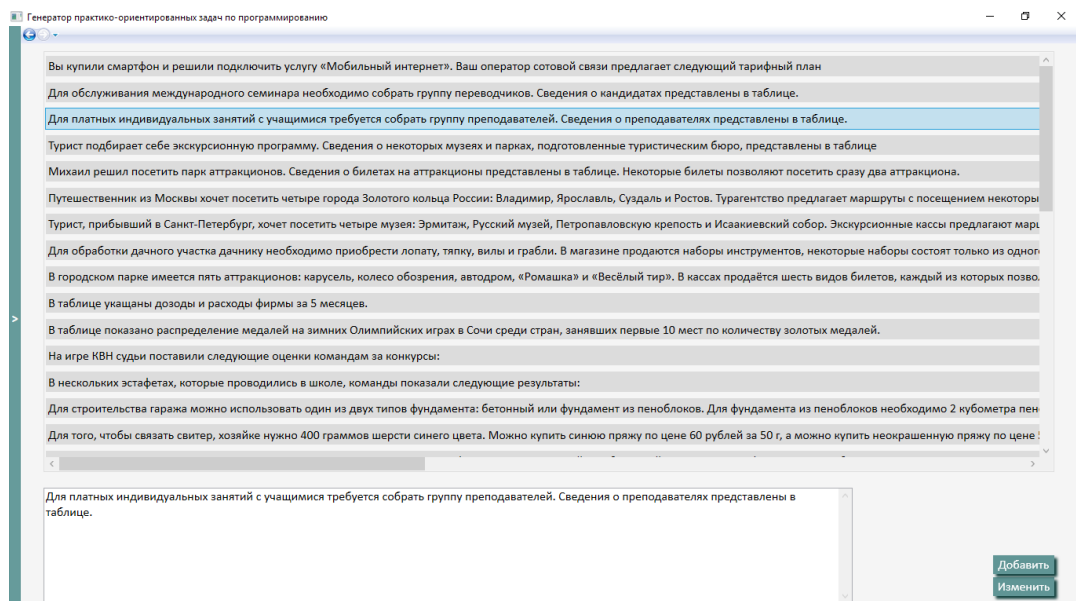
Рис. 13 – Просмотр задачи

На данной странице (рис. 13) отображаются все компоненты выбранной задачи

Описание опций данной страницы:

- TextBox, Datagrid, Image, отображает данные задачи, взятые из БД, таблица Zadachi
- Кнопка «Назад», возможность вернуться на предыдущую страницу.

### 3.2.22 Справочники



### Рис. 14 – Справочник стимула

На рисунке (рис. 14) представлен пример одного из справочников.

Справочники предоставляют возможность добавлять новые элементы или изменять имеющиеся в БД. В справочниках возможно добавлять: стимулы, источники, темы, типы, содержания и критерии задач.

### 3.3 Техничко-экономическое обоснование программного продукта

Целью выполнения технико-экономического обоснования является распределение и обоснованность растраты ресурсов на разработку программного продукта.

Разработка программного продукта требует трудовых затрат, приобретения расходных материалов и других затрат [4].

Расчет затрат на оплату труда представлен в таблице 2.

Таблица 2

Расчет затрат на оплату труда

№ п/п	Показатель	Единица измерения	Величина затрат
1	Затраты времени исполнителя на разработку	Дни	21
2	Ставка дневной заработной платы исполнителя	Руб.	500
3	Величина заработной платы исполнителя за выполнение разработки	Руб.	10500
4	Ставка страхового взноса в Пенсионный фонд [6]	%	22
5	Ставка страховых взносов в Фонд социального страхования [6]	%	2,9
6	Ставка страховых взносов в Федеральный фонд обязательного медицинского страхования [6]	%	5,1
7	Совокупный процент ставки страховых взносов	%	30
8	Величина страховых взносов	Руб.	3150

9	Страховые тарифы на обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний	%	2,9
10	Отчисления на обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний [6]	Руб.	304,5
11	Общая сумма страховых взносов	Руб.	3454,5
12	Суммарные затраты на оплату труда с учетом страховых взносов	Руб.	13954,5

Расчет затрат на материалы, приобретаемые для выполнения разработки, представлен в таблице 3

Таблица 3

Расчет затрат на материалы

№ п/п	Наименование материала	Единица измерения	Количество	Цена за единицу, руб.	Сумма затрат, руб.
1	Бумага формата А4	Лист	100	3	300
2	Ручка шариковая	Шт.	1	21	21
3	Карандаш простой	Шт.	1	19	19
Итого:					340

Расчет стоимости прочих затрат представлен в таблице 4.

Таблица 4

Расчет стоимости прочих затрат

№ п/п	Наименование материала	Единица измерения	Количество	Цена за единицу, руб.	Сумма затрат, руб.
1	Услуги Интернет-провайдера	Месяц	1	500	500
Итого:					500

Расчет полной себестоимости на разработку программного продукта представлена в таблице 5.

Таблица 5

Расчет полной себестоимости и цены договора на разработку программного продукта

Наименование показателя	Единица измерения	Значение показателя
Заработная плата исполнителей с учетом страховых взносов	Руб.	13954,5
Затраты на материалы	Руб.	340
Прочие прямые затраты	Руб.	500
Полная себестоимость проекта	Руб.	14794,5
Средний уровень прибыльности (рентабельности) проектов разработки программных продуктов	%	22,5
Планируемый размер прибыли	Руб.	3328,76

Планируемая договорная цена разработки программного продукта	Руб.	15000
Фактическая цена разработки программного продукта	Руб.	18123,26



### **Вывод по главе 3**

В третьей главе была спроектирована модель ИС, реализована совместно с разработкой ПП. В последствии данный продукт может стать хорошим помощником учителю информатики в проведении занятий. В ходе разработки проводились различные тесты и испытания ИС, их целью было минимизировать проблемы с работой ПП в будущем.

В рамках испытаний смоделировано максимальное количество типовых действий и ошибок пользователя, выявлены и устранены недочеты в алгоритмах информационной системы.

ИС отвечает требованиям технического задания, а также выполняет поставленную перед ней задачу.

Было разработано руководство пользователя ИС (с включенными в него скриншотами), которое облегчит использование ПП в дальнейшем.

Выполнено технико-экономическое обоснование, демонстрирующее экономическую целесообразность всей работы и разработки ПП в отдельности.

## **Заключение**

Главным результатом квалификационной работы является создание информационной системы «Конструктор практико-ориентированных задач по программированию». Которая, в свою очередь, облегчает работу учителя информатики по подготовке контрольных работ.

Была рассмотрена организационная структура МАОУ «Лицей №67 г. Челябинска», а также описан будущий функционал системы «Конструктор практико-ориентированных задач по программированию». Проведён комплексный анализ представления данных

После того как была описана и проанализирована характеристика объекта автоматизации, были описаны цели создания информационной системы.

- автоматизировать работу учителя информатики в формировании контрольных работ;
- уменьшение возникновения ошибок при составлении задач и контрольных работ.

Выделены подсистемы: подсистема хранения данных, подсистема генерации задач, подсистема формирования контрольных работ.

Основываясь на анализе аналогичных программных продуктов, была определена целесообразность данного проекта, доказана актуальность разработки информационной системы.

В ходе разработки проводились разнообразные тесты и испытания ИС, с целью минимизации проблем с работой программного продукта в будущем. Было смоделировано максимальное количество типовых действий и ошибок пользователя, выявлены и устранены недочеты в алгоритмах информационной системы.

Как следствие, разработанная информационная система, отвечает требованиям технического задания, а также выполняет поставленную перед ней задачу.

Было разработано руководство пользователя ИС (с включенными в него скриншотами), которое сильно облегчает использование ПП в дальнейшем.

Выполнено технико-экономическое обоснование, демонстрирующее экономическую целесообразность как всей работы, так и разработки ПП отдельно.

## Библиографический список

1. Введение в язык C# и .NET Framework. [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework> – 27.05.2019.
2. Виды испытаний (тестирования) информационной системы. [Электронный ресурс] – Режим доступа: <http://itconcord.ru/articles/testing/> - 27.05.2019.
3. Различные архитектурные решения, используемые при реализации многопользовательских СУБД. Краткий обзор СУБД. [Электронный ресурс] – Режим доступа: <https://www.intuit.ru/studies/courses/508/364/lecture/8643> - 27.05.2019. – 27.05.2019
4. Страховые взносы, тарифы, ставки, размеры – [Электронный ресурс] – Режим доступа: <https://www.glavbukh.ru/rubrika/157>, Дата доступа: 27.05.2019.
5. SQLite. [Электронный ресурс] – Режим доступа: <https://lecturesdb.readthedocs.io/databases/sqlite.html> - 27.05.2019.
6. Официальный сайт Муниципального автономного образовательного учреждения «Лицей № 67 г. Челябинска» [Электронный ресурс] – Режим доступа: <http://chel67.ru> – 20.02.2019.
7. Документы МАОУ Лицей №67 г. Челябинска [Электронный ресурс] – Режим доступа: <http://chel67.ru/svedeniya/doc/> 14.04.2018.
8. Структура и органы управления МАОУ Лицей №67 г. Челябинска <http://chel67.ru/svedeniya/структура-и-управления/>
9. Устав <http://chel67.ru/svedeniya/ustav/>
10. Полякова, Л.Н. Основы SQL [Текст] /Л.Н. Полякова. — Бином. Лаборатория знаний, 2007. — 368с.
11. Кузнецов, С.Д. Основы баз данных [Текст] /С.Д. Кузнецов. — 2005. — 488с.
12. Котляров, В.П. Основы тестирования программного обеспечения [Текст] /В.П. Котляров. — Бином. Лаборатория знаний, 2005. — 360с.

- 13.Новиков, Ю.В. Основы локальных сетей [Текст] /Ю.В. Новиков, С.В. Кондратенко. — ИНТУИТ.РУ, 2014. — 360с.
- 14.Албахари, Б. С# 7.0. Справочник. Полное описание языка [Текст] /Б. Албахари, Д. Албахари. — Диалектика, 2018. — 1024с.
- 15.Прайс, М.Дж. С# 7 и .NET Core. Кросс-платформенная разработка для профессионалов [Текст] /М.Дж. Прайс; пер. с англ. Гринчик Н. — Питер, 2018. — 640с.
- 16.WPF. Windows Presentation Foundation в . NET 4. 5 с примерами на С# 5.0 для профессионалов [Текст] /М. Макдональд; пер. с англ. Артеменко Ю. Н. — Вильямс, 2017. — 1024с.
- 17.Черемных, С.В. Моделирование и анализ систем. IDEF-технологии: практикум [Текст] /С.В. Черемных, И.О. Семенов, В.С. Ручкин. — Финансы и статистика, 2006. — 192с.
- 18.Компьютерные сети. Первый шаг [Текст] /У. Одом. — Вильямс, 2006. — 432с.
- 19.Компьютерные сети [Текст] /Д. Узеролл, Э. Таненбаум. — Питер, 2012. — 960с.
- 20.Компьютерные сети. Принципы, технологии, протоколы [Текст] /В. Олифер, Н. Олифер. — Питер, 2013. — 944с.
- 21.Кузин, А. В. Базы данных [Текст] / А.А. Кузин, С.В. Левонисова— Академия, 2014. – 320 с.
- 22.Глушаков, С. В. Базы данных [Текст] / С.В. Глушатков, Д.В. Ломотько – ООО «Издательство АСТ». 2014. – 415 с.
- 23.Голицына, О. Л. Базы данных [Текст] / О.Л. Голицына, Н.В. Максимов, И.И. Попов – ИНФРА-М, 2013. – 268 с.
- 24.Астахова, И. Ф. СУБД: язык SQL в примерах и задачах [Текст] / И.Ф. Астахова – Физматлит, 2013.
- 25.Астахова, И. Ф. SQL в примерах и задачах [Текст] / И.Ф. Астахова, А.П. Толстобров, В.М. Мельников – Новое знание, 2014.

26. WPF 4. Подробное руководство [Текст] / А. Натан. — Символ-Плюс, 2011.
27. Новиков, С.П. Применение новых информационных технологий в образовательном процессе [Текст] / С.П. Новиков // Педагогика. — 2003. — № 9. — 460с.
28. Visual C# 2010 [Текст] / К. Нейгел, К. Утсон — Диалектика, 2010. — 960 с.
29. Чалдаева, Л.А. Экономика предприятия: учебник для бакалавров [Текст] / Л.А. Чалдаева — 3-е изд., перераб. и доп.— Юрайт, 2013.— 411 с.
30. Амзараков, М.Б. Автоматическая генерация вариантов педагогического теста. Информационные технологии в образовании. [Текст] / М. Б. Амзараков. - Москва, 1999.
31. Беляев К.В. Об автоматической генерации закрытых тестовых заданий. Обзор прикладной и промышленной математики [Текст] / К.В. Беляев — Издательство «ТВП», 2005.
32. Посов И. А., Смирнов И. Б. Интернет-сервис для хранения базы генерируемых задач по математике [Текст] / И.А. Посов, И.Б. Смирнов — СПб.: СПбГЭТУ, 2010.
33. Сергушичева А. П. Метод и алгоритмы автоматизированного построения компьютерных тестов контроля знаний по техническим дисциплинам: диссертация на соискание ученой степени канд. техн. наук [Текст] / А.П. Сергушичева — СПб, 2007.