

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 Прикладная информатика
(код и наименование направления подготовки, специальности)

Бизнес-информатика
(направленность (профиль)/специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка программного обеспечения оценки знаний студентов (на примере ИП Кириченко И.В.)»

Студент

М.В. Степанова
(И.О. Фамилия)

(личная подпись)

Руководитель

Доктор физ-мат. наук, Профессор А.И. Сафронов
(ученая степень, звание, И.О. Фамилия)

Тольятти 2021

Аннотация

Выпускная квалификационная работа на тему: «Разработка программного обеспечения оценки знаний студентов (на примере ИП Кириченко И.В.)».

Ключевые слова: автоматизированная информационная система, база данных, программирование, программное обеспечение, тестирование знаний, обучение.

Объём работы - 68 страниц, на которых представлены 31 рисунок, 8 таблиц и 1 приложение. При написании работы использовалось 50 источников.

Объект исследования: автоматизированные информационные системы.

Предмет исследования: АИС оценки знаний студентов на примере ИП Кириченко И.В.

Цель работы: спроектировать базу данных и создать АИС оценки знаний студентов.

Практическая значимость работы заключается в том, что разработанную систему можно использовать на предприятии, при этом возможно дальнейшее совершенствование разработки.

Внедрение ПО в деятельность предприятия экономически выгодно. В дальнейшем он может быть использован в других образовательных учреждениях.

Структурно работа состоит из трех глав. Первая - анализ предметной области, вторая - разработка и реализация проектных решений, третья - оценка и обоснование экономической эффективности проекта.

Оглавление

Введение	5
Глава 1 Анализ предметной области.....	7
1.1 Техничко-экономическая характеристика предметной области	7
1.1.1. Характеристика предприятия.....	7
1.1.2 Краткая характеристика подразделения и его видов деятельности	7
1.1.3 Сущность задачи автоматизации	8
1.2 Концептуальное моделирование предметной области	9
1.3. Постановка задачи	11
1.3.1 Цель и назначение автоматизированного варианта решения задачи....	11
1.3.2 Общая характеристика организации решения задачи на ЭВМ	11
1.3.3 Формализация расчетов подзадач.....	14
1.4 Анализ существующих разработок и обоснование выбора технологии проектирования.....	15
1.4.1 Определение критериев анализа	15
1.4.2 Сравнительная характеристика существующих разработок	15
Глава 2 Разработка и реализация проектных решений	18
2.1 Логическое моделирование предметной области	18
2.1.1 Логическая модель и ее описание.....	18
2.1.2 Используемые классификаторы и системы кодирования	19
2.1.3 Характеристика нормативно-справочной и входной оперативной информации	19
2.1.4 Характеристика базы данных.....	19
2.2. Физическое моделирование АИС	21
2.2.1 Выбор архитектуры АИС	21
2.2.2 Функциональная схема проекта.....	22
2.2.3 Структурная схема проекта.....	24

2.2.4	Описание программных модулей	24
2.2.5	Схема взаимосвязи программных модулей и информационных файлов	28
2.3.	Технологическое обеспечение задачи.....	30
2.3.1	Организация технологии сбора, передачи, обработки и выдачи информации	30
2.3.2	Схема технологического процесса сбора, передачи, обработки и выдачи информации	30
2.4	Контрольный пример реализации проекта и его описание	31
Глава 3	Оценка и обоснование экономической эффективности проекта	41
3.1	Выбор и обоснование методики расчета экономической эффективности..	41
3.2	Расчет показателей экономической эффективности проекта	41
	Заключение	51
	Список используемой литературы и используемых источников	54
	Приложение А Скрипт создания базы данных.....	59

Введение

Деятельность каждого человека в условиях современного мира так или иначе связана с обработкой информации. Для успешной обработки информации и ее надежного хранения требуется большое количество хранилищ данных [24].

История развития вычислительной техники движется сразу в нескольких направлениях. Наиболее ярким из них считается применение техники в процессе решения различного рода задач: инженерных, математических, физических и многих других. Подобное направление по праву принято считать главным двигателем, обеспечивающим развитие науки и техники. Отличительной чертой применения устройств компьютерной техники является задействование сложных алгоритмов обработки данных, нацеленных на различные структуры, объем которых относительно невелик.

Второе перспективное направление - использование компьютерных технологий в пределах автоматизированных систем, где они отвечают за выполнение ряда функций:

- хранение больших объемов данных в памяти компьютера;
- выполнение специфической обработки и передачи данных в конкретной предметной области;
- реализация удобного интуитивно понятного пользовательского интерфейса [20].

В условиях современного рынка все большее количество компаний заинтересовано в специалистах узких направлений. Чаще всего обучение таких сотрудников требует прохождения особых курсов сферы дополнительного образования. При этом работодатель максимально заинтересован в том, чтобы его сотрудники обладали не только сертификатом о прохождении учебного курса, но и имели соответствующий уровень знаний.

Актуальность рассматриваемой темы очевидна – основным достоинством любого современного специалиста являются знания. Однако, чтобы оценить уровень знаний необходимы специальные технологии, позволяющие выставлять оценку объективно и непредвзято. Автоматизированная система тестирования является одним из вариантов решения данной задачи.

Объектом исследования данной работы являются автоматизированные информационные системы.

Предмет исследования – АИС оценки знаний студентов на примере ИП Кириченко И.В.

Цель работы – спроектировать базу данных и создать АИС оценки знаний студентов.

Для достижения поставленной цели необходимо решить ряд задач:

- проанализировать литературу по заданной теме;
- изучить предметную область;
- обосновать необходимость разработки информационной системы;
- описать техническое, программное и технологическое обеспечение;
- провести тестирование разработанной системы.

В работе представлено три главы:

- аналитическая, в которой содержится анализ предметной области, а также обоснование разработки ИС;
- проектная, в которой описаны процессы проектирования БД, представлена разработка ПО и приведен контрольный пример;
- экономическая, которая описывает расходы на проект и обосновывает экономическую эффективность проекта.

При написании работы в качестве опорных источников использовались Адигеев М.Г. – «Жизненный цикл программных систем» и Медведев М.А. – «Программирование на С#».

Глава 1 Анализ предметной области

1.1 Технико-экономическая характеристика предметной области

1.1.1. Характеристика предприятия

ИП Кириченко Инна Викторовна - организация сферы дополнительного образования. В настоящее время «рост внимания к дополнительному образованию наблюдается во всем мире. Это объясняется неудовлетворенностью граждан узкими границами содержания и формализмом школьного образования» [3].

«В соответствии с Законом об образовании дополнительное образование включает такие подвиды, как дополнительное образование детей и взрослых, а также дополнительное профессиональное образование» [3].

1.1.2 Краткая характеристика подразделения и его видов деятельности

Организационная структура предприятия изображена на рисунке 1.



Рисунок 1 - Организационная структура предприятия

Главным органом управления является административно-управленческий блок, отвечающий за организацию работы всего предприятия.

На следующем уровне расположены отделы и службы, отвечающие каждый за свой профиль задач. Каждый отдел может включать в себя несколько отделений.

Обособленно расположены детские образовательные объединения, что объясняется спецификой их деятельности.

1.1.3 Сущность задачи автоматизации

На сегодняшний день в системе образования наблюдается смена ее главной парадигмы. В настоящее время целью обучения является формирование всесторонне развитого студента. Однако имеющаяся система контроля знаний, утвердившаяся еще в прошлом столетии, не способна полностью формировать личности требуемых специалистов. Этот факт является главной причиной существующего сегодня противоречия между стремлением административного персонала высших учебных заведений к унификации контроля знаний, с одной стороны, и необходимостью рефлексии своего обучения каждым студентом - с другой [26].

Вопрос повышения объективности результатов является одним из ключевых в сфере образования. Главная проблема здесь заключается в том, что знания оцениваются преподавателями, которые ведут занятия, что может привести как к завышению, так и к занижению баллов.

Таким образом, разработка автоматизированной информационной системы тестирования знаний учащихся позволит не только сократить время, необходимое на проверку знаний, но и повысить объективность получаемых результатов.

Выводы по параграфу (актуальность задачи)

В рамках данного параграфа приводятся данные об общей деятельности учреждения, характеризуется организационная структура, а также описывается сущность задачи автоматизации, поставленной в рамках выполнения ВКР.

1.2 Концептуальное моделирование предметной области

В настоящее время большинство преподавателей ИП Кириченко И.В. использует классический подход к оценке знаний студентов, который предполагает проведение экзамена при помощи билетов. «Данный подход удобен для большинства сотрудников и студентов, но не является совершенным по следующим причинам» [3]:

- продолжительность процесса;
- присутствие человеческого фактора при выставлении оценки;
- недостаточный охват объема спрашиваемого на экзамене материала по сравнению с преподаваемым на протяжении учебного курса;
- возможность «списывания», допускающая получение удовлетворительной оценки при отсутствии знаний.

На рисунке 2 представлена схема данного процесса «КАК ЕСТЬ».

На рисунке 3 изображена UML-диаграмма процесса оценки знаний студентов.

На рисунке 4 представлена декомпозированная диаграмма процесса «КАК ЕСТЬ». Студент выбирает билет, готовится к ответу, отвечает преподавателю, получает результат, затем полученный результат попадает в ведомость.



Рисунок 2 - Диаграмма «КАК ЕСТЬ»

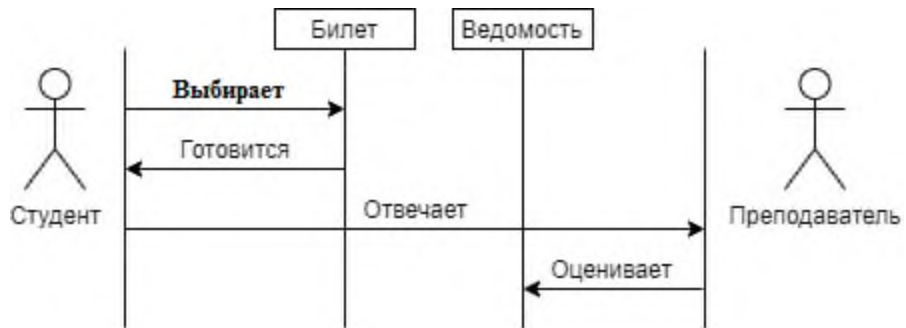


Рисунок 3 - UML-диаграмма существующего процесса оценки знаний студентов

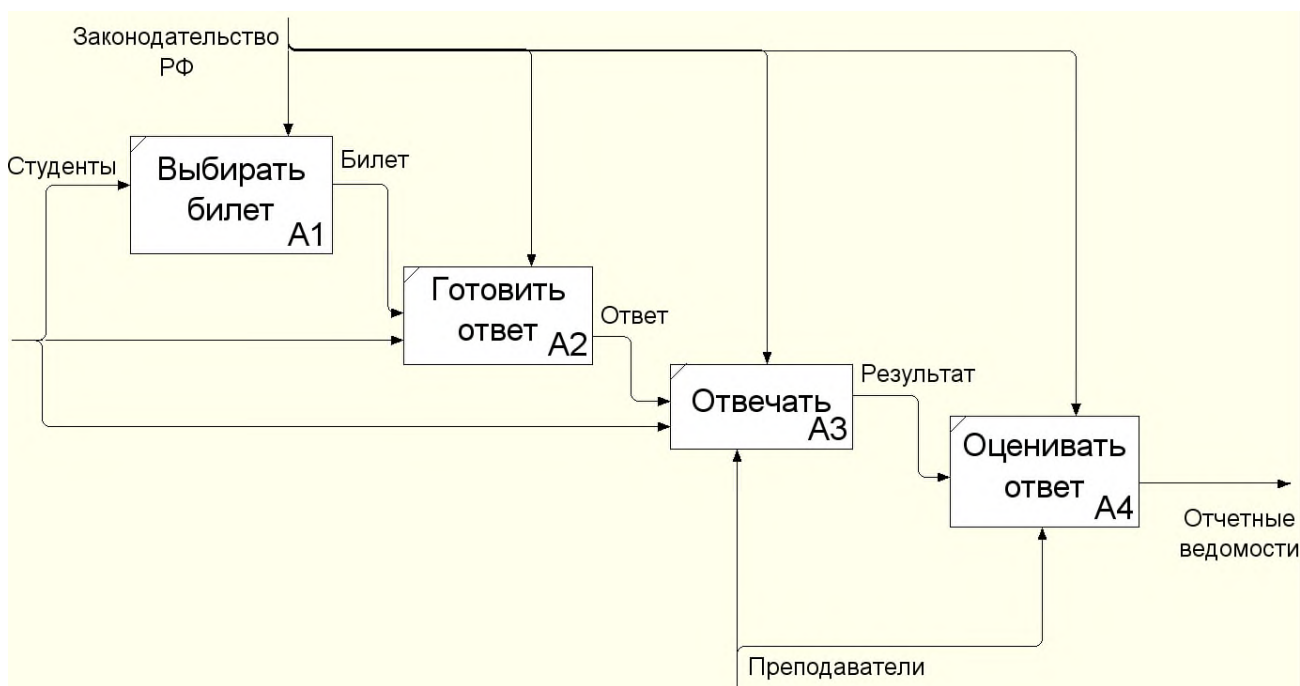


Рисунок 4 - Декомпозированная диаграмма процесса «КАК ЕСТЬ»

Данные схемы подтверждают недостатки, указанные выше.

1.3. Постановка задачи

1.3.1 Цель и назначение автоматизированного варианта решения задачи

Главная цель разработки автоматизированной системы - проверка знаний учащихся в различных областях науки.

Назначение реализации разрабатываемой системы обучения:

- автоматизация процесса проверки знаний учащихся;
- повышение общего уровня автоматизации Учреждения.

1.3.2 Общая характеристика организации решения задачи на ЭВМ

Проектируемая информационная система оценки знаний студентов ИП Кириченко И.В. должна иметь разграничение доступа при помощи двух ролей – студент и администратор, и удовлетворять следующим требованиям:

- а) доступ к базе данных правильных ответов на тесты должен быть доступным только пользователям, относящимся к учетной записи администратора;
- б) учетная запись «Администратор» должна предоставлять графический интерфейс с реализацией ряда функций:
 - добавление тестов;
 - просмотр существующих тестов и их редактирование;
 - удаление тестов;
 - просмотр информации по студентам, прошедшим тестирование;
 - назначение тестов студентам;
 - удаление тестов из предложенных к прохождению;
- в) учетная запись «Студент» должна предоставлять графический интерфейс с реализацией ряда функций:
 - просмотр и выбор теста для прохождения;
 - просмотр результатов выполненных тестов.

Для реализации работы с несколькими учетными записями в проектируемой информационной системе должны быть предусмотрены формы регистрации и авторизации пользователей.

На рисунках 5-6 представлена диаграмма процесса «КАК ДОЛЖНО БЫТЬ». В этом случае преподаватель подгружает тесты в систему, формируя их в таблице Excel. Загруженный тест назначается студенту для прохождения. Студент выбирает один из назначенных для него тестов и проходит его. Преподаватель выгружает результаты прохождения из АИС.

На рисунке 7 представлена диаграмма вариантов использования будущей системы.



Рисунок 5 - Диаграмма процесса «КАК ДОЛЖНО БЫТЬ»

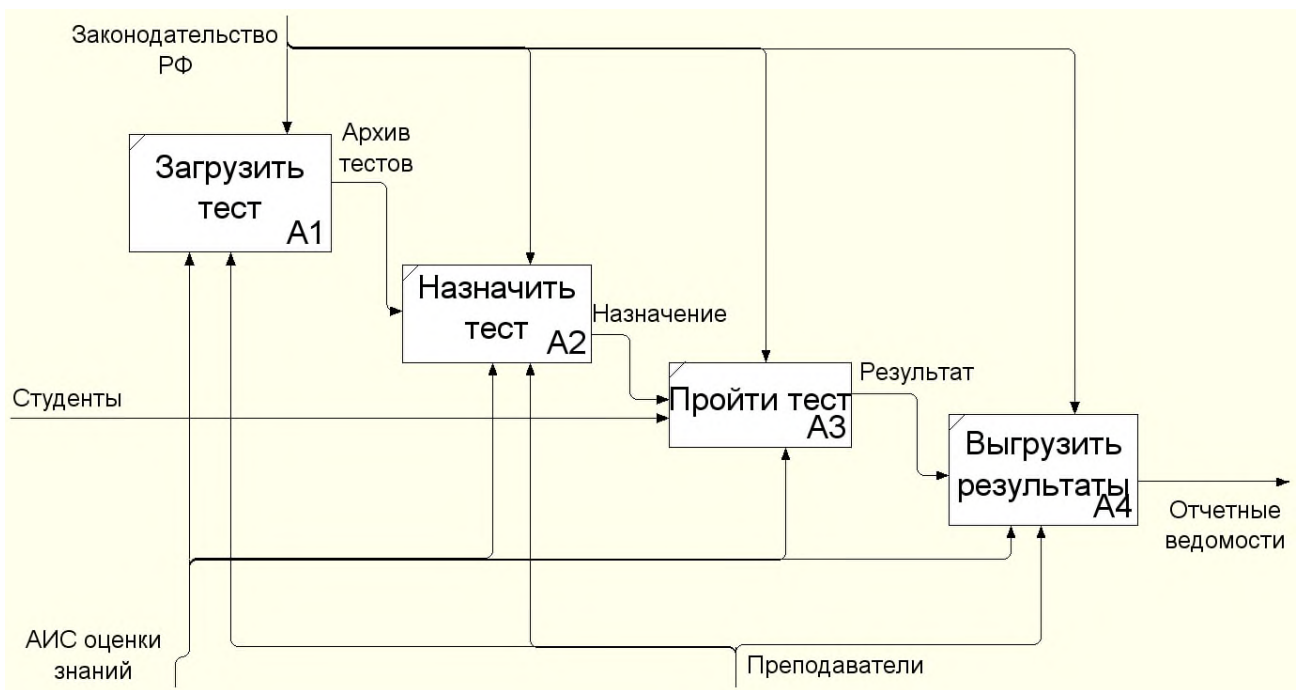


Рисунок 6 - Декомпозированная диаграмма процесса «КАК ДОЛЖНО БЫТЬ»

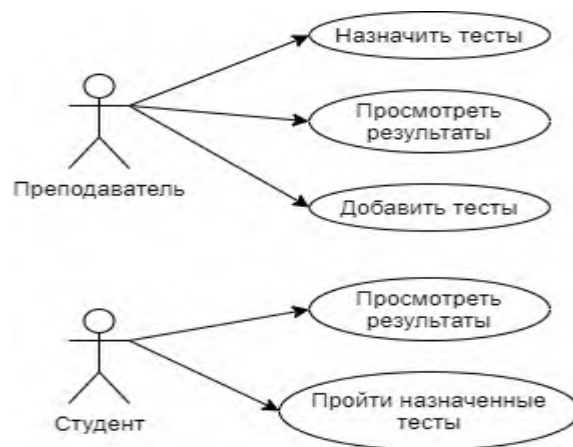


Рисунок 7 - Диаграмма вариантов использования

1.3.3 Формализация расчетов подзадач

Процесс разработки информационной системы включает в себя следующие этапы:

- изучение предметной области - деятельности образовательного учреждения;
- формулирование исходных требований;
- анализ аналогов программных продуктов, позволяющих решить задачу;
- анализ существующих средств разработки для реализации проекта;
- определение необходимых ресурсов системы;
- описание входных и выходных данных;
- проектирование физической и логической модели;
- разработка БД;
- разработка ПО;
- тестирование ПО;
- обоснование экономической эффективности проекта и подведение выводов.

Выводы по параграфу (слабые места, предложения по их устранению)

В данном параграфе приводится постановка задачи, которая должна быть выполнена в рамках работы над ВКР.

1.4 Анализ существующих разработок и обоснование выбора технологии проектирования

1.4.1 Определение критериев анализа

«В настоящее время на рынке программного обеспечения можно найти целое множество продуктов, которые направлены на обучение и тестирование.

При сравнении этих продуктов в пределах поставленной задачи необходимо уделять внимание двум ключевым параметрам:

- возможностям системы;
- стоимости системы» [3].

С точки зрения функциональных возможностей разрабатываемая система предназначена для решения следующих задач:

- предоставление отчетов;
- работа с теоретическими материалами;
- сбор статистики результатов;
- работа с практическими материалами.

1.4.2 Сравнительная характеристика существующих разработок

«Одной из наиболее популярных систем обучения, используемых в коммерческих компаниях, является Moodle - это среда дистанционного обучения, в которой можно создавать и хранить любые учебные материалы, а также задавать их последовательность обучения для сотрудников» [3].

Существующий «электронный формат предоставляет возможности работы не только с текстовыми материалами, но также и интерактивными ресурсами, начиная от web-статей и заканчивая видеороликами» [3].

Кроме того, «система Moodle предоставляет возможность совместного использования. В этом случае обучение может происходить как асинхронно, так и в режиме реального времени, когда сразу несколько людей изучают одни и те же материалы» [3].

Результаты обучения «каждого студента хранятся в личном портфолио, содержащем сведения о пройденных тестах, полученных результатах и комментариях преподавателей» [3].

«Данная система распространяется в открытом коде и является бесплатным продуктом. Кроме того, она предоставляет набор специальных средств разработки дистанционных курсов, за счет чего снижается стоимость решения проблем совместимости с другими курсами» [3].

Таким образом, рассмотренная система ориентируется на западную модель обучения.

«Еще одной популярной системой является WebTutor – система дистанционного обучения от компании WebSoft» [3].

Данная система предполагает использование двух ролей:

- «администраторы – пользователи, способные управлять всеми процессами существующей системы;
- обучаемые – непосредственные пользователи системы» [3].

«Система WebTutor состоит из нескольких модулей, основными из которых являются» [3]:

- «библиотека – модуль теоретических данных;
- дистанционное обучение – модуль, формирующий электронные курсы и отвечающий за реализацию доступа к этим курсам;
- управление знаниями – построение карты и профилей знаний, настройка связей между информационными материалами и т.п.;
- тестирование – модуль редактора и прохождения тестов» [3];

- «персонал – модуль, отвечающий за ведение списков пользователей и формирование их в специальные группы для обучения;
- подбор персонала - модуль предназначен для организации подбора персонала при помощи рекрутинговых агентств и специализированных сайтов» [3].

В силу того, что компания ИП Кириченко И.В. является международной компанией, работающей не только на территории России, но и на территории Евросоюза, использование сторонних приложений для создания корпоративной системы обучения невозможно. Следовательно, рассмотренные системы не подходят для дальнейшего использования.

Поэтому требуется разработка такой системы, которая не будет предоставлять сведения в глобальную сеть Internet, что позволит соответствовать требованиям корпоративных стандартов и законодательству.

Выводы по параграфу (решение)

В данном параграфе рассмотрены две популярные системы дистанционного образования, описаны их основные возможности.

Выводы по главе 1

В данной главе работы приведен анализ предметной области. Описана характеристика предприятия ИП Кириченко И.В.

Задача автоматизации, поставленная в рамках ВКР, заключается в разработке автоматизированной информационной системы тестирования знаний учащихся. В главе проведено концептуальное моделирование предметной области.

Кроме того, в данной главе описаны аналоги существующих разработок на примерах Moodle и WebTutor. В результате данного обзора было принято решение разработки собственного приложения, удовлетворяющего всем требованиям учреждения.

Глава 2 Разработка и реализация проектных решений

2.1 Логическое моделирование предметной области

2.1.1 Логическая модель и ее описание

Логическая модель разрабатываемой системы изображена на рисунке. «Логическая модель отображает схему движения входных, промежуточных и результативных потоков, а также функций предметной области» [8].

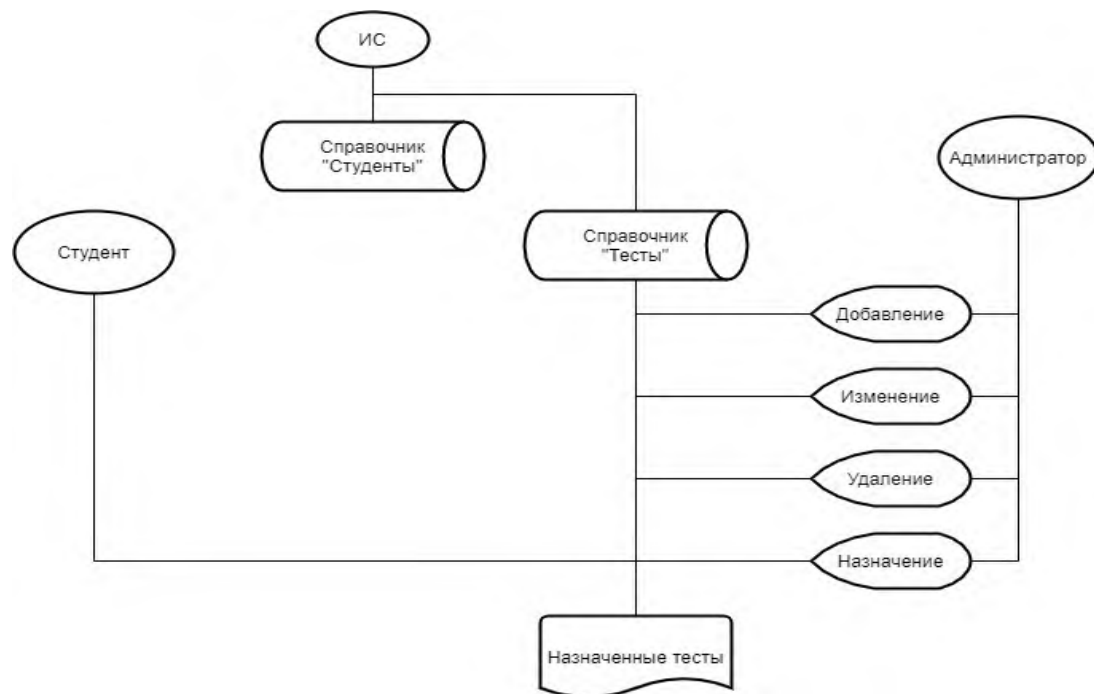


Рисунок 8 - Информационная модель

В информационной системе содержится два основных справочника – «Студенты» - перечень зарегистрированных студентов и «Тесты» - набор загруженных тестов.

Администратор может добавлять, редактировать и удалять данные о тестах, а также назначать их студентам.

Студент может смотреть список назначенных ему тестов, проходить их и смотреть результаты.

2.1.2 Используемые классификаторы и системы кодирования

В рамках разработки информационных систем под кодированием понимается «процесс присвоения объектам кодовых обозначений с целью однозначной идентификации объектов, а также для обеспечения требуемой достоверности кодируемой информации» [2]. Выбор той или иной системы кодирования зависит «от количества используемых классификационных признаков и разработанной системы классификации» [2].

В данной работе в качестве кодов использованы натуральные числа.

2.1.3 Характеристика нормативно-справочной и входной оперативной информации

Входными данными для работы с системой будет база данных, сущностями которой являются [39]:

- студенты;
- тесты;
- вопросы.

2.1.4 Характеристика базы данных

Согласно идеологии Т. Кодда для хранения данных были разработаны таблицы, структура которых представлена в таблицах 1-4.

Таблица 1 – Структура таблицы «Студенты»

Имя поля	Тип данных
Фамилия	Текстовый
Имя	Текстовый
Отчество	Текстовый
Группа	Текстовый
Логин	Текстовый
Пароль	Текстовый

Таблица 2 – Структура таблицы «Тесты»

Имя поля	Тип данных
Название	Текстовый
Вопросник	Счетчик

Таблица 3 – Структура таблицы «СтудентТест»

Имя поля	Тип данных
id	Счетчик
Логин	Текстовый
idТеста	Числовой
Прохождение	Логический
Балл	Числовой

Таблица 4 – Структура таблицы «Вопросы»

Тест	Числовой
Вопрос	Числовой
Текст	Текстовый
В1	Текстовый
В2	Текстовый
В3	Текстовый
В4	Текстовый
Ответ	Текстовый

Концептуальная модель проектируемой БД представлена на рисунке 9.



Рисунок 9 - Концептуальная модель

2.1.5 Характеристика результатной информации

В процессе работы с информационной системой по требованию администратора должна предоставляться различная информация:

- тесты, загруженные в систему;
- список групп;
- список студентов;
- результаты тестирования;
- список тестов, назначенных конкретному студенту.

По требованию студента должна предоставляться следующая информация:

- доступные тесты;
- результаты пройденных тестов.

Выводы по параграфу

В рамках данного параграфа описывается логическая модель предметной области – выделены основные сущности.

2.2. Физическое моделирование АИС

2.2.1 Выбор архитектуры АИС

На этапе физического моделирования создается модель, которая представляет собой «совокупность структуры, методов и средств редуцированного воплощения системы, необходимая для проверки в реальных условиях работоспособности будущей системы и адекватности ее моделей» [12].

Разрабатываемая система строится на базе клиент-серверной архитектуры. В этом случае на сервере размещается база данных, с которой взаимодействуют клиенты.

2.2.2 Функциональная схема проекта

На рисунке 10 представлена диаграмма возможных состояний программы, выполняющейся под учетной записью студента [37].

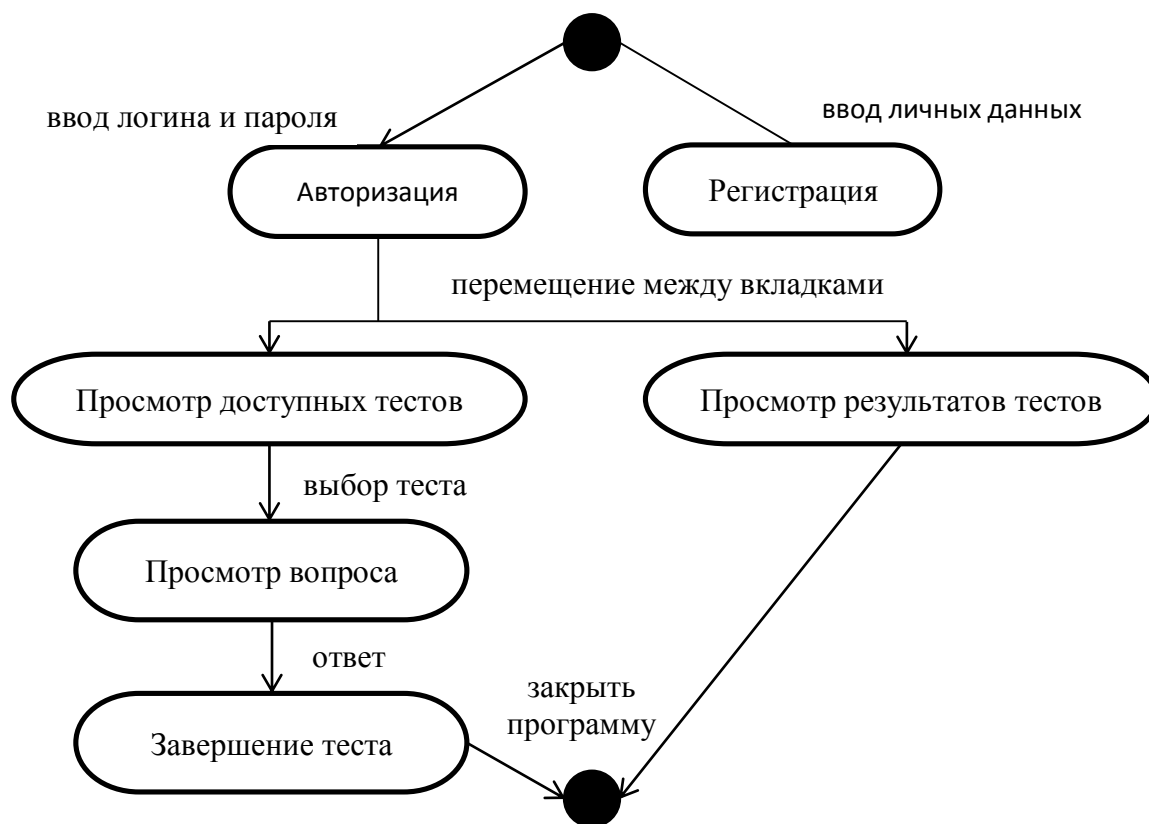


Рисунок 10 - Диаграмма состояний программы, выполняющейся под учетной записью «Студент»

При входе в систему оценки знаний студент может совершить два действия:

- пройти процедуру регистрации – заполнить личные данные, в результате чего программа сгенерирует логин и пароль;
- авторизоваться в системе – ввести логин и пароль для входа.

После входа в систему студенту может выбрать один из двух дальнейших вариантов событий:

- выбор теста для прохождения;
- просмотр результатов пройденных тестов.

Диаграмма возможных состояний программы, выполняющейся под учетной записью администратора, приведена на рисунке 11.

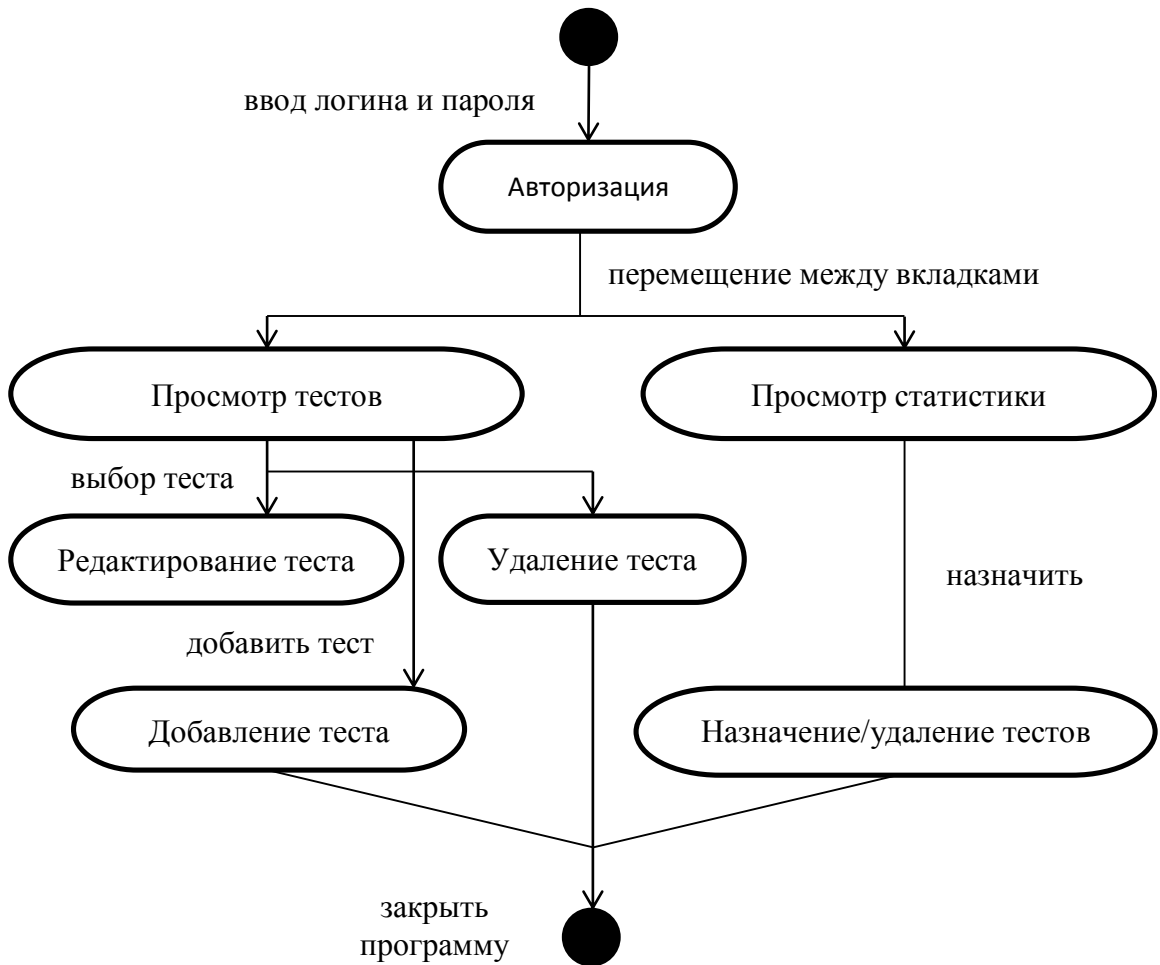


Рисунок 11 - Диаграмма состояний программы, выполняющейся под учетной записью «Администратор»

После входа в систему администратор может выполнять следующие функции:

- просмотр имеющихся в системе тестов;
- редактирование тестов;
- добавление и удаление тестов;

- просмотр результатов тестов, сданных студентами;
- назначение тестов студентам;
- удаление теста из списка для прохождения конкретного студента.

2.2.3 Структурная схема проекта

Структурная схема проекта изображена на рисунке 12.

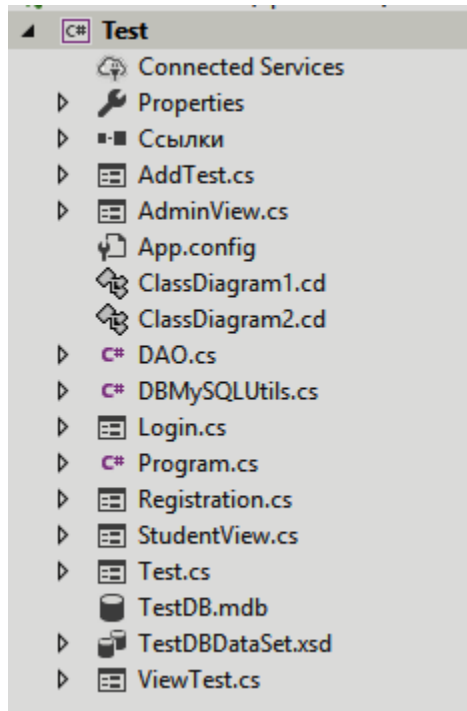


Рисунок 12 - Структурная схема проекта

Описание классов и программных модулей представлено ниже.

2.2.4 Описание программных модулей

Модули, созданные при разработке продукта:

- «DAO – класс, отвечающий за взаимодействие с базой данных» [32, с. 266]. В состав класса входят следующие поля и методы:
 - «string ConnectionString – строка подключения к БД;
 - OleDbConnection conn – соединение с БД;

- void createConnection() – метод установки соединения с БД;
- void closeConnection() – метод закрытия соединения;
- DataSet select(String commandText, String Table) – метод выборки данных из таблицы Table по запросу commandText;
- void update(String commandText) – метод обновления данных по запросу commandText;
- delete(string commandText) – метод удаления данных по запросу commandText;
- insert(string commandText) – метод вставки данных по запросу commandText» [39, с. 849];

б) FormLogin – класс, реализующий форму авторизации приложения.

Методы, реализуемые данным классом:

- "FormLogin – конструктор формы, с помощью которого происходит ее создание;
- void btnIn_Click – метод входа в систему, который проверяет корректность пары логин-пароль;
- void btnReg_Click() – метод, предназначенный для создания формы регистрации в системе;
- tbPass_KeyDown – метод, позволяющий скрыть текст, вводимый в поле пароля;

в) Registration - класс, описывающий дополнительную форму приложения, предназначенную для регистрации пользователя в системе. Методы, реализуемые данным классом:

- Registration - конструктор формы;
- void btnReg_Click - метод обработки нажатия кнопки «Зарегистрироваться». Создает нового пользователя в базе данных;

- string createPass – метод создания пароля на основе текущего времени [21];
 - string createLogin – метод создания логина на основе введенных данных и текущего времени;
 - char toLatin – метод перевода кириллических символов в латинские;
- г) AdminView – класс, описывающий систему с точки зрения администратора. Методы, реализуемые данным классом:
- AdminView - конструктор формы, с помощью которого происходит начальная инициализация отображаемых данных;
 - void AdminView_FormClosed – метод завершения работы приложения;
 - int getIDTest – метод получения уникального идентификатора теста по его имени;
 - void listTest_DoubleClick – метод токрытия выбранного теста для просмотра и редактирования;
 - void btnDel_Click – метод удаления теста;
 - void btnAdd_Click – метод импорта теста из файла Excel;
 - void btnAddTest_Click – метод, который служит для назначения тестов студентам;
 - void listTest_SelectedIndexChanged – метод, предназначенный для проверки доступности теста на удаление;
- д) FormAddTest - класс, описывающий дополнительную форму приложения, реализующую назначение тестов студентам. Методы данного класса:
- FormAddTest() - конструктор формы, необходимый для инициализации исходных данных;

- void cbGroup_SelectedIndexChanged – метод, позволяющий отслеживать изменения в значении выпадающего списка «Группа»;
 - void cbStud_SelectedValueChanged – метод, позволяющий отслеживать изменения в значении выпадающего списка «Студент»;
 - void listTest_SelectedIndexChanged – метод обработки выбора элемента в списке «Назначенные тесты»;
 - void listAll_SelectedIndexChanged – метод обработки выбора элемента в списке «Доступные тесты»;
 - string getLogin – метод получения логина;
 - int getID – метод получения уникального идентификатора теста;
 - btnAdd_Click – обработка назначения теста;
 - btnDel_Click – обработка удаления назначенного теста;
- е) ViewTest - класс, описывающий дополнительную форму приложения, целью которой является просмотр существующего теста. Методы данного класса:
- ViewTest() - конструктор формы, инициализирующий исходные данные;
 - void dgVoprosy_CellContentClick – метод, позволяющий редактировать значение из выбранного информационного поля;
 - void btnOk_Click – метод применения изменений;
 - void btnDelQ_Click – метод удаления вопроса;
- ж) StudentView - класс, описывающий дополнительную форму приложения, соответствующую учетной записи студента. Методы данного класса:

- StudentView() - конструктор формы, инициализирующий исходные данные;
 - string getLogin – метод получения Ф.И.О. по логину;
 - int getIDTest – метод получения уникального идентификатора теста;
 - void StudentView_FormmClosed – метод выхода из приложения;
 - void btnGo_Click – метод запуска теста для прохождения;
 - void tabControl1_selectedIndexChanged – метод переключения между вкладками;
 - void listTest_SelectedIndexChanged – метод обработки выбора теста в списке.
- з) Test - класс, описывающий дополнительную форму приложения, реализующую прохождение теста. Методы данного класса:
- Test() - конструктор формы, реализующий загрузку теста;
 - void btnNext_Click – метод перехода к следующему вопросу;
 - void rb1-rb4_CheckedChanged – метод обработки выбора варианта ответа;
 - void btnEnd_Click – метод завершения прохождения теста;
 - void Test_FormClosing – метод закрытия формы теста, не позволяющий выйти из тестирования до его завершения.

2.2.5 Схема взаимосвязи программных модулей и информационных файлов

На рисунке 13 представлена диаграмма классов проекта.

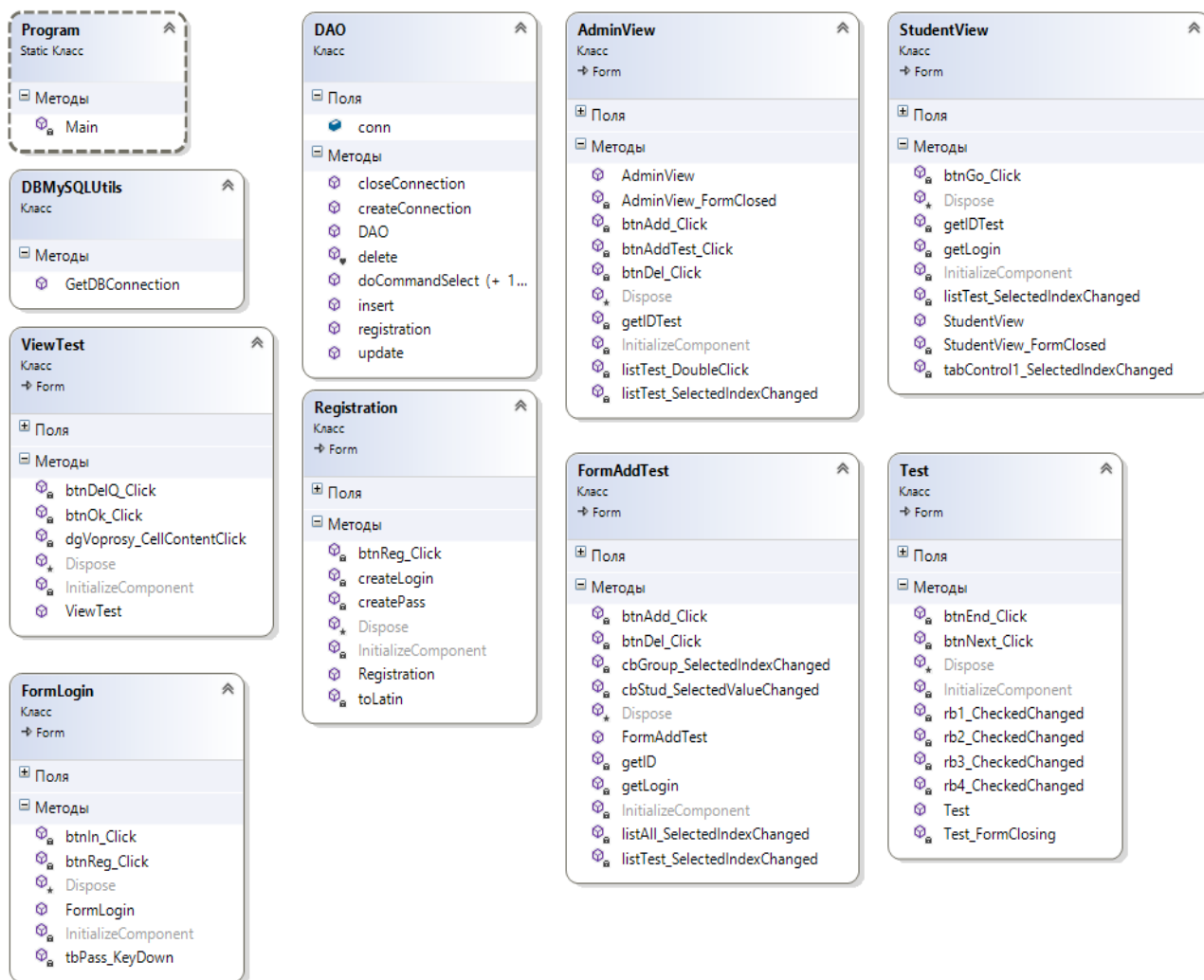


Рисунок 13 - Диаграмма классов

Выводы по параграфу

В данном параграфе приведена и описана физическая модель ИС, представлены классы и описаны функции для различных пользователей.

2.3 Технологическое обеспечение задачи

2.3.1 Организация технологии сбора, передачи, обработки и выдачи информации

Работа с программой начинается с загрузки на экран главной формы приложения для входа в систему.

Для взаимодействия с системой применяются событийный и диалоговый режимы.

Событиями называются процессы двух видов: активизируемые пользователем, а также программные события, на основании которых выполняются процедуры контроля допустимости данных.

Диалогом называется «предоставление пользователю нескольких альтернатив и обработка его выбора» [14].

В основе разработанной системы лежит «интерактивный режим ввода данных, исходными ресурсами которого являются файлы оперативной информации, подгружаемые по запросу» [14].

2.3.2 Схема технологического процесса сбора, передачи, обработки и выдачи информации

На рисунке 14 изображена схема представления технологического обеспечения [14].



Рисунок 14 - Схема выбора, передачи, обработки и выдачи информации

Выводы по параграфу

В данном параграфе приводится описание технологического обеспечения задачи.

2.4 Контрольный пример реализации проекта и его описание

В процессе тестирования приложения доказывается корректность его работы. Для этого выполняется проверка работы всех возможных сценариев.

Первая форма приложения, которую видит пользователь – форма авторизации (рисунок 15).

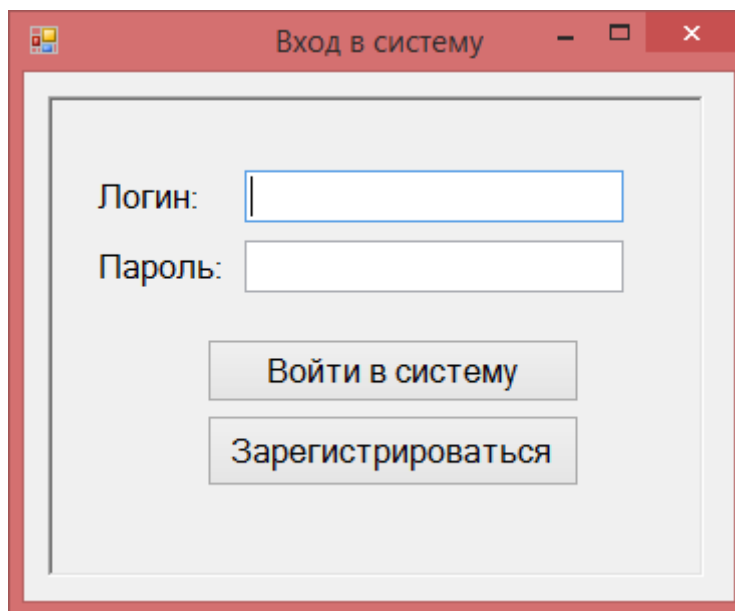
The image shows a screenshot of a web application's login form. The window title is "Вход в систему" (Login). The form has a light gray background and is enclosed in a red border. It contains two input fields: "Логин:" (Login) and "Пароль:" (Password). Below the fields are two buttons: "Войти в систему" (Login) and "Зарегистрироваться" (Register).

Рисунок 15 - Форма авторизации

Используя эту форму, пользователь может перейти на процесс регистрации в системе (рисунок 16), либо на работу с системой.

В случае успешной регистрации система добавляет пользователя в имеющуюся базу данных, а также сообщает ему логин и пароль (рисунок 17).

Для генерации логина используются первые буквы фамилии, имени и отчества студента, а также времени регистрации.

Регистрация

Фамилия Скворцов

Имя Владимир

Отчество Петрович

Группа 27МИ1

Зарегистрироваться

Рисунок 16 - Форма регистрации

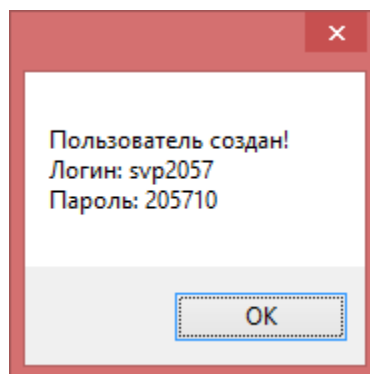


Рисунок 17 - Успешная регистрация

После получения логина и пароля студент может войти в систему. В том случае, если введенные данные будут неверны, система выдаст ошибку (рисунок 18).

Для того чтобы войти в систему под учетной записью администратора, необходимо использовать логин «admin» и пароль «admin».

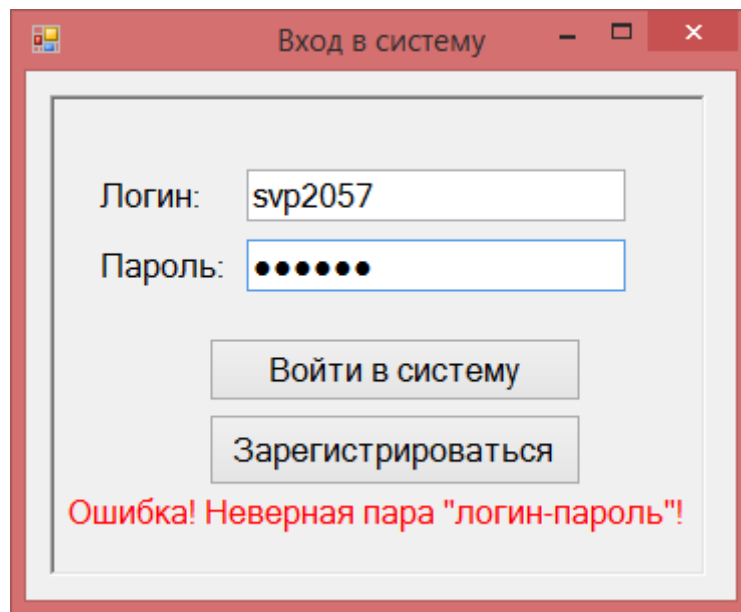


Рисунок 18 - Ошибка входа в систему

Интерфейс учетной записи администратора представлен на рисунке 19.

Данная форма содержит компонент label, содержащий текст: «Вы вошли как: Администратор» и компонент TabPage, включающий в себя две вкладки: «Тесты» и «Статистика».

На вкладке «Тесты» в виде списка (ListBox) выведены все тесты, имеющиеся в системе в данный момент времени.

Двойной клик по тесту открывает его с возможностью редактирования (рисунок 20).

При выборе конкретной ячейки ее значение переносится в окно для редактирования, расположенное в нижней части экрана. После внесения изменений и нажатия кнопки «Применить изменения» данные сохраняются в базе данных и отображаются в таблице (рисунок 21).

Кнопка «Удалить вопрос» удаляет вопрос не только из таблицы, отображаемой администратору, но и из базы данных.

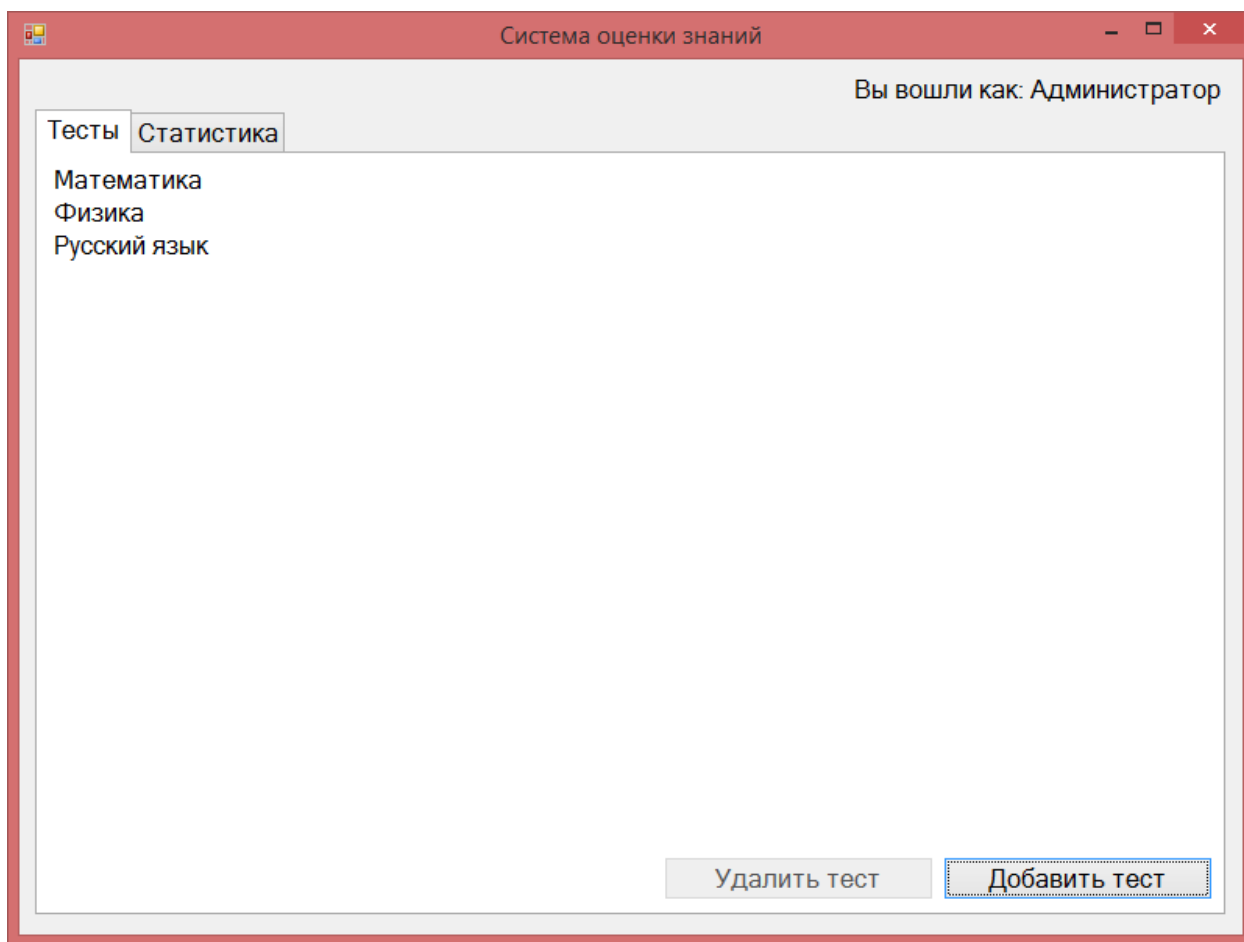


Рисунок 19 - Интерфейс учетной записи «Администратор»

Кнопка добавления теста («Добавить тест») создает стандартный диалог открытия файла с фильтром доступным файлов в формате Excel. После успешного добавления теста система выдает соответствующее сообщение (см. рисунок 22).

Интерфейс вкладки «Статистика» приведен на рисунке 23.

Данная вкладка отображает результаты студентов по пройденным тестам. Кроме того, с этой вкладки у администратора имеется возможность назначения/удаления тестов студентам (рисунок 24).

На данной форме администратору необходимо выбрать группу, затем конкретного студента, после чего в списке назначенных тестов отобразятся тесты, которые уже назначены студенту.

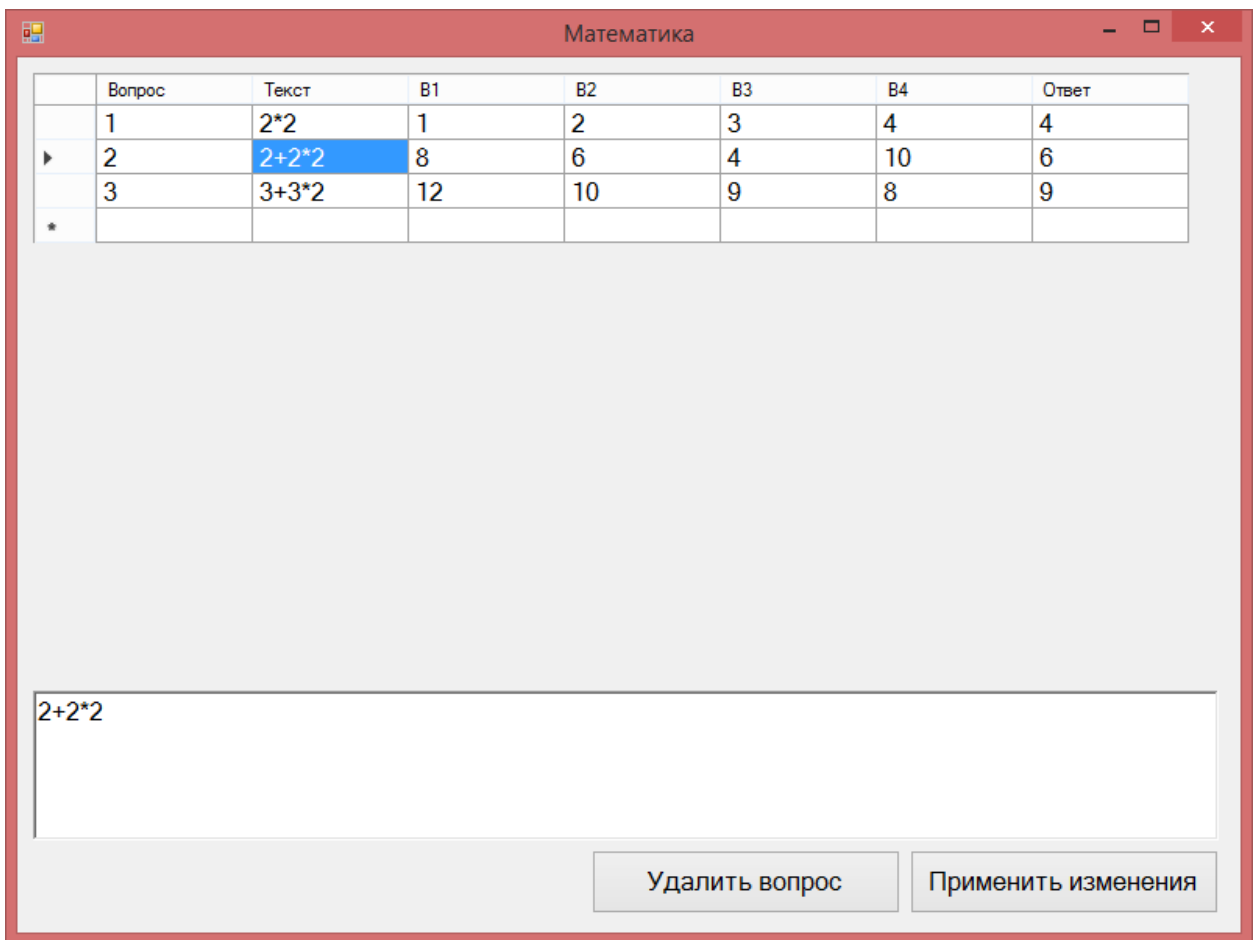


Рисунок 20 - Просмотр и редактирование теста

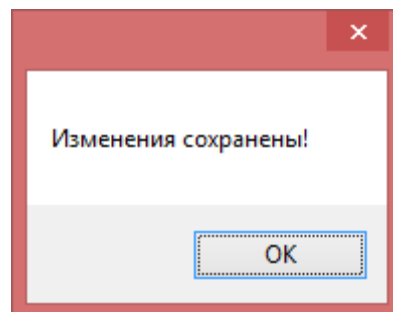


Рисунок 21 - Корректировка вопросов

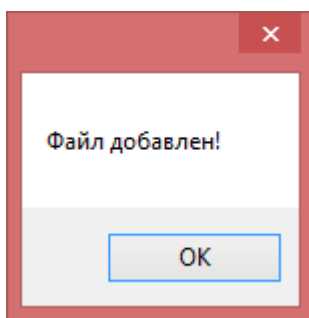


Рисунок 22 - Результат успешного добавления теста

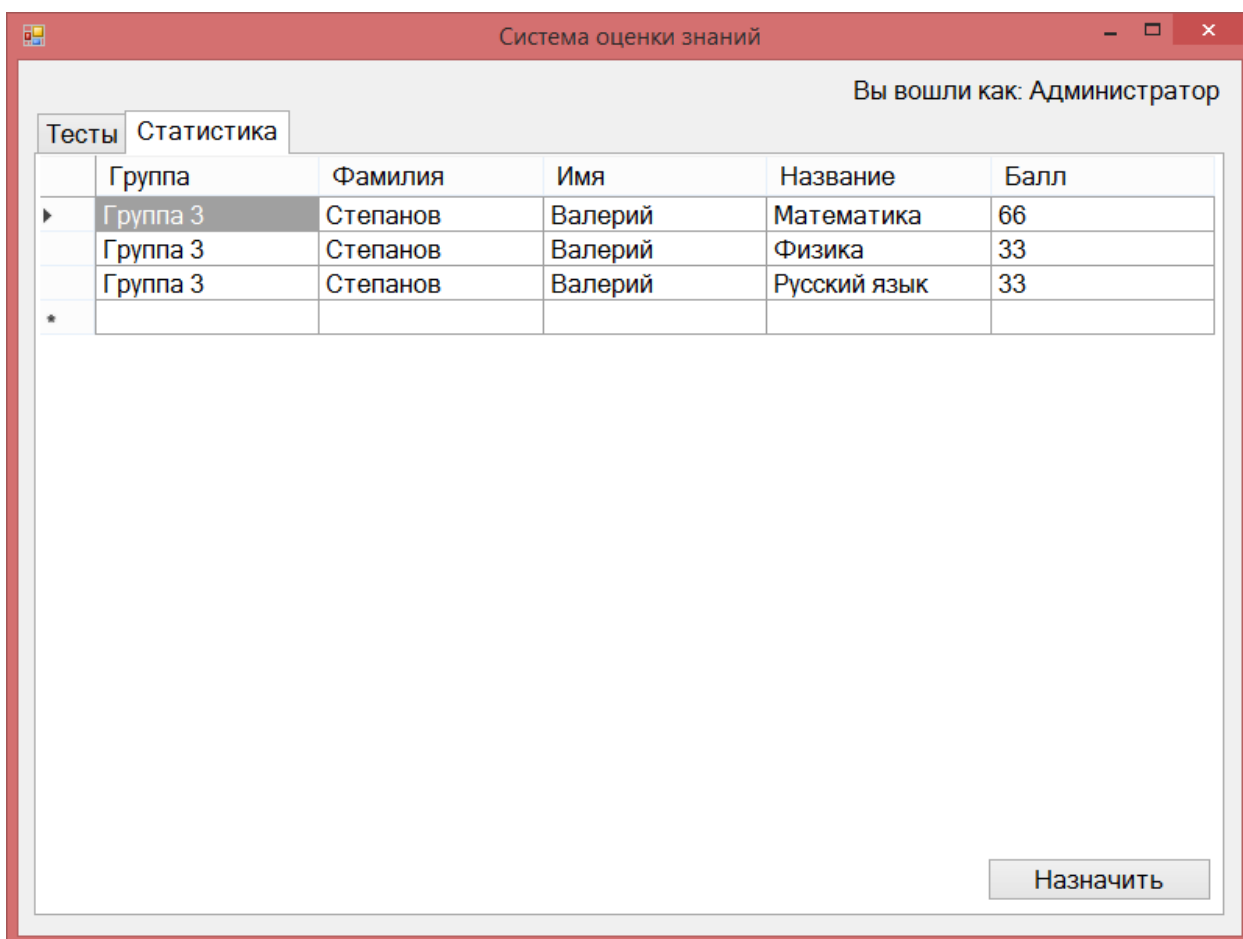


Рисунок 23 - Интерфейс администратора «Статистика»

Назначение тестов

Группа: Студент

Группа 1 Иванов И.И.

Все тесты: Назначенные тесты:

Математика Математика
Физика Физика
Русский язык Русский язык
Тест Добавления

Назначить Удалить

Рисунок 24 - Форма назначения тестов

В том случае, если администратор захочет добавить тест, уже существующий в списке студента, система выдаст ошибку (рисунок 25).

В случае удаления теста, тест пропадает из списка назначенных и из базы данных.

Интерфейс учетной записи «Студент» приведен на рисунке 26.

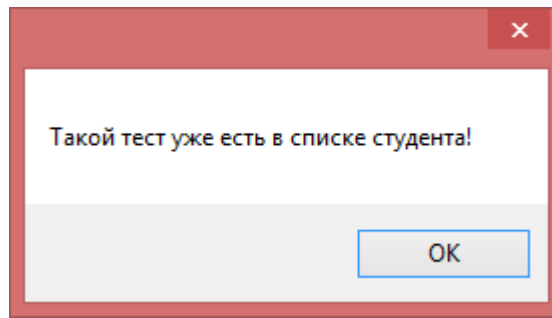


Рисунок 25 - Ошибка назначения теста

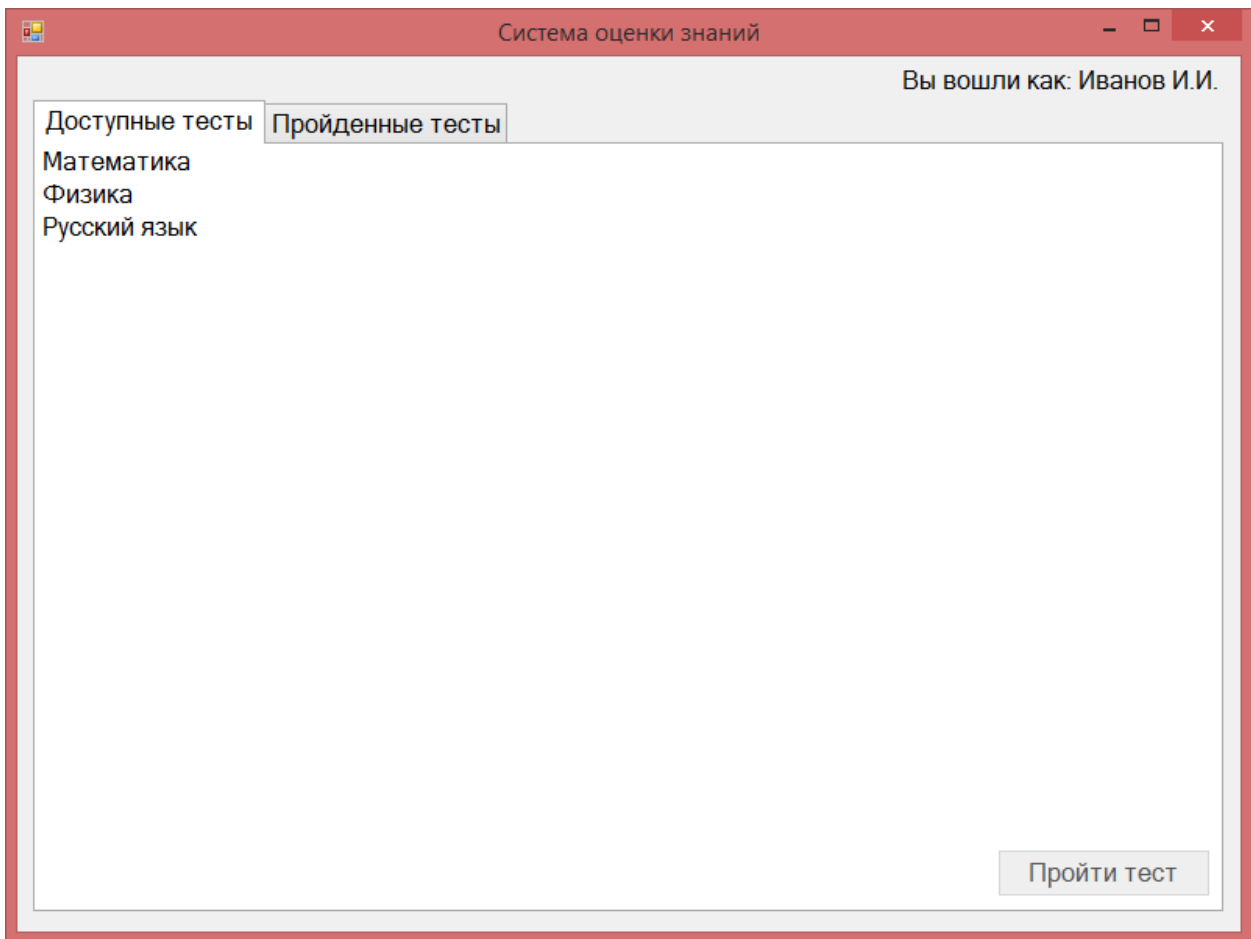


Рисунок 26 - Интерфейс учетной записи «Студент»

Данная форма содержит компонент label, содержащий текст: «Вы вошли как: Администратор» и компонент TabPage, включающий в себя две вкладки: «Доступные тесты» и «Пройденные тесты».

На вкладке «Доступные тесты» отображается список тестов, необходимых для прохождения. Чтобы запустить тест, студенту необходимо выбрать его и нажать на кнопку «Пройти тест», в результате чего появится новая форма с тестом (см. рисунок 27).

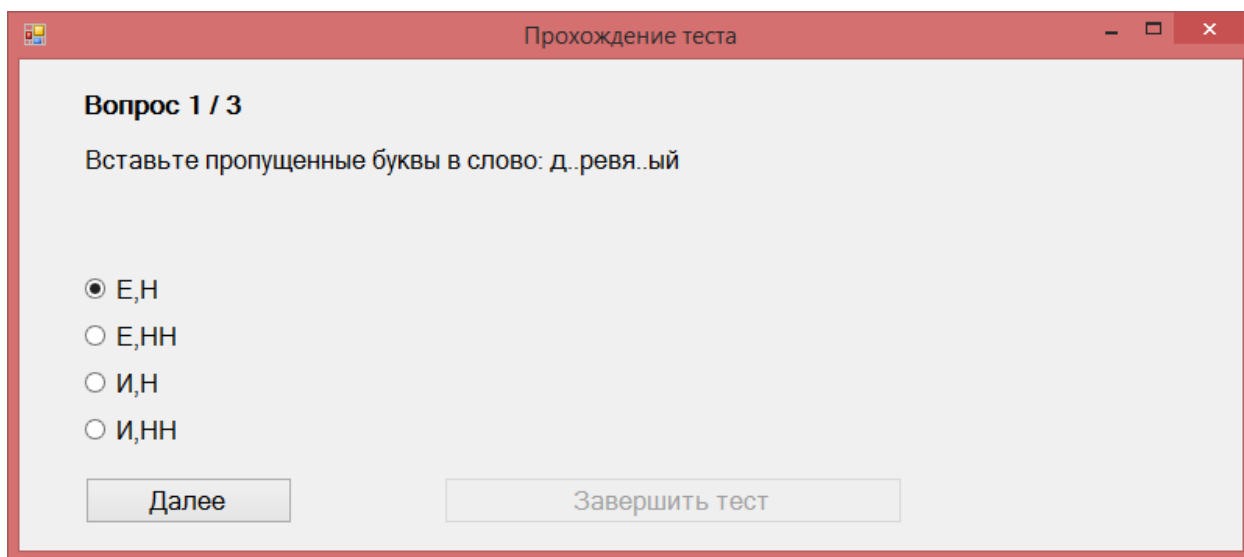


Рисунок 27 - Форма прохождения теста

Данная форма отражает порядковый номер вопроса в тесте, общее число вопросов, а также возможные варианты ответов.

По завершении теста студент получает сообщение о результатах прохождения, которые отражают информацию об общем количестве вопросов в тесте, а также о количестве правильных ответов (рисунок 28).

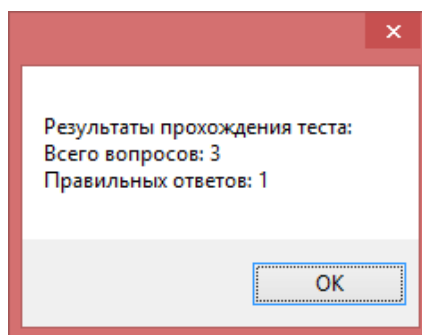
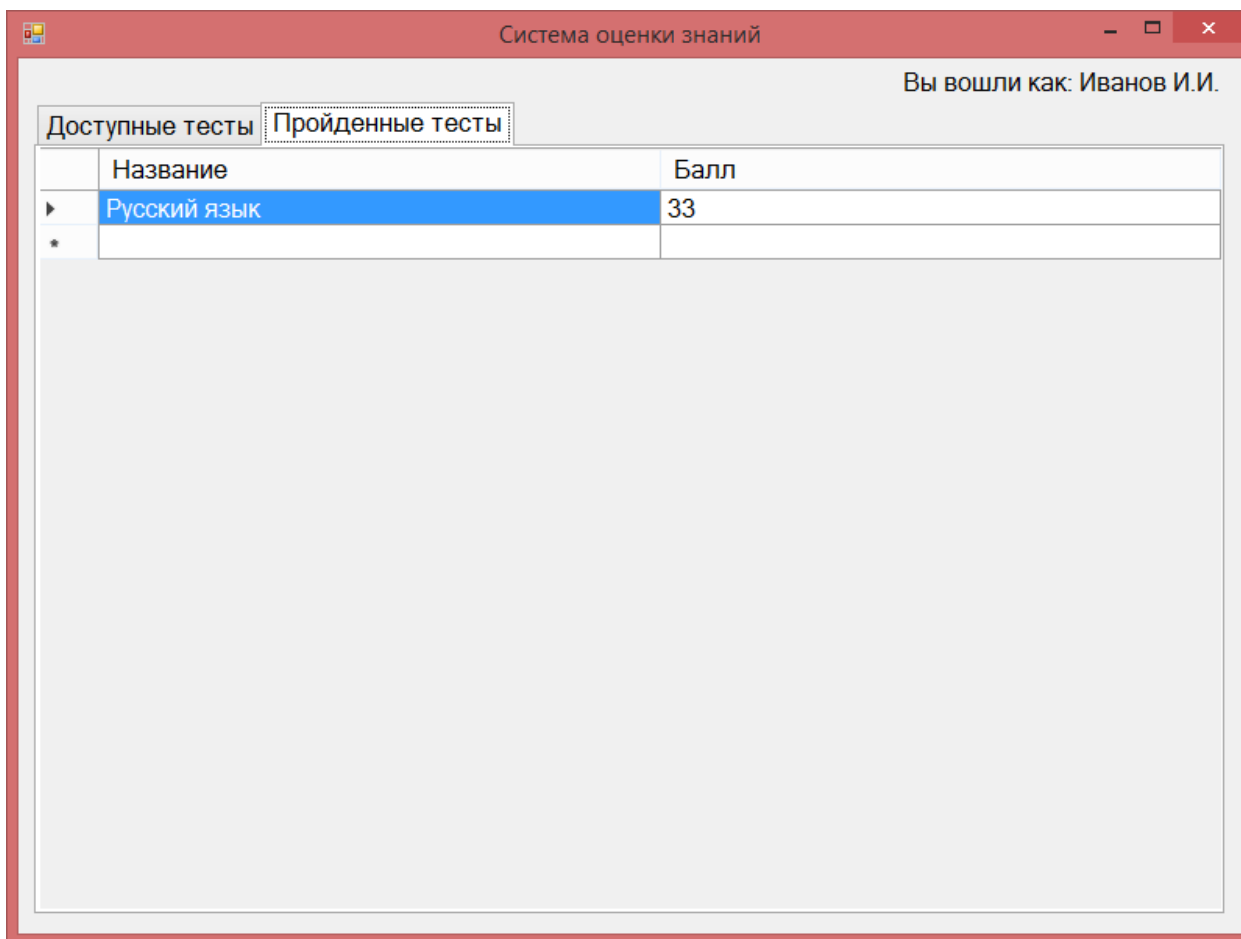


Рисунок 28 - Результаты прохождения теста

После завершения теста данные о прохождении автоматически добавляются в базу данных и становятся доступными на вкладке «Пройденные тесты» (рисунок 29). Результаты тестов рассчитываются в процентах.



	Название	Балл
▶	Русский язык	33
*		

Рисунок 29 - Результаты пройденных тестов

Выводы по параграфу

В параграфе приведено описание работы приложения и продемонстрирован контрольный пример работы.

Выводы по главе 2

В рамках данной главы приводится описание процессов разработки и реализации информационной системы.

Глава 3 Оценка и обоснование экономической эффективности проекта

3.1 Выбор и обоснование методики расчета экономической эффективности

Трудоемкость работ по разработке информационной системы «рассчитывается с учетом срока завершения работ, выбранного языка программирования, а также объема реализуемых функций. Выбор комплекса работ по разработке информационной системы определяется согласно ГОСТ 19.102-77 «Единая система программной документации». Руководитель формирует постановку задачи и отвечает за работу по созданию системы. Исполнитель отвечает за проектирование информационного и методического обеспечения, разрабатывает программное обеспечение, отвечает за функциональность системы. Трудоемкость создания новой информационной системы оценивается на основе трудоемкости разработки аналогичного программного продукта с учетом характерных особенностей проекта, отражаемых специальными поправочными коэффициентами» [50].

3.2 Расчет показателей экономической эффективности проекта

Затраты на разработку проекта Q_p вычисляются по формуле:

$$Q_p = \sum t_i \quad (1)$$

«где, t_i – i -й этап проекта» [50].

Для вычисления чел-часов, необходимых на реализацию, необходимо количество чел-дней умножить на рабочее время:

$$Q_p = 68 \text{ чел-дней} = 544 \text{ чел-часов.}$$

Основные этапы проекта отражены в таблице 5.

Таблица 5 - Этапы проекта

Этап	№ работы	Содержание работы	Трудоемкость	
			(чел-дни)	(чел-час)
«Изучение архитектуры компании»	1	Изучение бизнес-процессов компании	3	24
	2	Изучение используемого программного обеспечения	7	56
Проектирование базы данных	1	Сбор информации о сущностях	3	24
	2	Построение модели БД	4	32
Разработка программного продукта	1	Сбор первоначальных требований	3	24
	2	Выбор инструментов для разработки	1	8
	3	Проектирование будущего программного продукта	7	56
	4	Написание программного кода	21	168
Тестирование полученного программного продукта	1	Подготовка тестов для системы	3	24
	2	Проведение тестов для функций системы и сбор результатов тестирования	7	56
	3	Анализ результатов тестирования	2	16
	4	Внесение изменений в программный код (при необходимости)» [13]	7	56

«Количество исполнителей, необходимых для реализации проекта, вычисляется по формуле» [50]:

$$N = \frac{Q_p}{F}, \quad (2)$$

«где Q_p – затраты труда на выполнение проекта» [50],

« F – фонд рабочего времени, вычисляемый по формуле» [50]:

$$F = T \cdot F_M, \quad (3)$$

«где T – количество месяцев, затраченных на реализацию проекта,

F_M – фонд времени в текущем месяце, вычисляемый по формуле» [50]:

$$F_M = \frac{t_p \cdot (D_K - D_B - D_{\Pi})}{12} \quad (4)$$

«где t_p – продолжительность рабочего дня,

D_K – число дней в году,

D_B – число выходных дней в году,

D_{Π} – число праздничных дней в году» [13].

$$F_M = \frac{8(365-118)}{12} \approx 165.$$

$$F = 2 * 165 = 330.$$

Здесь 2 – количество месяцев, затраченное на выполнение проекта.

$$\text{Количество исполнителей } N = \frac{544}{330} \approx 1.$$

В таблице 6 отражены данные о работах, необходимых для построения сетевой модели и времени, затрачиваемом на выполнение указанных работ.

Таблица 6 – Работы проекта

Событие	Код работы	Работа	t, чел-дни
«Начало проекта	0-1	Изучение бизнес-процессов компании	3
Бизнес-процессы изучены	1-2	Изучение используемого программного обеспечения	7
Используемое программное обеспечение изучено	2-3	Сбор информации о сущностях	3
Данные о сущностях системы собраны	3-4	Сбор первоначальных требований	4
Требования собраны	4-5	Построение модели БД	3
Модель базы данных построена	3-6	Выбор инструментов для разработки	1
Инструменты для разработки системы определены	6-7	Проектирование клиентской части системы	7
Проект клиентской части составлен	7-8	Написание программного кода» [13]	21

Продолжение таблицы 6

«Событие	Код работы	Работа	t, чел-дни
Написание программного кода завершено	3-9	Подготовка тестов для системы	3
Получены результаты тестирования системы	8-9	Анализ результатов тестирования	2
Проведен анализ результатов тестирования	9-10	Внесение изменений в программный код (при необходимости)	7
Изменения в код внесены (при необходимости)	10-11	Ожидание внесения изменений» [13]	0

На рисунке 30 представлен график сетевой модели.

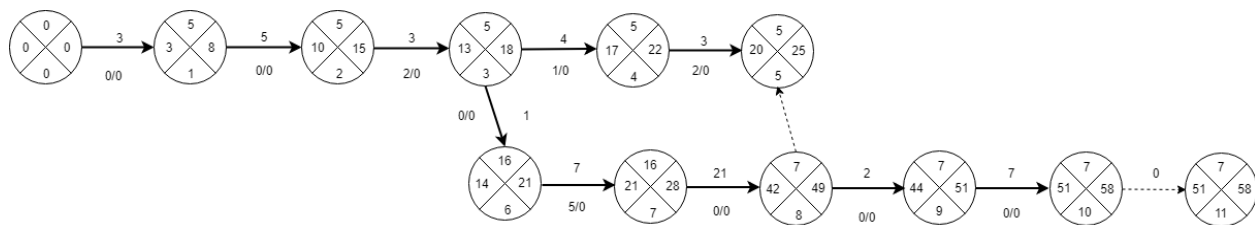


Рисунок 30 - Сетевая модель

Критический путь модели проходит через вершины:

0-1-2-3-4-5-6-7-8-9-10.

Для получения данных, представленных на графике определены ранние и поздние сроки совершения событий, резервы времени событий.

Ранние сроки наступления событий:

$$T_0^p = 0 \quad (5)$$

$$T_1^p = 0 + 3 = 3 \quad (6)$$

$$T_2^p = 3 + 7 = 10 \quad (7)$$

$$T_3^p = 10 + 3 = 13 \quad (8)$$

$$T_4^p = 13 + 4 = 17 \quad (9)$$

$$T_5^p = 17 + 3 = 20 \quad (10)$$

$$T_6^p = 13 + 1 = 14 \quad (11)$$

$$T_7^p = 14 + 7 = 21 \quad (12)$$

$$T_8^p = 21 + 21 = 42 \quad (13)$$

$$T_9^p = 42 + 2 = 44 \quad (14)$$

$$T_{10}^p = 44 + 7 = 51 \quad (15)$$

$$T_{11}^p = 51 + 0 = 51 \quad (16)$$

Поздние сроки наступления событий:

$$T_{11}^n = 58 \quad (17)$$

$$T_{10}^n = 58 \quad (18)$$

$$T_9^n = 58 - 7 = 51 \quad (19)$$

$$T_8^n = 51 - 2 = 49 \quad (20)$$

$$T_7^n = 49 - 21 = 28 \quad (21)$$

$$T_6^n = 28 - 7 = 21 \quad (22)$$

$$T_5^n = 25 \quad (23)$$

$$T_4^n = 25 - 3 = 22 \quad (24)$$

$$T_3^n = 22 - 4 = 18 \quad (25)$$

$$T_2^n = 18 - 3 = 15 \quad (26)$$

$$T_1^n = 15 - 7 = 8 \quad (27)$$

$$T_0^n = 0 \quad (28)$$

На рисунке 31 представлена диаграмма Ганта, отображающая календарный график работ.

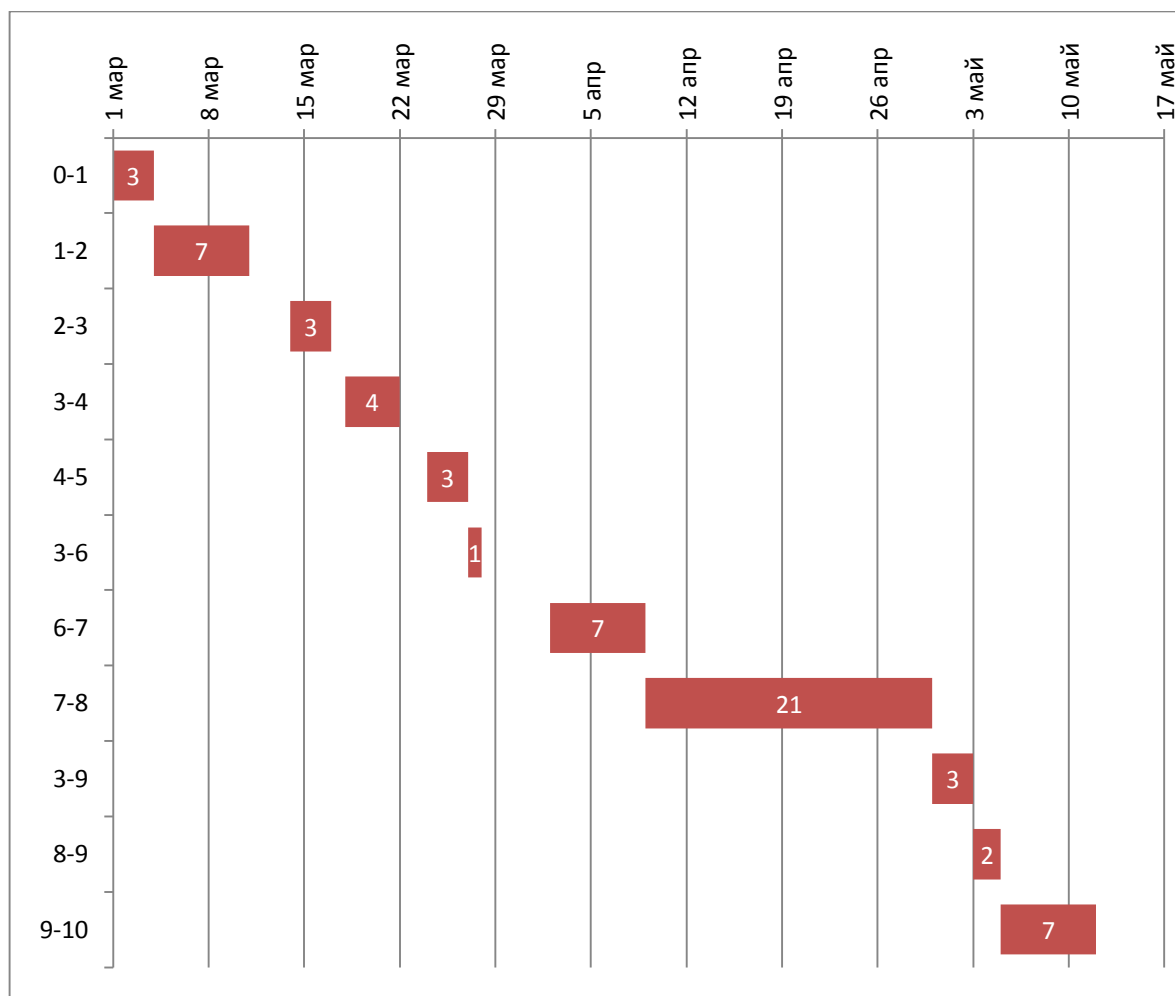


Рисунок 31 - Диаграмма Ганта

«Затраты на выполнение проекта вычисляются по формуле:

$$K = C_{\text{ЗАРП}} + C_{\text{ОБ}} + C_{\text{ОРГ}} + C_{\text{НАКЛ}} \quad (29)$$

где, K – суммарные затраты» [50],

$C_{\text{ЗАРП}}$ – «зарплата исполнителей, вычисляемая по формуле» [50]:

$$C_{\text{ЗАРП}} = C_{\text{З.ОСН}} + C_{\text{З.ДОП}} + C_{\text{З.ОТЧ}} \quad (30)$$

«где, $C_{\text{З.ОСН}}$ и $C_{\text{З.ДОП}}$ – основная и дополнительная заработная плата соответственно, $C_{\text{З.ОТЧ}}$ – отчисления с заработной платы» [50].

«Для определения $C_{\text{З.ОСН}}$ используется формула:

$$C_{\text{З.ОСН}} = T_{\text{ЗАН}} \cdot O_{\text{ДН}} \quad (31)$$

где $T_{\text{ЗАН}}$ – количество дней, отработанных исполнителем» [50],

$O_{\text{ДН}}$ – «дневной оклад, вычисляемый по формуле:

$$O_{\text{ДН}} = \frac{O_{\text{МЕС}} \cdot 8}{F_{\text{М}}} \quad (32)$$

где $O_{\text{МЕС}}$ – месячный оклад,

$F_{\text{М}}$ – фонд рабочего времени» [50].

Данные о затратах на основную заработную плату приведены в таблице 7.

Таблица 7 – Основная заработная плата

Должность	Оклад	Дн. оклад	Труд. затраты	Зар. плата
Программист	18000	986,2	68	104614,2

$$O_{\text{МЕС}} = 18000 \cdot 1,13 = 20340 \text{ р.} \quad (33)$$

$$O_{\text{ДН}} = \frac{20340 \cdot 8}{165} \approx 986,2 \text{ р.}$$

$$C_{\text{З.ОСН}} = 986,2 \cdot 68 = 67060,36 \text{ р.} \quad (34)$$

Дополнительная зарплата определяется по формуле:

$$C_{\text{З.ДОП}} = 0,2 \cdot 67060,36 = 13412,07 \text{ р.} \quad (35)$$

Величина отчислений определяется по формуле:

$$C_{\text{З.ОТЧ}} = (67060,36 + 13412,07) \cdot 0,3 = 24141,73 \text{ р.} \quad (36)$$

Где 0,3 – размер единого социального налога в 2021 году.

$$C_{\text{ЗАРП}} = 67060,36 + 13412,07 + 24141,73 = 104614,2 \text{ р.} \quad (37)$$

$C_{\text{ОБ}}$ – «затраты на оборудование» [50],

$$C_{\text{ОБ}} = 6000 \text{ р.} \quad (38)$$

$C_{\text{ОРГ}}$ – «затраты на организацию рабочих мест, вычисляются по формуле» [50]:

$$C_{\text{ОРГ}} = \frac{C_{\text{КВМ}}}{12} S \frac{T_{\text{АР}} \cdot 8}{F_{\text{М}}} \quad (40)$$

где $C_{\text{КВМ}}$ – «стоимость аренды квадратного метра площади за год» [50],

S – «площадь рабочего помещения» [50],

T_{AP} – «срок аренды» [50].

$$C_{ОРГ} = 1041 \cdot 40 \cdot \frac{68 \cdot 8}{165} = 137285,8 \text{ р.} \quad (41)$$

$C_{НАКЛ}$ – «накладные расходы, определяемые по формуле» [50]:

$$C_{НАКЛ} = 0.6 \cdot C_{З.ОСН} \quad (42)$$

$$C_{НАКЛ} = 0.6 \cdot 67060,36 = 40236,22 \text{ р.}$$

«Суммарные затраты на выполнение проекта» [50] равны»:

$$K = 67060,36 + 6000 + 137285,8 + 40236,22 = 250582,4 \text{ р.} \quad (43)$$

«Затраты на внедрение разработанной системы вычисляются по формуле» [50]:

$$K_{ВН} = C_{ВН.ЗАРП} + C_{ВН.ОБ} + C_{ВН.ОРГ} + C_{ВН.НАКЛ} \quad (44)$$

«где $C_{ВН.ЗАРП}$ – заработная плата исполнителям, участвующим во внедрении,

$C_{ВН.ОБ}$ – затраты на оборудование,

$C_{ВН.ОРГ}$ – затраты на обеспечение рабочих мест сотрудников, работающих с внедренным программным обеспечением,

$C_{ВН.НАКЛ}$ – накладные расходы при внедрении» [50].

В силу того, что для использования внедряемой системы не предполагается закупка нового оборудования и модернизация рабочих мест, параметры $C_{ВН.ОБ}$ и $C_{ВН.ОРГ}$ равны нулю.

$$\begin{aligned} C_{ВН.ЗАРП} &= 986,2 \cdot 22 + 0,2 \cdot 986,2 \cdot 22 + \\ &+ (986,2 \cdot 22 + 0,2 \cdot 986,2 \cdot 22) \cdot 0,3 = 33845,76 \text{ р.} \end{aligned} \quad (45)$$

где 22 дня – время внедрения проекта.

$$C_{ВН.НАКЛ} = 986,2 \cdot 22 \cdot 0,6 = 13017,6 \text{ р.} \quad (46)$$

$$K_{ВН} = 33845,76 + 0 + 0 + 13017,6 = 46863,36 \text{ р.} \quad (47)$$

Общие затрат включают в себя затраты на разработку и затраты на внедрение:

$$K_{OB} = K_{ВН} + K$$

$$K_{OB} = 46863,36 + 250582,4 = 297445,8 \text{ р.} \quad (48)$$

«Стоимость программного обеспечения определяется по формуле:

$$K_{ПО} = (\Delta K + K_{ВН}) \cdot (1 + D_{ПРИБ}) \quad (49)$$

где ΔK – стоимость одной копии программы,

$K_{ВН}$ – затраты на внедрение программного обеспечения» [50],

$D_{ПРИБ}$ – «процент прибыли от продажи программного обеспечения, заложенный в его стоимость, вычисляемый по формуле» [50]:

$$D_{ПРИБ} = \left(\frac{K_{ПР}}{\Delta K + K_{ВН}} - 1 \right) \cdot 100\% \quad (50)$$

$$D_{ПРИБ} = 15\%.$$

«Для определения прибыли, получаемой от продажи каждой установки программного продукта, используется формула» [50]:

$$C_{ПРИБ} = K_{ПР} \cdot D_{ПРИБ} \cdot (1 - N_{НДС}) \quad (51)$$

где $N_{НДС}$ – величина налога на добавочную стоимость (в процентах).

$$C_{ПРИБ} = 9000 \text{ р.} \quad (52)$$

В таблице 8 «представлен фрагмент таблицы расходов и прибыли по проекту» [50].

Таблица 8 – Фрагмент таблицы расходов и прибыли по проекту

Период расчета	Разработка (проект)				Прибыль	
	Частичная стоимость	Сальдо начальное по кредиту	Погашение кредита	Сальдо конечное по кредиту	Расчетная прибыль	Чистая прибыль
1-4. 200х	88906,12	-320062,03	88906,12	-231155,9	13779,38	11023,51
5-8. 200х	88906,12	-231155,9	88906,12	-142249,79	13779,38	11023,51
9-12. 200х	88906,12	-142249,79	88906,12	-53343,67	13779,38	11023,51
1-3. 200х	88906,12	-53343,67	55006,12	0	49341,83	39473,47
4-6. 200х	88906,12	0	0	0	102685,5	82148,4

Выводы по параграфу

Срок выполнения проекта составляет 68 дней, суммарные затраты на проект составляют 297445,8 р.

В результате проведенных расчетов можно сделать вывод о том, что спроектированная система является экономически целесообразным решением задачи автоматизации.

Выводы по главе 3

В рамках данной главы посчитана себестоимость проекта и сделан вывод о разумности разработки автоматизированной информационной системы.

Заключение

В рамках выполнения данной выпускной квалификационной работы было создано программное обеспечение оценки знаний студентов.

Затронутая тема является очень актуальной, т.к. на сегодняшний день высшее образование является одним из ключевых моментов при устройстве на работу в любой сфере человеческой деятельности. В большинстве случаев работодателю не столько важно наличие диплома, сколько наличие знаний. Автоматизированные системы тестирования существенно упрощают процесс проверки остаточных знаний – они способны за ограниченный промежуток времени провести опрос по многим областям знаний. Кроме того, одним из преимуществ подобных систем является адекватность и объективность их оценки – у компьютера отсутствует предвзятое отношение к тестируемому, он не способен завышать или занижать реально полученный балл.

Недостатком систем оценки знаний является факт того, что, как и любые программные продукты, они могут быть взломаны и обмануты. В истории использования автоматизированных систем тестирования известно немало подобных случаев.

Еще одним из недостатков является возможность «списывания» - в век информационных технологий и глобального распространения сети Интернет студенту очень просто воспользоваться возможностями поисковых систем, особенно в тех случаях, когда тестирование ведется через клиент-серверное приложение. Отсюда вытекает преимущество оффлайн систем – их использование возможно на компьютерах, не подключенных к сети, что сокращает возможность обмана системы.

В рамках данной работы была разработана именно такая оффлайн-система, реализующая возможность тестирования студентов.

В основе системы лежит база данных, разработанная при помощи СУБД MySQL. Спроектированная база данных хранит в себе минимум информации:

- идентификационные данные студентов;
- перечень тестов;
- списки тестов, назначенных студентам;
- результаты тестирования.

В качестве среды разработки самого программного приложения была выбрана среда Microsoft Visual Studio 2013 и язык программирования C#, который позволяет легко и быстро создавать приложения с привычным пользователю интерфейсом системы MS Windows.

В рамках данного приложения были созданы следующие экраны:

- Login – форма входа в систему;
- Registration – форма регистрации в системе;
- AdminView – форма учетной записи администратора;
- StudentView – форма учетной записи студента;
- AddTest – форма назначения тестов студентам (доступна только под учетной записью администратора);
- Test – форма прохождения теста (доступна только под учетной записью студента);
- ViewTest – форма просмотра и редактирования тестов (доступна только под учетной записью администратора).

В качестве учетных данных для администратора были выбраны следующие:

- логин – admin;
- пароль – admin.

Эти данные не хранятся в разработанной базе данных и могут быть изменены только программным путем.

Одним из достоинств данной программы является поддержка загрузки тестов из файлов формата Excel, что является очень удобной возможностью, т.к. для добавления тестов в систему не придется набирать их вручную.

Однако, имеются и небольшие недостатки, которые не были отражены в техническом задании и могут восприниматься как дальнейшее развитие программы. Например, ограничение тестов по времени.

Подводя итог, можно сделать вывод, что разработанная система полностью удовлетворяет поставленным требованиям и может быть использована не только в учебных заведениях, но и в любых других сферах деятельности, где необходима проверка знаний.

Список используемой литературы и используемых источников

1. Адигеев М.Г. Жизненный цикл программного обеспечения. – Ростов-на-Дону: Изд-во ЮФУ, 2013. – 41 с.
2. Албахари Дж. С# 6.0. Справочник. Полное описание языка / Дж. Албахари, Б. Албахари. – М.: Вильямс, 2016. – 1040 с.
3. Альпатов А.В. Разработка программы, предназначенной для оценки знаний студентов. – Воронеж: Изд-во Воронежского института высоких технологий, 2018 – 7 с.
4. Андрианова А.А. Алгоритмизация и программирование. Практикум / А.А. Андрианова, Л.Н. Исмагилов, Т.М. Мухтарова. – СПб.: Изд-во Лань, 2019. – 240 с.
5. Бабанов А.М. Технология разработки программного обеспечения: структурный подход. – Томск: ТГУ, 2016. – 157 с.
6. Балакин А.О Разработка WPF приложений С#. – Екатеринбург: Изд-во РГППУ ИИПО, 2018. – 58 с.
7. Беляков А.Ю. Объектно-ориентированное программирование. – Пермь: ИЦЦ «Прокрость», 2017. – 88 с.
8. Берг Д.Б. Модели жизненного цикла / Д.Б. Берг, Е.А. Ульянова, П.В. Добряк. – Екатеринбург: Изд-во Урал. ун-та, 2014. – 74 с.
9. Биллинг В.А. Основы объектного программирования на С#. – Москва: Бином. Лаборатория знаний, 2016. – 584 с.
10. Волосевич А.А. Средства платформы .NET для работы с базами данных: метод. пособие по дисц. «Инструменты и средства программирования» и «Избранные главы информатики» для студ. спец. 1-31 0304 «Информатика» всех форм обуч. / А.А. Волосевич, С.В. Актанорович. – Минск: БГУИР, 2015. – 52 с.

11. Гвоздева В.А. Информатика, автоматизированные информационные технологии и системы. - М.: ИНФРА-М, 2013. - 544 с.
12. Грекул В.И. Проектирование информационных систем / В.И. Грекул, Н.Л. Коровкина, Г.А. Левочкина. – М.: Издательство Юрайт, 2017. – 385 с.
13. Григораш О.В. Тестовые задания – один из эффективных способов оценки качества знаний студентов; Краснодар: Кубанский государственный аграрный университет. Научный журнал КубГАУ, № 101 (07), 2014. – 17 с.
14. Громов Ю.Ю. Технология программирования / Ю.Ю. Громов, О.Г. Иванова, М.П. Белев, Ю.В. Минин. – Тамбов: Изд-во ФГБОУ ВПО «ТГТУ», 2013. – 172 с.
15. Демина А.В. Базы данных. Эффективная работа в Access. Использование языка структурированных запросов SQL; Саратовский государственный социально-экономический университет. – Саратов, 2013. – 44 с.
16. Долженко А.И. Управление информационными системами. – Ростов-на-Дону: Изд-во РГУ, 2017. – 191 с.
17. Жданов А.И. Система балльно-рейтинговой оценки академической успешности и компетентностной сформированности обучающегося в высшем учебном заведении / А.И. Жданов, И.В. Юргелас, Р.Н. Фролов. – Современные наукоемкие технологии №6, 2017. – 6 с.
18. Киреенко В.П. Система управления базами данных Microsoft Access: метод. рекомендации / В.П. Киреенко, Н.Н. Поснов. – Минск: ГИУСТ БГУ, 2013. – 100 с.
19. Коцюба И.Ю. Основы проектирования информационных систем / И.Ю. Коцюба, А.В. Чунаев, А.Н. Шиков. – СПб.: Университет ИТМО, 2015. – 206 с.

20. Краснянский М.Н. Проектирование информационных систем управления документооборотом научно-образовательных учреждений / М.Н. Краснянский, С.В. Карпушкин, А.В. Остроух. – Тамбов: Изд-во ФГБОУ ВПО «ТГТУ», 2015. – 216 с.
21. Кудрина Е.В. Основы алгоритмизации и программирования на языке С# / Е.В. Кудрина, М.В. Огнева. – М.: Изд-во Юрайт, 2019. – 322 с.
22. Кумагина Е.А. Модели жизненного цикла и технологии проектирования программного обеспечения / Е.А. Кумагина, Е.А. Неймарк. – Нижний Новгород: Изд-во ННГУ, 2016. – 41 с.
23. Кумскова И.А. Базы данных. – М.: КНОУРС, 2016. – 488 с.
24. Лученецкая-Бурдина И.Ю. Контроль знаний студентов в системе электронного обучения / И.Ю. Лученецкая-Бурдина, А.А. Федотова. – Ярославский педагогический вестник №3, 2017. – 5 с.
25. Маряшина И.В. Балльно-рейтинговая система контроля и оценки знаний и умений учащихся по физике как средство достижения новых образовательных результатов; Казань, ФГАОУ ВПО Казанский федеральный университет, 2014. – 300 с.
26. Медведев М.А. Программирование на СИ# / М.А. Медведев, А.Н. Медведев. – Екатеринбург: Изд-во Урал. ун-та, 2015. – 64 с.
27. Мезенцев, К.Н. Автоматизированные информационные системы. – М.: ИЦ Академия, 2013. - 176 с.
28. Михеева Е.В. Информатика / Е.В. Михеева, О.И. Титова. – М.: Издательский центр «Академия», 2014. – 352 с.
29. Мокеев В.В. Бизнес-информатика / В.В. Мокеев, Е.В. Бунова, О.С. Буслаева. – Челябинск: издательский центр ЮУрГУ, 2015. – 67 с.
30. Никифорова Ю.В. Электронное учебное пособие «Создание проектов Windows Forms в С#». – Екатеринбург: Изд-во РГППУ ИИПО, 2017. – 56 с.

31. Одинцов Б.Е. Информационные системы управления эффективностью бизнеса. - Люберцы: Юрайт, 2015. - 206 с.
32. Пахомов Б.И. С# для начинающих. – СПб.: БХВ-Петербург, 2014. – 432 с.
33. Привалов Н.И. Тестовый контроль знаний студентов / Н.И. Привалов, А.С. Полянина. – Международный журнал прикладных и фундаментальных исследований №4, 2018. – 5 с.
34. Рудаков А.В. Технология разработки программных продуктов. – М.: Академия, 2014. – 190 с.
35. Рыжко А.Л. Информационные системы управления производственной компанией / А.Л. Рыжко, А.И. Рыбников, Н.А. Рыжко. - Люберцы: Юрайт, 2016. - 354 с.
36. Семакин И.Г., Шестаков А.П. Основы программирования: Учебник. - М.: Мастерство, НМЦ СПО; Высшая школа, 2013. - 432 с.
37. Скит Джон. С# для профессионалов: тонкости программирования, 3-е изд.: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2014. – 608 с.
38. Соловей Л.В. Программирование на языке С# / Л.В. Соловей, Н.Н. Мирошниченко, Н.Г. Пономарев. – Харьков: Изд-во НТУ «ХПИ», 2016. – 356 с.
39. Стиллмен Эндрю, Грин Дженнифер. Изучаем С#. 3-е изд. — СПб.: Питер, 2014. — 816 с.
40. Тебайкина Н.И. Применение концепции ITSM при вводе в действие информационных систем. – Екатеринбург: Изд-во Урал. ун-та, 2014. – 72 с.
41. Троелсен Эндрю. Язык программирования С# 5.0 и платформа .NET 4.5, 6-е изд. : Пер. с англ. — М. : ООО “И.Д. Вильямс”, 2013. — 1312 с.
42. Трубилин А.И. Система оценки знаний и рейтингового тестирования студентов / А.И. Трубилин, О.В. Григораш. – Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета №3, 2016. -13 с.

43. Тюгашев А.А. Основы программирования. – СПб.: Университет ИТМО, 2016. – 160 с.
44. Федорова Г.Н. Информационные системы. - М.: ИЦ Академия, 2013. - 208 с.
45. Фленов М.Е. Библия С#. – 3-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2016. – 560 с.
46. Хайруллин Р.С. Программирование на С#. – Казань: Изд-во Казан. гос. архитектур.-строит. ун-та, 2017. – 159 с.
47. Хорев П.Б. Объектно-ориентированное программирование с примерами на С#. – Москва: Форум, Инфра-М, 2016. – 200 с.
48. Шилдт Г. С# 4.0. Полное руководство. – М.: Вильямс, 2015. – 1056 с.
49. Шустова Л.И. Базы данных / Л.И. Шустова, О.В. Тараканов. – М.: Инфра-М, 2016. – 304 с.
50. Ясенев В.Н. Информационные системы и технологии в экономике. - М.: ЮНИТИ, 2014. - 560 с.

Приложение А
Скрипт создания базы данных

```
--  
-- Script was generated by Devart dbForge Studio 2019 for MySQL, Version  
8.1.22.0  
-- Product home page: http://www.devart.com/dbforge/mysql/studio  
-- Script date 09.08.2021 12:26:06  
-- Server version: 5.2.3-falcon-alpha-community-nt  
-- Client version: 4.1  
--  
--  
-- Disable foreign keys  
--  
/*!40014 SET  
@OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,  
FOREIGN_KEY_CHECKS=0 */;  
  
--  
-- Set SQL mode  
--  
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,  
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;  
  
--  
-- Set character set the client will use to send SQL statements to the server  
--
```

Продолжение Приложения А

```
SET NAMES 'utf8';

--
-- Set default database
--
USE testdb;

--
-- Drop table `questions`
--
DROP TABLE IF EXISTS questions;

--
-- Drop view `appointed`
--
DROP VIEW IF EXISTS appointed CASCADE;

--
-- Drop view `finished`
--
DROP VIEW IF EXISTS finished CASCADE;

--
-- Drop view `statistics`
--
DROP VIEW IF EXISTS statistics CASCADE;
```

Продолжение Приложения А

```
--  
-- Drop table `tests`  
--  
DROP TABLE IF EXISTS tests;  
  
--  
-- Drop view `groups`  
--  
DROP VIEW IF EXISTS groups CASCADE;  
  
--  
-- Drop table `students`  
--  
DROP TABLE IF EXISTS students;  
  
--  
-- Drop table `studtest`  
--  
DROP TABLE IF EXISTS studtest;  
  
--  
-- Set default database  
--  
USE testdb;  
  
--
```

Продолжение Приложения А

```
-- Create table `studtest`  
--  
CREATE TABLE studtest (  
  id INT(11) NOT NULL AUTO_INCREMENT,  
  login VARCHAR(255) DEFAULT NULL,  
  idTest INT(11) DEFAULT NULL,  
  finish TINYINT(1) DEFAULT NULL,  
  res INT(11) DEFAULT NULL,  
  PRIMARY KEY (id)  
)  
...  
--  
-- Create table `students`  
--  
CREATE TABLE students (  
  surname VARCHAR(255) DEFAULT NULL,  
  name VARCHAR(50) DEFAULT NULL,  
  ...  
)  
ENGINE = INNODB,  
AVG_ROW_LENGTH = 1489,  
CHARACTER SET cp1251,  
COLLATE cp1251_general_ci;  
--  
-- Create view `groups`
```

Продолжение Приложения А

```
--  
CREATE  
DEFINER = 'root'@'localhost'  
VIEW groups  
AS  
SELECT DISTINCT  
  `students`.`group` AS `group`  
FROM `students`;  
  
--  
-- Create table `tests`  
--  
...  
--  
-- Create view `statistics`  
--  
CREATE  
DEFINER = 'root'@'localhost'  
VIEW statistics  
AS  
SELECT  
  `students`.`group` AS `group`,  
  `students`.`surname` AS `surname`,  
  `students`.`name` AS `name`,  
  `tests`.`Title` AS `Title`,  
  `studtest`.`res` AS `res`,
```

Продолжение Приложения А

```
`studtest`.`finish` AS `finish`  
FROM (`tests`  
JOIN `students`  
JOIN `studtest`  
ON ((`students`.`login` = `studtest`.`login`)))  
ON ((`tests`.`Questions` = `studtest`.`id`)))  
WHERE (`studtest`.`finish` = 1);  
  
--  
-- Create view `finished`  
--  
CREATE  
DEFINER = 'root'@'localhost'  
VIEW finished  
AS  
SELECT  
`students`.`login` AS `login`,  
`tests`.`Title` AS `Title`,  
`studtest`.`finish` AS `finish`,  
`studtest`.`res` AS `res`  
FROM (`tests`  
JOIN `students`  
JOIN `studtest`  
ON ((`students`.`login` = `studtest`.`login`)))  
ON ((`tests`.`Questions` = `studtest`.`idTest`)))  
WHERE (`studtest`.`finish` = 1);
```


Продолжение Приложения А

```
--  
-- Create view `appointed`  
--  
CREATE  
DEFINER = 'root'@'localhost'  
VIEW appointed  
AS  
SELECT  
  `students`.`surname` AS `surname`,  
  `students`.`name` AS `name`,  
  `students`.`patronymic` AS `patronymic`,  
  `tests`.`Title` AS `Title`,  
  `students`.`login` AS `login`  
FROM (`tests`  
JOIN (`students`  
JOIN `studtest`  
  ON ((`students`.`login` = `studtest`.`login`)))  
  ON ((`tests`.`Questions` = `studtest`.`idTest`)));  
  
--  
-- Create table `questions`  
--  
CREATE TABLE questions (  
  Test INT(11) NOT NULL,  
  Question INT(11) NOT NULL,  
  Text VARCHAR(255) DEFAULT NULL,
```

Продолжение Приложения А

```
V1 VARCHAR(255) DEFAULT NULL,  
V2 VARCHAR(255) DEFAULT NULL,  
...  
  
ENGINE = INNODB,  
AVG_ROW_LENGTH = 2730,  
CHARACTER SET cp1251,  
COLLATE cp1251_general_ci;  
  
--  
-- Dumping data for table tests  
--  
INSERT INTO tests VALUES  
(  
  ('Криптографические методы защиты информации', 1),  
  ('Основы программирования', 2);  
  
--  
-- Dumping data for table studtest  
--  
INSERT INTO studtest VALUES  
(1, 'aa1138', 2, 1, 85),  
(2, 'dov1139', 1, 1, 86),  
(3, 'na1140', 3, 1, 90),  
(4, 'dov1139', 2, 1, 0),  
(5, 'dov1139', 3, 0, NULL),  
(6, 'na1140', 1, 1, 0),
```

Продолжение Приложения А

(7, 'svp2057', -1, 0, 0),
(11, 'na1140', -1, 0, 0),
(12, 'na1140', -1, 0, 0),
(13, 'na1140', -1, 0, 0),
(14, 'na1140', -1, 0, 0),
(15, 'na1140', -1, 0, 0),
(16, 'na1140', -1, 0, 0),
(17, 'na1140', -1, 0, 0),
(18, 'na1140', -1, 0, 0),
(19, 'pps1247', -1, 0, 0),
(20, 'na1140', -1, 0, 0),
(21, 'na1140', -1, 0, 0),
(22, 'na1140', -1, 0, 0),
(23, 'na1140', -1, 0, 0),
(24, 'na1140', -1, 0, 0);

--

-- Dumping data for table students

--

INSERT INTO students VALUES

('Ашотян', 'Арина', 'Анатольевна', '1-16-5', 'aa1138', '113810'),
('Дмитриенко', 'Константин', 'Владимирович', '1-16-5', 'dkv1341', '134123'),
('Давыдова', 'Ольга', 'Владимировна', '1-16-5', 'dov1139', '113930'),
('Иванов', 'Иван', 'Иванович', '1-16-5', 'iii1247', '124730'),
('Михайлов', 'Николай', 'Ивнович', '1-16-5', 'mhi1249', '124922'),
('Николаев', 'Андрей', 'Дмитриевич', '1-16-5', 'na1140', '114005'),

Продолжение Приложения А

```
('Петров', 'Павел', 'Сергеевич', '1-16-5', 'pps1247', '124758'),
('Сидоров', 'Дмитрий', 'Геннадьевич', '1-16-5', 'sdg1248', '124848'),
('Смирнов', 'Сергей', 'Владимирович', '1-16-5', 'ssv1248', '124819'),
('Степанов', 'Валерий', 'Николаевич', '1-16-5', 'svn1343', '134316'),
('Скворцов', 'Владимир', 'Петрович', '1-16-5', 'svp2057', '205710');

--
-- Dumping data for table questions
--

--
-- Restore previous SQL mode
--

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;

--
-- Enable foreign keys
--

/*!40014                                     SET
FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
```