Образовательное учреждение

высшего образования

«Южно-Уральский институт управления и экономики»

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедры

И.Ю. Коробейникова

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Внутрихозяйственное землеустройство сельскохозяйственной организации

НАУЧНАЯ РАБОТА

ОУ ВО «ЮУИУиЭ» – ВК. 09.03.02 35 ПЗ

Руководитель работы

Т.Н. Лебедева

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Автор работы

О.В. Просяник

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Нормоконтроль

И.И. Новикова

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Челябинск

2016

ОУ ВО «Южно-Уральский институт управления экономики»

Кафедра Информационных, математических и естественнонаучных дисциплин

Утверждаю:

Заведующий кафедры

«Информационных, математических

и естественнонаучных дисциплин»

Коробейникова И.Ю.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«16» декабря 2015 г.

ЗАДАНИЕ

На НАУЧНУЮ работу

Направление 09.03.02 «Информационные системы и технологии»

На тему: Разработка программы тестирования обучающихся для ОУ ВО «ЮУИУиЭ»

Обучающийся группы: И-404

Фамилия, имя, отчество: Просяник Оксана Вадимовна

Руководитель: Лебедева Татьяна Николаевна

Содержание научной работы:

[Введение](#_Toc452577185)

[Глава 1](#_Toc452577186) Анализ предметной области

[1.1](#_Toc452577187) Анализ состояния исследуемого вопроса в конкретной области науки или технологии

[1.2](#_Toc452577188) Формирование требований к системе

1.3 Условия эксплуатации

1.4 Анализ известных решений и программных средств

1.5 Выбор средств проектирования и средств создания АИС

Вывод по главе 1

[Глава 2](#_Toc452577189) Проектирование информационной системы

[2.1](#_Toc452577190) Описание деятельности предприятия

[2.2](#_Toc452577191) Техническое задание на разработку

Вывод по главе 2

[Глава 3 Ра](#_Toc452577193)зработка и внедрение информационной системы

[3.1](#_Toc452577194) Определения вида серверной базы данных

[3.2](#_Toc452577195) Проектирование логической и физической модели БД

3.3 Проведение компьютерного тестирования

3.4 Работа системы с различными типами данных

3.5 Написание встроенного текстового редактора

3.6 Проектирование базы данных

3.7 Оценка экономических затрат на проект

Вывод по главе 3

[Заключение](#_Toc452577198)

[Список литературы](#_Toc452577199)

Приложение А

Приложение Б

Приложение В

Приложение Г

Приложение Д

Приложение Е

Приложение Ж

Приложение З

СОДЕРЖАНИЕ

[Введение 11](#_Toc453618906)

[Глава 1 Анализ предметной области 13](#_Toc453618907)

[1.1 Анализ состояния исследуемого вопроса в конкретной области науки или технологии 13](#_Toc453618908)

[1.2 Формирование требований к системе 13](#_Toc453618909)

[1.3 Условия эксплуатации 14](#_Toc453618910)

[1.4 Анализ известных решений и программных средств 19](#_Toc453618911)

[1.5 Выбор средств проектирования и средств создания АИС 22](#_Toc453618912)

[Вывод по главе 1 32](#_Toc453618918)

[Глава 2 Проектирование информационной системы 33](#_Toc453618919)

[2.1 Описание деятельности предприятия 33](#_Toc453618920)

[2.2 Техническое задание на разработку 42](#_Toc453618926)

[Вывод по главе 2 47](#_Toc453618934)

[Глава 3 Разработка и внедрение информационной системы 48](#_Toc453618935)

[3.1 Определения вида серверной базы данных 50](#_Toc453618937)

[3.2 Проектирование логической и физической модели БД 51](#_Toc453618938)

[3.3 Проведение компьютерного тестирования 51](#_Toc453618939)

[3.4 Работа системы с различными типами заданий 51](#_Toc453618940)

[3.5 Написание встроенного текстового редактора 53](#_Toc453618941)

[3.6 Проектирование базы данных 53](#_Toc453618942)

[3.7 Оценка экономических затрат на проект 58](#_Toc453618948)

[Вывод по главе 3 66](#_Toc453618949)

[Заключение 67](#_Toc453618950)

[Список литературы 68](#_Toc453618951)

[Приложение А 72](#_Toc453618952)

[Приложение Б 73](#_Toc453618953)

[Приложение В 74](#_Toc453618954)

[Приложение Г 75](#_Toc453618955)

[Приложение Д 76](#_Toc453618956)

[Приложение Е 77](#_Toc453618957)

[Приложение Ж 78](#_Toc453618958)

[Приложение З 79](#_Toc453618959)

# ВВЕДЕНИЕ

Одной из задач повседневного учительского труда является необходимость осуществлять контроль знаний учащихся. Формы контроля, применяемые учителями, очень разнообразны, но наиболее часто используются письменный или устный опросы. К сожалению, эти формы не лишены недостатков. При проведении устного опроса - это относительно большая затрата времени урока при небольшом количестве выставляемых оценок, при проведении письменных работ количество оценок возрастает, но много времени уходит на проверку.

Тестирование как эффективный способ проверки знаний находит все большее применение. Одним из основных и несомненных его достоинств является минимум временных затрат на получение надежных итогов контроля. При тестировании используют как бумажные, так и электронные варианты. Последние особенно привлекательны, так как позволяют получить результаты практически сразу по завершении теста.

Тестирование в педагогике выполняет три основные взаимосвязанные функции: диагностическую, обучающую и воспитательную:

Диагностическая функция заключается в выявлении уровня знаний, умений, навыков учащегося. Это основная, и самая очевидная функция тестирования. По объективности, широте и скорости диагностирования, тестирование превосходит все остальные формы педагогического контроля;

Обучающая функция тестирования состоит в мотивировании учащегося к активизации работы по усвоению учебного материала. Для усиления обучающей функции тестирования, могут быть использованы дополнительные меры стимулирования студентов, такие, как раздача преподавателем примерного перечня вопросов для самостоятельной подготовки, наличие в самом тесте наводящих вопросов и подсказок, совместный разбор результатов теста;

Воспитательная функция проявляется в периодичности и неизбежности тестового контроля. Это дисциплинирует, организует и направляет деятельность учащихся, помогает выявить и устранить пробелы в знаниях, формирует стремление развить свои способности.

Тестирование - более справедливый метод, оно ставит всех учащихся в равные условия, как в процессе контроля, так и в процессе оценки, практически, исключая субъективизм преподавателя.

Следует отметить, что именно тестирование постепенно становится основной формой сдачи экзаменов. С 2009 года для всех выпускников школ основной формой итоговой государственной аттестации в школе Российской Федерации является Единый Государственный Экзамен. И реалии таковы, что требуется обязательно вводить тестовые технологии в систему обучения.

Целью данной работы является разработка системы автоматизированного контроля знаний, возможности которой должны включать:

* проведение компьютерного тестирования;
* сбор и анализ результатов;
* использование во время тестирования различных вариантов тестовых заданий;
* оценивание результатов тестирования;
* расширенные возможности форматирования текста вопросов и вариантов ответа;
* использовать не только текстового варианта в качестве вопроса, но и возможность задания вопроса графическим способом (в виде прикрепленного изображения к вопросу) и комбинирование этих методов;
* использование базы данных для централизованного хранения информации (тестовых заданий, списки групп и пользователей системы, результаты тестирования).

# ГЛАВА 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## Анализ состояния исследуемого вопроса в конкретной области науки или технологии

Тестирование как эффективный способ проверки знаний находит все большее применение. Одним из основных и несомненных его достоинств является минимум временных затрат на получение надежных итогов контроля. При тестировании используют как бумажные, так и электронные варианты. Последние особенно привлекательны, так как позволяют получить результаты практически сразу по завершении теста.

Использование тестов в учебном процессе позволяет существенно увеличить интенсивность деятельности преподавателей и студентов, повысить мотивацию студентов института к изучению дисциплин, активизировать познавательную сферу студентов, развить способность к рефлективному восприятию и осмыслению материала, а при необходимости заменить традиционный экзамен тестовой формой его проведения

## 1.2 Формирование требований к системе

В соответствии с требованиями к системам (программам) тестирования знаний и умений по учебным дисциплинам, программы тестирования знаний и умений по учебным дисциплинам включают как промежуточные тесты и тесты для самоконтроля, так и итоговый (экзаменационный) тест. Эти программы должны выполнять следующие функции:

В отношении обучающихся:

* учитывать уровень подготовки обучаемого;
* не требовать специальных знаний и усилий для ввода ответа, свести к минимуму рутинные операции по вводу ответа;
* анализировать правильность ответа простых тестовых заданий, информировать о допущенных ошибках, содержать информацию, достаточную для их устранения;
* анализировать логику и ход решения сложных учебных задач, оказывать содействие при решении учебных задач, обеспечивая педагогически обоснованную помощь, с учетом характера затруднения и модели обучаемого, достаточную для того, чтобы решить задачу и усвоить способ ее решения;
* выдавать результат прохождения теста с комментариями о допущенных ошибках и рекомендациями для повторения теоретического материала.

В отношении преподавателя:

* не требовать специальных знаний и усилий для работы с программой, свести к минимуму рутинные операции по составлению теста;
* неограниченное количество заданий;
* множество вариантов ответа (до 10);
* богатые возможности форматирования текста;
* возможность прикреплять изображения.

## 1.3 Условия эксплуатации

Климатические условия эксплуатации, при которых должны обеспечиваться функциональные характеристики программы, должны удовлетворять требованиям, предъявляемым к техническим средствам разработчиков [систем управления базами данных](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)  (далее СУБД) и пользовательским компьютерам в части условий их эксплуатации.

Требования к видам обслуживания. Клиент и сервер не нуждаются в постоянном обновлении или обслуживании.

Требования к численности и квалификации персонала. Весь персонал, с учетом различий в уровне требований к его квалификации, условно может быть разделен на перечисленные ниже категории:

* конечные пользователи – наиболее массовая категория пользователей, являющаяся потребителями предоставляемых программой функциональных возможностей;
* администраторы СУБД – персонал, в обязанности которого входит выполнение функций администрирования СУБД серверов;
* администраторы клиентской стороны программы, в обязанности которых входят заведение новых пользователей в систему, сброс паролей, и удаление пользователей из системы;
* эксплуатационный персонал – специалисты, обеспечивающие функционирование технических средств сервера.

Требования к квалификации персонала, обслуживающего сервер, устанавливаются с учетом:

* принадлежности к категориям персонала;
* минимального набора квалификационных требований, без удовлетворения которых невозможно обеспечить работу Сервера во всех режимах функционирования.

Для использования системы от пользователя требуется наличие минимальных навыков работы с персональным компьютером.

К администраторам, помимо требований, предъявляемых к пользователям и информационным редакторам, должны предъявляться требования знаний состава и структуры баз данных, применяемых сервером и способность решать задачи администрирования, в том числе:

* управлять реестром учетных записей пользователей и групп пользователей;
  + - управлять правами доступа к данным и функциональными полномочиями пользователей;
    - изменять настройки СУБД;
* проводить инсталляции программного обеспечения системы;
* знание методов и приемов работы с подсистемами системы, в том числе знание настроек сервера, согласно руководство пользователя;
  + - владение навыками администрирования локальных вычислительных сетей (далее ЛВС) и системы управления базами данных.

В том числе способность самостоятельно выполнять:

* управление учетными записями пользователей операционной системы;
* настройку сетевых протоколов передачи данных, используемых в ЛВС;
* диагностические процедуры по определению целостности базы данных (далее БД).

Эксплуатационный персонал должен обладать навыками и умениями:

* удовлетворять требованиям, предъявляемым к конечным пользователям;
* знать назначение и владеть методами работы с программой, применение которых необходимо для обеспечения их собственной деятельности.

Требования к составу и параметрам технических средств. Состав и параметры технических средств серверов должны соответствовать минимальным требованиям, указанным разработчиками СУБД.

Требования к оборудованию сервера:

* монитор для работы графических средств СУБД требуется разрешение VGA или выше. Минимальное разрешение: 1024x768 точек;
* устройства манипулирования: требуется мышь microsoft mouse или совместимое указывающее устройство. А также клавиатура ps/2 или совместимая;
* дисковод для компакт-дисков или DVD-дисков: для установки с компакт-дисков или DVD-дисков требуется соответствующий дисковод;
* объем оперативной памяти: минимальный объем оперативной памяти для запуска сервера программы без учета базы данных: 4Гб;
* сетевое оборудование (сетевая карта).

В состав технических средств пользователя программы, должен входить пользовательский компьютер, обеспечивающий функционирование программного обеспечения в штатном режиме.

Требования к исходным кодам и языкам программирования.

Исходные коды программы должны быть реализованы на языке (C#). Взаимодействие с СУБД и создание базы данных реализуется на языке SQL.

Требования к программным средствам, используемым программой. Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы Windows 2003 Server (не ниже).

Специальные требования. Программа должна обеспечивать взаимодействие с пользователем (оператором) посредством графического пользовательского интерфейса, разработанного согласно рекомендациям компании-производителя операционной системы. Программа должна обеспечивать быстрый просмотр необходимой информации посредством отчетов.

1.3.1 Предварительный состав программной документации

Предварительный состав программной документации должен включать в себя:

* техническое задание;
* руководство пользователя.

1.3.2 Технико-экономические показатели

Требования к технико-экономическим показателям не предъявляются.

Экономические преимущества разработки описываются в главе 3 настоящей работы.

Стадии разработки. Согласно п. 2 ГОСТ 19.102-77 [1] и примечаний к нему разработка должна быть проведена в стадии:

* «техническое задание»;
* «техно-рабочий проект»;
* «тестирование»;
* «внедрение»;
* «эксплуатация».

Этапы разработки. На стадии «Техническое задание» должен быть выполнен этап «Разработка и утверждение технического задания».

На стадии «Техно-рабочий проект» должны быть выполнены перечисленные ниже этапы работ:

* «разработка программы»;
* «разработка программной документации»;
* «испытания программы».

На стадии «внедрение» должен быть выполнен этап «Подготовка и передача программы».

1.3.3 Содержание работ по этапам

На этапе «Разработка и утверждение технического задания» должны быть выполнены перечисленные ниже работы:

* определение требований к программе;
* определение стадий, этапов и сроков разработки программы и документации на нее;
* согласование и утверждение технического задания.

На этапе «Разработка программы» должна быть выполнена работа по программированию (кодированию) и отладке программы.

Последовательность реализации компонентов программы:

* создается и заполняется база данных;
* создается графический интерфейс клиента;
* создается движок управления клиента;
* создается менеджер ресурсов клиента.

На этапе «Разработка программной документации» должна быть разработана программная документация.

На этапе «Испытания программы» должны быть выполнены перечисленные ниже виды работ:

* проведение приемо-сдаточных испытаний;
* корректировка программы и документации по результатам испытаний.

Инсталляция программы сводится к:

* копированию файлов на клиентские компьютеры;
* переносу исходной базы данных в СУБД.

## 1.4 Анализ известных решений и программных средств

OpenTEST. Данный программный продукт имеет довольно узкую направленность, т. е. используется только для контроля знаний по различным дисциплинам в пределах учебного заведения. Поэтому, явных конкурентов у него нет.

Но существуют и разрабатываются программы, которые не являются аналогами в прямом смысле этого слова, но имеют аналогичную направленность, т. е. осуществляют проверку знаний. Одна из таких систем OpenTEST 2.0.

Компьютерная система тестирования знаний в дистанционном обучении OpenTEST 2.02.0 - это компьютерная система тестирования знаний созданная для очного итогового контроля качества усвоения теоретического материала, приобретенных знаний и практических навыков обучаемых в крупных организациях масштаба предприятия со сложной распределённой структурой.

Данная система дистанционного тестирования знаний обучаемых построена на основе Web-технологий и может работать как в корпоративной сети (INTRANET), так и в глобальной сети INTERNET.

Использование и администрирование системы OPENTEST полностью реализовано с помощью Web-интерфейса, что дает ряд существенных преимуществ:

* использование гипертекста (обеспечивает связность разнородной информации);
* браузер или клиентская программа WWW-сервера (дает единый, достаточно простой, быстрый и привычный интерфейс пользователя); интерфейс (позволяет осуществлять удаленное администрирование).

Основной особенностью системы OpenTEST 2.0 является её направленность на обеспечение тестирований учащихся с максимально строгой отчётностью.

При создании OPENTEST 2.0 были использованы такие программные продукты, как:

* язык написания сценариев, внедренный в HTML;
* небольшая и быстрая реляционная СУБД. Ее преимуществами являются многопоточность, поддержка нескольких одновременных запросов, записи фиксированной и переменной длины;
* самый распространенный в мире Web-сервер.

Все эти продукты являются OpenSource проектами, поэтому они бесплатные и свободные для распространения. Данные средства мультиплатформенные, т.е. они будут одинаково успешно работать под операционными системами Windows, UNIX и другими.

Структура системы тестирования является модульной, что позволяет подключать модули с дополнительными функциями, изменять режимы работы отдельных модулей, управляя системой настроек и не затрагивая при этом остальной ее части.

В системе OpenTEST 2.0 уделено много внимания проблемам безопасности при проведении тестирований. Так как при контрольных тестированиях конечная оценка играет огромную роль, объективность её выставления должна быть максимальная, а также все возможные варианты фальсификации результатов должны быть исключены. Для этого в системе OpenTEST 2.0 разработаны уникальные программные методы обеспечения безопасности при компьютерном тестировании:

* защита от взлома паролей ко всем модулям системы методом перебора (временная блокировка аккаунта, блокировка IP адреса);
* уникальный алгоритм динамического изменения хеш-идентификатора клиента, не позволяющий одновременную работу под одним аккаунтом с нескольких компьютеров и предотвращающий какие-либо несанкционированные попытки использования данных на промежуточных звеньях сети;
* использование защищённого протокола передачи данных SSL;
* ведение подробного структурированного лога событий;
* интерфейс для проведения гибкого анализа лога событий по указанным критериям;
* ежедневный автоматический анализ логов с формированием отчёта по безопасности;
* шифрование паролей на стороне клиента средствами JavaScript не позволяют перехватить чей-либо пароль при работе с системой OpenTEST 2.0 даже при отказе от использования протокола SSL.

Требования к оборудованию сервера:

* Pentium III 1Гц, 64Мб ОЗУ;
* сетевая карта или модем;
* отправка и получение данных.

Требования к программному обеспечению сервера:

* MS Windows 95 - MS Windows 2003 Server, Linux, MacOS, UNIX;
* любой браузер;
* сервер Apache;
* база данных MySQL.

Достоинства:

* функциональный редактор тестов;
* кроссплатформенность;
* дистанционное тестирование;
* низкие требования к аппаратным и программным ресурсам;
* расширения функциональности без необходимости внесения изменений в существующие модули.

Недостатки:

* крайне неудобный в работе интерфейс;
* избыточный функционал.

## 1.5 Выбор средств проектирования и средств создания АИС

## 1.5.1 Обоснование выбора СУБД

## Система управления базами данных (СУБД) – это система, которая обеспечивает поиск, хранение, корректировку данных, формирование ответов на запросы. Система обеспечивает сохранность данных, их конфиденциальность, перемещение и связь с другими программными средствами. Также с помощью СУБД можно создавать базы данных, наполнять их и работать с ними.

Реляционные СУБД – это системы управления реляционными базами данных. Такие модели управления характеризуются простотой, удобным табличным представлением, а также возможностями использования формального аппарата алгебры отношений и реляционного исчисления для осуществления обработки данных.

Основные функции СУБД:

* непосредственное управление данными во внешней памяти;
* управление буферами оперативной памяти;
* управление транзакциями;
* журнализация;
* поддержка языков БД.

Обычно СУБД работает с несколькими видами объектов:

* таблицы – основные объекты базы данных. Реляционная база данных может иметь много взаимосвязанных таблиц;
* запросы – это специальные структуры, предназначенные для обработки данных базы. С помощью запросов данные упорядочивают, фильтруют, отбирают, изменяют, объединяют, то есть обрабатывают. Существует много различных видов запросов, и самые простые из них и, к тому же наиболее часто используемые – это запросы на выборку;
* формы – это объекты, с помощью которых в базу вводят новые данные или просматривают имеющиеся;
* отчеты – это объекты, с помощью которых данные выдают на принтер в удобном и наглядном виде;
* макросы – это макрокоманды. Если какие-то операции с базой производятся особенно часто, то имеет смысл сгруппировать несколько команд в один макрос и назначить его выделенной комбинации клавиш;
* модули – это программные процедуры, написанные на каком-либо языке программирования.

## 1.5.2 MS SQL Server 2008

Microsoft SQL Server – система управления реляционными базами данных (СУРБД), разработанная корпорацией Microsoft. Основной используемый язык запросов – Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия.

Microsoft SQL Server также поддерживает OpenDatabaseConnectivity (ODBC) – интерфейс взаимодействия приложений с СУБД.

SQL Server включает несколько технологий управления данными и анализа данных:

* database engine – представляет собой основную службу для хранения, обработки и обеспечения безопасности данных. Этот компонент обеспечивает управляемый доступ к ресурсам и быструю обработку транзакций, что позволяет использовать его даже в самых требовательных корпоративных приложениях обработки данных;
* службы DataQualityServices – является решением для очистки данных на основе знаний. Службы DQS позволяют создать базу знаний, а затем выполнить в ней исправление данных и удаление дубликатов с помощью как автоматизированных, так и интерактивных средств;
* службы Analysis Services – это платформа аналитических данных и набор средств для бизнес-аналитики на личном уровне, уровне рабочей группы и организации. Серверный и клиентский конструкторы поддерживают стандартные решения OLAP, новые решения для создания табличных моделей, а также самостоятельную аналитику и совместную работу с помощью PowerPivot, Excel и среды SharePointServer;
* integration services – представляет собой платформу для создания высокопроизводительных решений по интеграции данных, в том числе пакетов для хранения данных, обеспечивающих извлечение, преобразование и загрузку данных;
* master data services – это решение SQL Server для управления основными данными. Позволяет обеспечить правильность информации, используемой для построения отчетов и выполнения анализа;
* репликация – представляет собой набор технологий копирования и распространения данных и объектов баз данных между базами данных, а также синхронизации баз данных для поддержания согласованности;
* службы ReportingServices – предлагают средства создания корпоративных отчетов с поддержкой веб-интерфейса, которые позволяют включать в отчеты, данные из различных источников, публиковать отчеты в разнообразных форматах, а также централизованно управлять безопасностью и подписками.

## 1.5.3 Обоснование выбора платформы для разработки

Для разработки данного программного продукта нужно выбрать такой язык программирования, который использует принципы объектно-ориентированного программирования.

В число современных сред разработки программного обеспечения, подходящих для решения поставленной задачи можно отнести Microsoft Visual Studio 2015, C++ Builder.Visual Studio 2015 - интегрированная среда разработки, включающая инструментальные средства для проектирования, кодирования, транслирования, отладки и выполнения программ[1]. Visual Studio 2015 позволяет быстро создавать и внедрять разнообразные приложения на базе ОС Windows, веб-приложения и приложения для мобильных устройств.

В Visual Studio предлагается целый ряд шаблонов приложений, полезных при создании программ, и несколько языков программирования, на которых можно написать эти программы: Visual Basic, Visual C#, Visual C++, JScript и т.д.[12] Studio поддерживает подключение библиотеки DevExpress с широким спектром возможностей.

В приложениях создаваемые с помощью Visual Studio можно внедрять самые разные технологии. Ниже приведено описание некоторых из них[4]:

.NET Framework, .NET Framework 3.5, .NET Framework 3.0, .NET Compact Framework - это интегрированный компонент Windows, который поддерживает создание и выполнение нового поколения приложений и веб-служб XML.Presentation Foundation (WPF) - WPF представляет собой набор типов .NET Framework, который можно использовать для создания внешнего вида клиентских приложений Windows. WPF состоит из таких компонентов, как расширяемый язык исправления для приложений XAML, элементы управления, привязка данных, двухмерная и трехмерная графика, анимация, стили, шаблоны, документы, мультимедийные данные, текст и типографические средства.- это независимая от обозревателя и платформы технология, позволяющая проектировать, разрабатывать и поставлять интерфейсы с поддержкой мультимедиа и многофункциональные приложения в Интернете.Forms - позволяет разрабатывать простые в развертывании и обновлении приложения с широкими графическими возможностями. Помимо этого, при доступе приложений Windows Forms к ресурсам на локальном компьютере обеспечивается более высокий уровень безопасности, чем при работе традиционных приложений Windows.

Язык XAML - это язык разметки для декларативной разработки приложений. Windows Presentation Foundation (WPF) реализует загрузчик XAML и обеспечивает поддержку языка XAML для типов WPF, поэтому большую часть пользовательского интерфейса приложения можно создавать с помощью разметки XAML..NET предоставляет платформу, которую можно использовать для создания веб-приложений. В ее состав входят такие службы, как управление состоянием, обработчики HTTP, модули HTTP и маршрутизация ASP.NET.++ Builder - программный продукт, инструмент быстрой разработки приложений (RAD), интегрированная среда программирования (IDE), система, используемая программистами для разработки программного обеспечения на языке C++ [14].++ Builder объединяет в себе комплекс объектных библиотек (STL, VCL, CLX, MFC и др.), компилятор, отладчик, редактор кода и многие другие компоненты. Цикл разработки аналогичен Delphi. Большинство компонентов, разработанных в Delphi, можно использовать и в C++ Builder без модификации.++ Builder содержит инструменты, которые при помощи drag-and-drop действительно делают разработку визуальной, упрощает программирование благодаря встроенному WYSIWYG - редактору интерфейса и пр.

В C++ Builder 2007 реализована полная поддержка API Microsoft Windows Vista, увеличена полнота соответствия стандарту ANSI C++, увеличена скорость компиляции и сборки, включена поддержка MSBuild, архитектур баз данных DBX4 и «VCL для Web», поддерживающий AJAX.

Для системы разработки расчетов уставок необходима удобная работа с возможностями API функций, удобное подключение библиотек C++, среди рассмотренных сред разработки больше всего подходит Visual Studio 2015.

## 

## 1.5.4 Язык программирования C#

В предыдущем пункте была выбрана среда разработки, теперь необходимо выбрать язык программирования. Ниже приведено описание языков, на которых можно писать программы в Visual Studio:Basic предоставляет простой и быстрый способ создания приложений Windows, веб-приложений и приложений для мобильных устройств на базе .NET Framework. Как и все программы, предназначенные для платформы .NET Framework, приложения, написанные в Visual Basic выгодно отличаются высоким уровнем безопасности и широкими возможностями взаимодействия между языками.

Язык Visual C# предназначен для разработки самых разнообразных приложений, выполняемых в среде .NET Framework. Язык Visual C# прост, строго типизирован и объектно-ориентирован. Благодаря множеству нововведений Visual C# обеспечивает возможность быстрой разработки приложений, но при этом сохраняет выразительность и элегантность, присущую языкам C.C++ мощный язык, предназначенный для обеспечения всестороннего и детального контроля над процессом построения как обычных приложений Windows (COM+), так и управляемых приложений Windows на базе .NET Framework.

C# относится к семье языков с [C](http://progopedia.ru/language/c/)-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет строгую статическую типизацию, поддерживает полиморфизм, перегрузку операторов, указатели на функции-члены классов, атрибуты, события, свойства, исключения, комментарии в формате XML. Переняв многое от своих предшественников — языков C++, Delphi, Modula и Smalltalk — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем: так, C# не поддерживает множественное наследование классов (в отличие от C++) или вывода типов (в отличие от Haskell).

Как объектно-ориентированный язык, C# поддерживает понятия инкапсуляции, наследования и полиморфизма. Все переменные и методы, включая метод Main – точку входа приложения – инкапсулируются в определения классов. Класс может наследовать непосредственно из одного родительного класса, но может реализовывать любое число интерфейсов. Для методов, которые переопределяют виртуальные методы в родительском классе, необходимо ключевое слово override, чтобы исключить случайное повторное определение. В языке C# структура похожа на облегченный класс: это тип, распределяемый в стеке, реализующий интерфейсы, но не поддерживающий наследование.

В дополнение к основным описанным объектно-ориентированным принципам, язык C# упрощает разработку компонентов программного обеспечения благодаря нескольким инновационным конструкциям языка, в число которых входят следующие:

* Инкапсулированные сигнатуры методов, называемые *делегатами*, которые поддерживают типобезопасные уведомления о событиях.
* Свойства, выступающие в роли методов доступа для закрытых переменных-членов.
* Атрибуты с декларативными метаданными о типах во время выполнения.
* Встроенные комментарии XML-документации.
* LINQ, предлагающий встроенные возможности запросов в различных источниках данных.

Если потребуется обеспечить взаимодействие с другим программным обеспечением Windows, таким как объекты COM или собственные библиотеки DLL Win32, в языке C# можно использовать процесс, который называется "Interop." Процесс Interop позволяет программам на C# выполнять практически любые действия, которые может выполнять исходное приложение на C++. Язык C# поддерживает даже указатели и понятие "небезопасного" кода для тех случаев, когда прямой доступ к памяти имеет крайне важное значение.

Процесс построения C# по сравнению с C и C++ прост и является более гибким, чем в Java. Нет отдельных файлов заголовка, а методы и типы не требуется объявлять в определенном порядке. В исходном файле C# может быть определено любое число классов, структур, интерфейсов и событий.

Язык С# имеет ряд преимуществ. Например, в C# определение класса состоит из одного файла (\*.cs), в отличие от C++, где определение класса разбито на заголовок (\*.h) и реализацию (\*.cpp). Синтаксические конструкции C# унаследованы не только от C++, но и от Visual Basic. Например, в C#, как и в Visual Basic, используются свойства классов. Как C++, C# позволяет производить перегрузку операторов для созданных типов. C# - это фактически гибрид разных языков. При этом C# синтаксически не менее чист, чем Java, так же прост, как Visual Basic, и обладает практически той же мощью и гибкостью, что и C++. Управление памятью в C# производится автоматически.

В целом язык C# унаследовал многое из хорошего, что есть в современных языках программирования. В рамках данной дипломной работы применение Visual C# предпочтительнее.

[Архитектура платформы .NET Framework](javascript:void(0)).

Программа на языке C# выполняется в среде .NET Framework – интегрированном компоненте Windows, содержащем виртуальную систему выполнения (среда CLR) и унифицированный набор библиотек классов. Среда CLR представляет собой коммерческую реализацию Майкрософт инфраструктуры CLI (common language infrastructure), международного стандарта, основы сред выполнения и разработки с тесным взаимодействием языков и библиотек.

Исходный код, написанный на языке C#, компилируется в промежуточный язык (IL) в соответствии со спецификацией CLI. Код IL и ресурсы, такие как растровые изображения и строки, хранятся на диске в исполняемом файле, называемом сборкой, с расширением EXE или DLL в большинстве случаев. Сборка содержит манифест со сведениями о типах сборки, версии, языке и региональных параметрах, и требованиях безопасности.

При выполнении программы на C# сборка загружается в среду CLR в зависимости от сведений в манифесте. Далее, если требования безопасности соблюдены, среда CLR выполняет JIT-компиляцию для преобразования кода IL в инструкции машинного кода. Среда CLR также предоставляет другие службы, относящиеся к автоматическому сбору мусора, обработке исключений и управлению ресурсами. Код, выполняемый средой CLR, иногда называют "управляемым кодом" в противопоставление "неуправляемому коду", который компилируется в машинный код, предназначенный для определенной системы. Далее показаны отношения во время компиляции и время выполнения между файлами с исходным кодом C#, библиотеками классов .NET Framework, сборками и средой CLR.

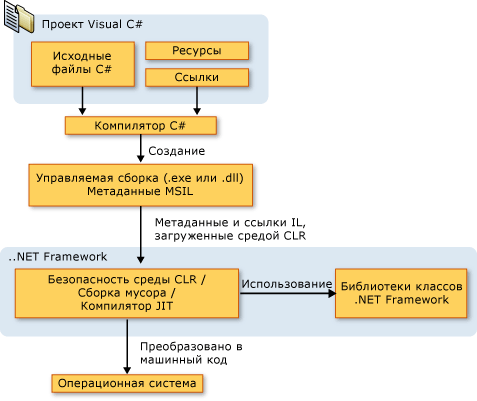


Рисунок 1 - Отношения во время компиляции и время выполнения между файлами с исходным кодом C#

Взаимодействие между языками является ключевой особенностью .NET Framework. Поскольку код на промежуточном языке (IL), создаваемый компилятором C#, соответствует спецификации CTS, код IL на основе C# может взаимодействовать с кодом, создаваемым версиями языков Visual Basic, Visual C++, Visual J# платформы .NET Framework и еще более чем 20 CTS-совместимых языков. В одной сборке может быть несколько модулей, написанных на разных языках платформы .NET Framework, и типы могут ссылаться друг на друга, как если бы они были написаны на одном языке.

Помимо служб времени выполнения, в .NET Framework также имеется обширная библиотека, состоящая из более чем 4000 классов, организованных по пространствам имен, которые обеспечивают разнообразные полезные функции для любых действий, начиная от ввода и вывода файлов для управлением строками для разбивки XML, и заканчивая элементами управления Windows Forms. В обычном приложении на языке C# библиотека классов .NET Framework интенсивно используется для "устройства" кода.

# ВЫВОД ПО ГЛАВЕ 1

Руководством компании было подготовлено техническое задание, которое легло в основу разработки автоматизированной информационной системы.

В качестве инструмента для разработки автоматизированной информационной системы был выбран язык программирования C#, т.к. он идеально подходит для создания программных продуктов любой сложности, а основа готовых элементов управления заложенных в сборку .Net Framework позволит сделать исполняющий файл более легким, разработку более быстрой и само приложение производительным в ОС Windows.

В качестве базы данных для разработки была выбрана MS SQL Server, т.к. на предприятии уже имеется данная платформа, и она отлично распараллеливает нагрузку запросов между всеми ядрами серверных процессоров. В язык программирования C# и библиотек Net. Framework заложены большое количество инструментов создания клиент-серверных запросов к базам банных, что снимает с разработчика какие-либо ограничения.

Для программной реализации задания была выбрана интегрированная среда разработки Visual Studio 2015.

# ГЛАВА 2 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

## 2.1 Описание деятельности предприятия

## 2.1.1 Краткая характеристика предприятия

Образовательное учреждение высшего образования «Южно-Уральский институт управления и экономики» (далее ОУ ВО «ЮУИУиЭ») образовано 22 апреля 1996года. Институт входит в состав 100 лучших институтов России, является одним из лидеров в Уральском регионе.

Ежегодно выпускает специалистов по следующим направлениям (специальностям):

* «Архитектура»;
* «Дизайн»;
* «Дизайн архитектурной среды»;
* «Информационные системы и технологии»;
* «Менеджмент»;
* «Реклама и связи с общественностью»;
* «Экономика».

А также совсем новым направлениям (специальностям):

* «Бизнес-информатика»;
* «Государственное и муниципальное управление»;
* «Землеустройство и кадастры»;
* «Строительство»;
* «Технология транспортных процессов»;
* «Торговое дело»;
* «Экономическая безопасность»;
* «Электроэнергетика и электротехника»;
* «Юриспруденция».

Более трех тысяч студентов получают в Институте фундаментальные знания по различным формам обучения: очной, заочной, очно-заочной. Подготовка ведется по 16 специальностям и направлениям.

Ежегодно институт выпускает большое количество методических пособий, преподаватели института работают над созданием учебников по дисциплинам учебного плана той или иной специальности.

Институт ежегодно проводит научные конференции, круглые столы по самым актуальным вопросам развития государства и экономики, мастер классы для дизайнеров.

В Институте также существует система дополнительного получения профессиональных знаний для школьников старшего и среднего школьного возраста, а также для студентов и специалистов, желающих получить дополнительные квалификации.

Южно-Уральский институт управления и экономики – это высококвалифицированные преподаватели.

Со дня основания института выпущено более двух тысяч специалистов. Многие выпускники занимают ключевые посты в государственных структурах регионального и муниципального уровней, стали крупными организаторами производств и финансово-экономических структур, а также частных предприятий.

## 2.1.2 Структурное взаимодействие в ОУ ВО ЮУИУиЭ

Рассмотрим структурное взаимодействие в ЮУИУиЭ, в соответствии с рисунком 1.

Краткая информация о структуре ОУ ВО «ЮУИУиЭ»:

* «Ученый совет» – постоянно действующий выборный представительный орган вуза, занимающийся решением стратегических вопросов его развития;
* «Собрание учредителей» – собрание, в которое входят юридические или физические лица, создавшие организацию;
* «Ректор» – руководитель высшего учебного заведения. Руководит всеми подразделениями вуза;
* «Проректор по научной работе» – подчиняется ректору и отвечает за организацию научно-исследовательскую работу в учреждении;
* «Проректор по внеучебной деятельности» – подчиняется ректору и отвечает за внеучебную работу в учреждении;
* «Проректор по административно-хозяйственной части» – подчиняется ректору и отвечает за адмиистративно-хозяйственную часть в учреждении;
* «Проректор по учебной работе» – подчиняется ректору и отвечает за учебную работу в учреждении.

Ректор и проректора должны взаимодействовать с существующими кафедрами и обеспечивать их всеми необходимыми средствами.

В соответствии с положением о кафедрах ОУ ВО «ЮУИУиЭ» основными видами деятельности кафедр института являются:

* учебно-методическая;
* научно-исследовательская;
* экспертно-методическая;
* воспитательная и внеучебная;
* профориентационная;
* подготовка и повышение квалификации научно-педагогических кадров кафедры.

Одна из учебно-методических задач кафедры является разработка технологий формирования в процессе обучения компетенций выпускников, их профессионально значимых качеств личности как специалистов в различных видах деятельности. Поэтому многие кафедры организуют и проводят предметные олимпиады (состязание учащихся учреждений среднего общего, высшего или профессионального образования, требующее от участников демонстрации знаний и навыков в области одной или нескольких изучаемых дисциплин) по тематике кафедры. При этом предметные олимпиады могут быть как внутри вузовские, так и межвузовские и всероссийские.

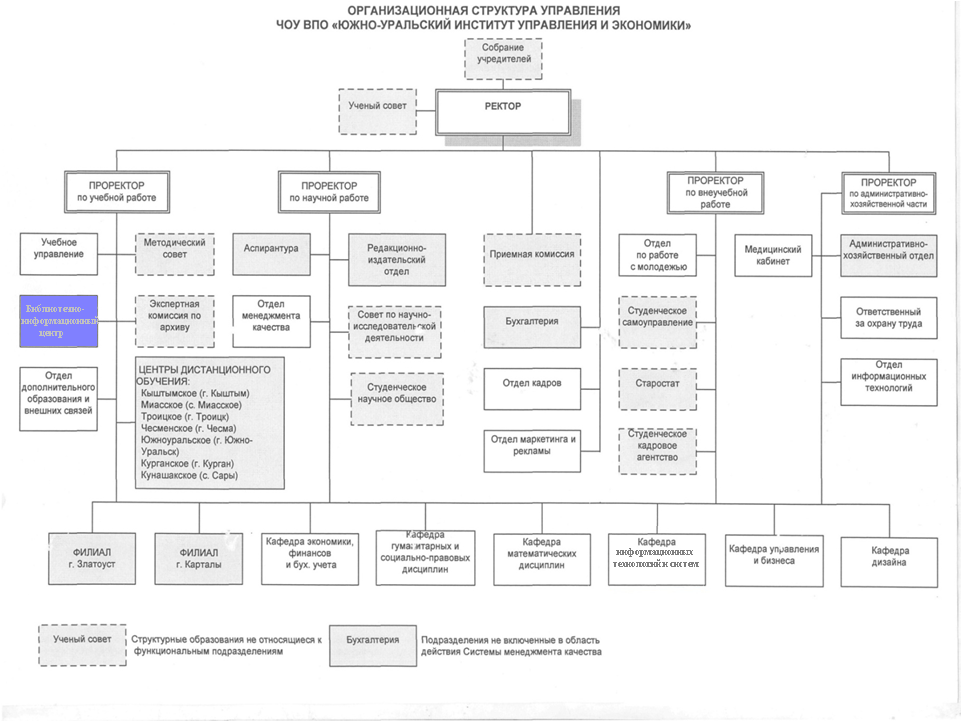


Рисунок 2 – Структура ОУ ВО «ЮУИУиЭ»

## 2.1.3 Структура локальной сети ОУ ВО «ЮУИУиЭ»

В ОУ ВО «ЮУИУиЭ» установлено около 100 АРМ (автоматизированных рабочих мест) принтеры, сканеры.

Персональная вычислительная техника по комплектации самая различная.

На предприятии все компьютеры соединены по локальной сети. Для локальной связи между компьютерами используются витая пара, коммутаторы типа «роутер», сетевые адаптеры.

Тип сети: сеть с выделенным сервером. Сеть построена на основе топологии «звезда».

Распределение ресурсов: совместный доступ к данным, общие принтеры, другие совместно используемые периферийные устройства, организуется путем предоставления локальных ресурсов и периферийных устройств в общее пользование, хотя не исключено и использование сетевых устройств, главным образом принтеров.

На рисунке 3 представлен отдельный фрагмент схемы локальной вычислительной сети.

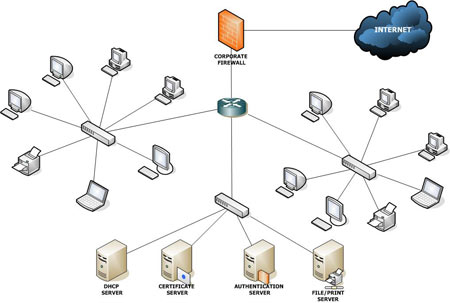


Рисунок 3 – Схема локальной сети ЮУИУиЭ

## 2.1.4 Используемое программное обеспечение

В таблице 1 приведены программные обеспечения, используемые в ОУ ВО «ЮУИУиЭ».

Таблица 1 – Основное программное обеспечение (далее ПО) используемое в ОУ ВО «ЮУИУиЭ»

|  |  |  |
| --- | --- | --- |
| Название программного продукта | | Назначение |
| Microsoft Developer Network Academic Alliance (MSDN AA) | Windows Server™ 2003 R2 Enterprise Edition | Операционные системы корпорации Microsoft |
| Windows Vista |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Windows XP Professional |  | | |
| Visual Studio™ 2005 | Средства разработки | | |
| SQL Server™ 2005 | Система анализа и управления реляционными базами данных | | |
| Microsoft Exchange Server 2007 | Платформа передачи сообщений, поддерживающая электронную почту, составление расписаний, интерактивные формы и средства создания пользовательских приложений совместной работы и обмена сообщениями. | | |
| Microsoft Office Access | Приложение для управления базами данных | | |
| Microsoft Visio Professional | Приложение для работы с бизнес-диаграммами и техническими диаграммами | | |
| Microsoft Project Professional | Решение задач управления проектами в масштабах предприятия | | |
|  | Microsoft Office OneNote | | Приложение для записи заметок и управления ими |
| Microsoft Office InfoPath | | Приложение сбора данных и управления ими |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
|  | Microsoft Office SharePoint Portal Server | | Организация централизованного доступа ко всей ключевой деловой информации и приложениям |
| Microsoft Office 2007 OpenLicensePack | [Microsoft Office Excel](http://soft.softline.ru/program_page_pricing.php?id=25880&show_price=1) | | Приложения для построения диаграмм и аналитические инструменты для работы с электронными таблицами и совместного доступа к информации |
| [Microsoft Office Word](http://soft.softline.ru/program_page_pricing.php?id=25853&show_price=1) | | Текстовый редактор для создания и редактирования документов. |
| [Microsoft Office Outlook](http://soft.softline.ru/program_page_pricing.php?id=25907&show_price=1) | | Решение, предназначенное для организации и управлением средствами связи (электронной почты и мгновенных сообщений) |
| [Microsoft Office PowerPoint](http://soft.softline.ru/program_page_pricing.php?id=25908&show_price=1) | | Приложение для создания презентации. |
|  | ABBYY Fine Reader | | Система оптического распознавания |
|  | | [Autodesk 3ds Max](http://soft.softline.ru/program_page_pricing.php?id=38293&show_price=1) | Приложение для визуального проектирования, разработки компьютерных и видеоигр, создания визуальных эффектов | |
|  | | Adobe In Design CS3 | Приложение для создания профессиональных макетов и дизайнерских решений. | |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
|  | CorelDraw Graphics Suite X3 | Графический редактор для создания иллюстраций с помощью мощных инструментов для работы с векторной графикой |
|  | Adobe Dreamweaver CS3 | Приложение для разработки Web-приложений |
| [Adobe Flash CS3 Professional](http://soft.softline.ru/program_page_pricing.php?id=41650&show_price=1) | Приложение для создания Web-ресурсов. |
| MathCAD | Интегрированная система решения математических, инженерно-технических и научных задач. |
| Nero 8 Site | Приложение для организации и управления мультимедиа содержимым, позволяющая создавать и записывать диски Blu-ray и HD DVD |
|  | WinRAR | Архиватор |
|  | [Kaspersky Open Space Security](http://soft.softline.ru/program_page_pricing.php?id=42186&show_price=1) | Решение для централизованной защиты рабочих станций в корпоративной сети и за ее пределами от всех видов современных интернет-угроз: вирусов, шпионских программ, хакерских атак и спама. |
| [Borland Developer Studio 2006 Professional](http://soft.softline.ru/program_page_pricing.php?id=32013&show_price=1) | Решение для разработки под Windows, объединяющее продукты Delphi, C++Builder и C#Builder. |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| 1С: Бухгалтерия 7.7 | 1с:Зарпалата и Кадры  1с:Предприятие | Приложение для анализа и учета бухгалтерской деятельности. |
| 1С: Бухгалтерия 8.1 | Комплект 1с для учебных заведений | Приложение для анализа и учета бухгалтерской деятельности. |
| Directum |  | Система электронного  документооборота и управления взаимодействием |
| Галактика ERP |  | Система управление ресурсами предприятия |
| Project Expert |  | Программа разработки бизнес плана и оценки инвестиционных проектов |

## 

## 2.1.5 Основные задачи и функции отдела

Основными задачами отдела являются:

* развитие и внедрение новых информационных технологий, системных программных средств и сопутствующего программного обеспечения;
* организация обмена информацией;
* обеспечение информационной безопасности;
* повышение профессионального уровня специалистов отрасли по вопросам практического применения новых информационных технологий и сопутствующего программного обеспечения.

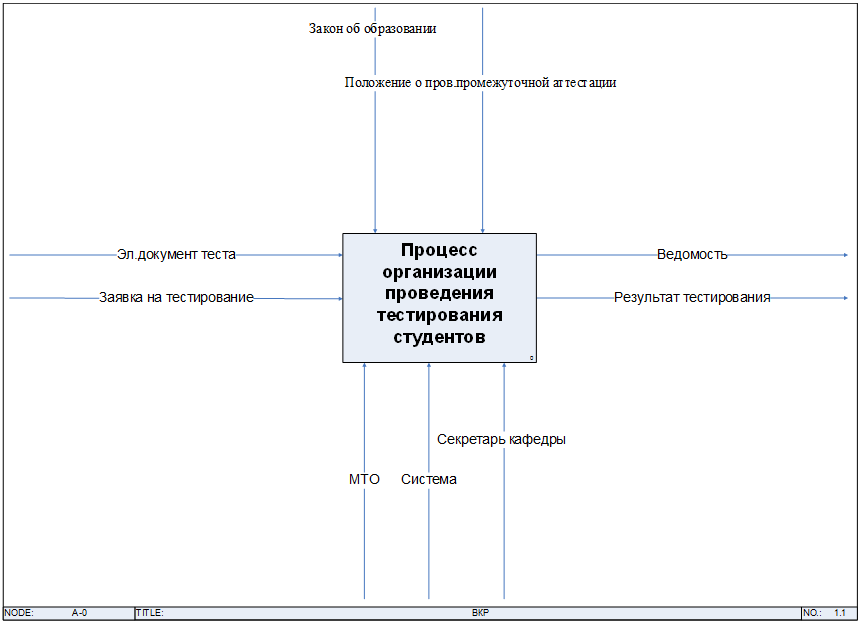
Основными функциями отдела являются:

* изучение инновационных, в том числе, новых информационных, технологий и последующее внедрение;
* согласование технических заданий на разработку и внедрения нового программного обеспечения в отрасли на предмет обеспечения сетевого и системного взаимодействия;
* настройка, администрирование, и модернизация программного обеспечения компьютерных;
* администрирование АТС и ведомственной телефонной сети;
* поддержка Internet-технологий в отрасли, обеспечение доступа к Internet-услугам;
* сопровождение системного, сетевого и сопутствующего программного обеспечения;
* обучение специалистов отрасли работе с системным ПО;
* разработка новых системных программ, модернизация используемых системных программных продуктов;
* консультации по вопросам системного программного обеспечения;
* регистрация и рассылка служебной корреспонденции в электронном виде;

создание и правка электронных документов с применением компьютерной техники и офисных программ.

2.1.6 Построение модели деятельности предприятия

IDEF0 — методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-‎процессов. Отличительной особенностью IDEF0 является её акцент на соподчинённость объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность.



40

Рисунок 4 - Процесс организации проведения тестирования студентов

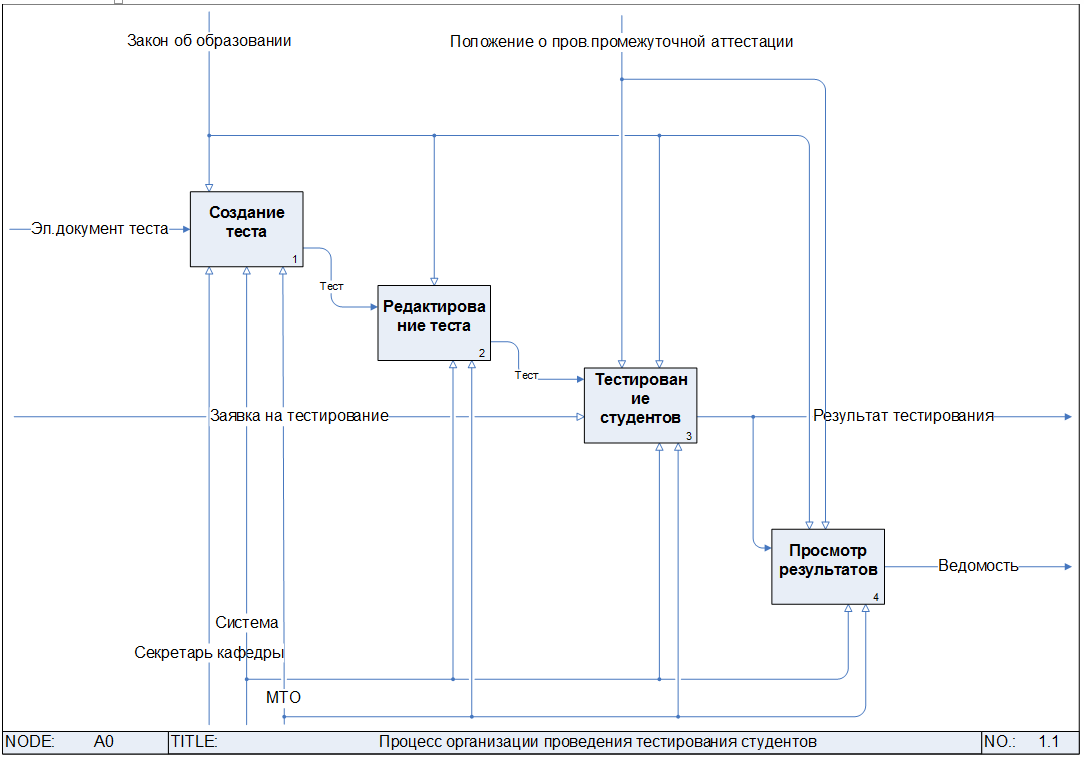


Рисунок 5 - Декомпозиция процесса организации проведения тестирования студентов

## 2.2 Техническое задание на разработку

Техническое задание (далее ТЗ) – исходный документ на проектирование технического объекта (изделия). ТЗ устанавливает основное назначение разрабатываемого объекта, его технические характеристики, показатели качества и технико-экономические требования, предписание по выполнению необходимых стадий создания документации (конструкторской, технологической, программной и т. д.) и её состав, а также специальные требования.

Техническое задание также используется при создании творческого объекта (видеоролик, статья, графическое изображение, сайт).

Техническое задание является юридическим документом – как приложение включается в договор между заказчиком и исполнителем на проведение проектных работ и является его основой: определяет порядок и условия работ, в том числе цель, задачи, принципы, ожидаемые результаты и сроки выполнения. Т.е. должны быть объективные критерии, по которым можно определить, сделан ли тот или иной пункт работ или нет.

Все изменения, дополнения и уточнения формулировок ТЗ обязательно согласуются с заказчиком и им утверждаются. Это необходимо и потому, что в случае обнаружения в процессе решения проектной задачи неточностей или ошибочности исходных данных возникает необходимость определения степени вины каждой из сторон-участниц разработки, распределения, понесенных в связи с этим убытков.

Наименование программы: «TestRed».

Тестовые задания могут составляться с использованием разнообразных компьютерных инструментов, начиная от различных редакторов и программ для разработки презентаций и до использования языков программирования и возможностей сети Интернет.

Но разработка качественного тестового инструментария - длительный, трудоемкий и дорогостоящий процесс.

С помощью разработанной системы тестирования возможна организация и проведение тестирования, экзаменов в любых образовательных учреждениях (вузы, колледжи, школы) как с целью выявить уровень знаний по любым учебным дисциплинам, так и с обучающими целями.

Разработанная компьютерная система тестирования знаний универсальна не только по возможностям применения, но и по своей структуре. Продукт состоит из двух модулей, которые работают с общей базой данных.

## Требования к функциональным характеристикам:

* отдельные функциональные модули;
* простота создания тестов;
* удобный интерфейс;
* различные режимы тестирования;
* неограниченное количество заданий;
* множество вариантов ответа (до 10);
* богатые возможности форматирования текста;
* возможность прикреплять изображения;
* возможность сделать скриншот произвольной области экрана и вставка полученного изображения в текст задания;
* возможность перемешивания заданий и вариантов.

## Требования к обеспечению надежного функционирования программы. Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением Заказчиком совокупности организационно-технических мероприятий, перечень которых приведен ниже:

* организацией бесперебойного питания технических средств;
* использованием лицензионного программного обеспечения;
* регулярным выполнением рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г. Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию персональных компьютеров и оргтехники и сопровождению программных средств;
* регулярным выполнением требований ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов;
* автоматизированная информационная система (далее АИС) должна обеспечивать устойчивость функционирования при наличии ошибок во входных данных.

## Время восстановления после отказа. Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать двадцати минут при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

## Отказы из-за некорректных действий пользователей системы. Отказы программы вследствие некорректных действий пользователя при взаимодействии с программой недопустимы.

## Условия эксплуатации

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

Минимальное количество персонала, требуемого для работы программы, должно составлять не менее трех штатных единиц – системный администратор, администратор базы данных и конечный пользователь программы – оператор. Системный администратор должен иметь высшее профильное образование и сертификаты компании-производителя операционной системы. В перечень задач, выполняемых системным администратором, должны входить:

* задача поддержания работоспособности технических средств;
* задачи установки (инсталляции) и поддержания работоспособности системных программных средств – операционной системы;
* задача установки (инсталляции) программы;
* задача создания резервных копий базы данных.

Администратор базы данных должен иметь высшее профильное образование и сертификаты компании-производителя операционной системы. В перечень задач, выполняемых администратором базы данных, должны входить:

* обеспечение целостности данных;
* недопущение ознакомления с структурой базы данных третьих лиц;
* в случаи нарушения целостности базы данных, в кротчайшие сроки восстановить целостность данных, за счет резервных копий, создаваемых администратором.

## Требования к составу и параметрам технических средств

В состав технических средств должен входить совместимый персональный компьютер (далее ПЭВМ), выполняющий роль сервера, включающий в себя:

* процессор Intel Atom-1.7Hz, не менее;
* оперативную память объемом, 2Гигабайт, не менее;
* HDD, 750 Гигабайт, не менее;
* два блока питания подключенных на разных фазах.

## Требования к информационным структурам и методам решения

Базы данных работают под управлением Microsoft SQL Server. Всего задействована 1база данных.

Такая структура баз данных обусловлена перспективой развития разрабатываемого программного обеспечения. В перспективе, программа будет доступна, выложена в публичный доступ и возможность быстро создать тест будет у любого преподавателя. В базе данных test\_server хранится информация не желательная для публичного распространения, например такая, как список заказчиков, поэтому данные были изначально разбиты на две базы данных. Одна будет распространяться с самой программой, а вторая будет находиться только на сервере нашей компании.

Исходный код должен быть структурированным и интуитивно понятным для чтения (в случаях реализации труднодоступного кода, оставить комментарии с разъяснением).

При выборе языка программирования необходимо уделить внимание наиболее гибкому и распространенному языку программирования.

Клиентскую часть программы необходимо использовать на семействе Windows начиная с версии XP SP3.

Данные о заказчиках должны быть вынесены в отдельную базу данных.

Программа должна обеспечивать одновременную работу пользователей посредством Windows Form интерфейса.

Состав программной документации должен включать в себя:

* техническое задание;
* программу и методики испытаний;
* руководство оператора.

# ВЫВОД ПО ГЛАВЕ 2

Сформулированы требования к рабочим местам, где будет выполняться запуск приложения, а также определены его функциональные возможности.

Сформулированы требования к серверу, на котором будут расположены базы данных, а также поля и свойства полей баз данных. В сервер заложены требования по надежности, включающие оборудования для поддержания бесперебойной работы питания и повышающие отказоустойчивость системы.

Были описаны область применения и назначения разработанного программного продукта.

Анализируя главу, можно сказать, что разработанный программный продукт должен соответствовать кроме базового функционала, такого как:

* интуитивный понятный интерфейс;
* обработка ошибок некорректного ввода;
* редактирования справочников.

# Глава 3 Разработка и внедрение информационной системы

Для решения поставленной задачи необходимо решить несколько основных вопросов:

* определить вид серверной базы данных;
* спроектировать логическую и физическую модель базы данных;
* проведение компьютерного тестирования;
* организовать работу системы с различными типами заданий;
* продумать методику оценивания результатов тестирования;
* написание встроенного текстового редактора для расширенного форматирования текста;
* реализация добавления изображения и функции «Скриншот выделенного участка экрана» с последующей вставкой его в задание теста.

При разработке системы были разработаны следующие диаграммы‎:

Диаграмма прецедентов (‎диаграмма вариантов использования‎) в UML — диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Основное назначение диаграммы — описание функциональности и поведения, позволяющее заказчику, конечному пользователю и разработчику совместно обсуждать проектируемую или существующую систему.

При моделировании системы с помощью диаграммы прецедентов мы стремились‎:

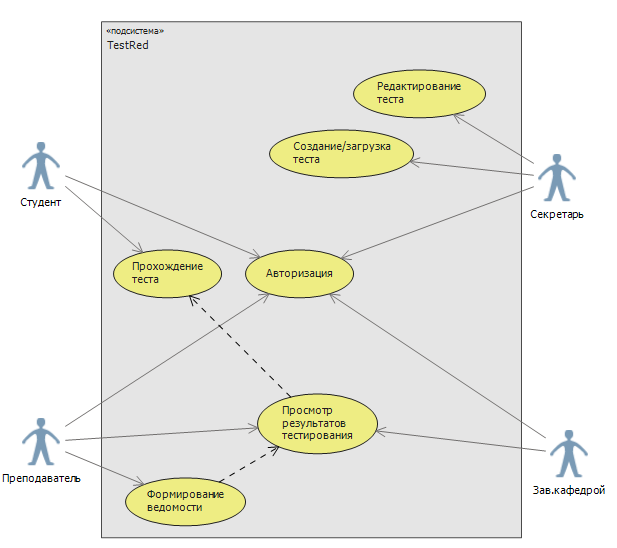
* чётко отделить систему от её окружения;
* определить действующих лиц (‎актёров‎), их взаимодействие с системой и ожидаемый функционал системы;
* определить в глоссарии предметной области понятия, относящиеся к детальному описанию функционала системы.

Элементы.

Рамки системы — прямоугольник с названием в верхней части и эллипсами (‎прецедентами‎) внутри. Часто может быть опущен без потери полезной информации.

Актёр — стилизованный человечек, обозначающий набор ролей пользователя (‎понимается в широком смысле‎: человек, внешняя сущность, класс, другая система‎), взаимодействующего с некоторой сущностью. Актёры не могут быть связаны друг с другом.

Прецедент — эллипс с надписью, обозначающий выполняемые системой действия, приводящие к наблюдаемым актёрами результатам. Надпись может быть именем или описанием (‎с точки зрения актёров‎) того, «что» делает система (‎а не «как»‎). Имя прецедента связано с непрерываемым сценарием — конкретной последовательностью действий, иллюстрирующей поведение. В ходе сценария актёры обмениваются с системой сообщениями. Сценарий может быть приведён на диаграмме прецедентов в виде UML-‎комментария. С одним прецедентом может быть связано несколько различных сценариев.



## Рисунок 6 - Диаграмма прецедентов

## 3.1 Определения вида серверной базы данных

Система управления базами данных представляет собой программный комплекс, позволяющий обеспечить пользователей (прикладные программы) языковыми средствами описания и манипулирования данными, обеспечить поддержку логических моделей данных, обеспечить операции создания и манипулирования данными, защиту и целостность данных. При выборе БД требуется учесть множество факторов, среди которых нужды и цели, под которые она будет использоваться и тип распространения.

## 3.2 Проектирование логической и физической модели БД

Требуется создать схему базы данных с набором отношений, указанием первичных ключей, а также «связей» между отношениями, представляющих собой внешние ключи.

## 3.3 Проведение компьютерного тестирования

Поскольку все тестовые задания хранятся на сервере, то тестирование знаний обучаемых в автоматизированной системе контроля знаний должно осуществляться путем организации сеанса тестирования для каждого пользователя (тестируемого). Для каждого пользователя осуществляется индивидуальная регистрация в системе с использованием пароля. Сеанс характеризуется количеством вопросов, выдаваемых пользователю в случайном порядке. Во время сеанса пользователь в окне клиентского приложения в случайном порядке должен получать набор вопросов с вариантами ответов. Кроме собственно выбора ответов, он может также пропускать вопросы, перемещаться между ними и отвечать на вопросы в произвольном порядке.

## 3.4 Работа системы с различными типами заданий

Программа тестирования должна оперировать с семью типами заданий. Пять из которых закрытого типа (одиночный выбор, множественный выбор, установление порядка следования, установление соответствия, указание истинности или ложности утверждений) и два открытого(ручной ввод числа (чисел), ручной ввод текста). В тесте требуется реализовать возможность использования любое количество любых типов заданий - можно только один, можно и все сразу. Т.к. взаимодействие тестируемого с системой происходит через клиентское приложение, то это накладывает определенные особенности на форму представления вопросов и вариантов ответов.

Типичная структура представления вопроса имеет следующий вид:

1. Описание ситуации. В этой части вопроса могут присутствовать текст, рисунки и графики. Эта часть вопроса не является обязательной, но все же рекомендуется ее применять, так как одна и та же ситуация может быть основой для множества однотипных вопросов.
2. Текст вопроса. Он должен быть сформулирован четко, и не содержать двузначных формулировок. Текст вопроса достаточно часто может быть совмещен с описанием ситуации.
3. Выбор (ввод) ответов. В клиентском приложении выборочные варианты ответов на вопросы реализуются через RadioButton («одиночный выбор»), CheckBox («множественный выбор»), ComboBox (выпадающий список для типа «сопоставление», «указание порядка», «указание истинности или ложности утверждений»). Визуально они различимы, поэтому обучаемый всегда может определить, какой тип вопроса ему предложен. Порядок следования вариантов ответов в исходном задании теста абсолютно несущественен (при выдаче тестов должно происходить случайное перемешивание порядка следования вариантов ответов). При использовании вопросов открытого типа со свободным вводом ответа (ввод произвольной последовательности для типов «ручного ввода числа», «ручного ввода текста») последовательность вводится в поле ввода в TextBox.

Методика оценивания результатов тестирования.

По завершении теста должен выдаваться результат тестирования. Следует учитывать количество правильных и не правильных ответов и соответственно количество набранных баллов, максимальное количество баллов, которые можно получить за тест.

## 3.5 Написание встроенного текстового редактора

В программе должны быть реализованы возможности форматирования текста вопросов и вариантов ответа. Возможность определить шрифт, цвет символов, разбивать текст на абзацы и применять к ним расширенное форматирование. Для большего удобства в программе потребуется реализовать собственный текстовый редактор.

Реализация добавления изображения и функции «Скриншот выделенного участка экрана» с последующей вставкой его в задание теста.

В вопросе могут присутствовать помимо текста рисунки. Эта часть вопроса не является обязательной, но все же рекомендуется ее применять, так как одна и та же ситуация может быть основой для множества однотипных вопросов. Помимо этого, полезной функцией было бы реализация процесса снятия скриншота произвольной области экрана с последующей вставкой полученного изображения в задание. Т.е. составитель тестовых заданий будет иметь возможность выделить нужную часть экрана, после чего отобразить ее как картинку к описанию в задании.

Архитектурное и детальное проектирование системы.

Проектирование системы проходило в несколько этапов, среди которых можно выделить следующие:

* разработка базы данных;
* разработка структуры программы.

## 3.6 Проектирование базы данных

Проектирование базы данных состоит из трех этапов: проектирование логической модели данных проектирование физической модели данных, реализации на конкретной СУБД.

## 3.6.1 Логическая модель базы данных

Кафедра - список кафедр:

* код кафедры - идентификатор кафедры;
* название кафедры - полное название кафедры;
* аббревиатура кафедры - аббревиатура названия кафедры.

Направления и специальности факультета:

* код специальности - идентификатор специальности;
* название специальности - полное название специальности;
* аббревиатура специальности - аббревиатура названия специальности.

Учебные группы - список учебных групп факультета:

* код группы - идентификатор группы;
* шифр группы;
* год - год формирования (приема) группы;
* код специальности - идентификатор специальности.

Дисциплины - изучаемые дисциплины:

* код дисциплины - идентификатор дисциплины;
* название - полное название;
* аббревиатура - аббревиатура названия;
* код специальности - идентификатор специальности.

Студент - информация о студенте:

* код студента - идентификатор студента;
* фамилия - фамилия студента;
* имя - имя студента;
* отчество - отчество студента;
* логин - логин для входа в систему;
* пароль - пароль для входа в систему;
* код группы - идентификатор учебной группы;
* код доступа - идентификатор в группе уровней доступа.

Преподаватель - информация о преподавателе:

* код преподавателя - идентификатор преподавателя;
* фамилия - фамилия преподавателя;
* имя - имя преподавателя;
* отчество - отчество преподавателя;
* логин - логин для входа в систему;
* пароль - пароль для входа в систему;
* код доступа - идентификатор в группе уровней доступа.

Права - уровень прав доступа в системе:

* код группы - идентификатор кода группы;
* тип группы - тип группы с правами доступа (студент, преподаватель).

Категории тестов:

* код категории - идентификатор категории;
* название - название категории (например, название дисциплины)

Список тестов - список тестов в категории:

* код теста - идентификатор теста;
* код категории - идентификатор категории, к которой принадлежит тест;
* название - название теста.

Список заданий - список заданий, содержащихся в тесте:

* код задания - идентификатор задания;
* название - название задания;
* код теста - идентификатор теста.

Тип задания - типы заданий для тестирования (одиночный выбор, множественный выбор и т.д.):

* код типа задания - идентификатор типа задания;
* название - название типа задания.

Вопросы - общий список вопросов:

* код вопроса - идентификатор вопроса;
* код задания - идентификатор задания;
* текст вопроса;
* изображение - изображение, прикрепленное к вопросу;
* тип - идентификатор типа задания;
* вес вопроса.

Список ответов:

* код ответа - идентификатор ответа;
* код вопроса - идентификатор вопроса;
* текст ответа - ответ на вопрос;
* эталон ответа;
* истинность - истинность ответа (верен или нет).

Результат - результат тестирования:

* код результата - идентификатор результата;
* код студента - идентификатор студента;
* код теста - идентификатор теста;
* количество верных ответов;
* количество неверных ответов;
* балл - количество набранных баллов во время тестирования;
* оценка - оценка тестирования.

3.7 Внедрение информационной системы

## 3.7.1 Установка и настройка базы данных на сервере.

Все что необходимо для установки двух баз данных на sql сервер – это присоединить их в среде SQL Server Management Studio. Подробнее описывать этот процесс нет необходимости, т.к. в требования по внедрению программы на предприятие описано, что для этого необходим сотрудник, специализирующийся на администрировании СУБД. Для сотрудника с данной квалификацией не вызывает проблем с подключением к SQL серверу этих баз данных.

## 3.7.2 Руководство пользователя

Разрабатываемая система осуществляет создание тестов и проведение компьютерного тестирования, сбора и анализа результатов, выставления оценки.

Программа легка и удобна в использовании и освоении. Предоставляет возможность работать с семью типами заданий: одиночный выбор, множественный выбор, установление порядка следования, установление соответствия, указание истинности или ложности утверждений, ручной ввод числа, ручной ввод текста. В тесте можно использовать любое количество любых типов, можно только один, можно и все сразу. В заданиях с выбором ответа (одиночный, множественный выбор, указание порядка, указание истинности) можно использовать до 10 (включительно) вариантов ответа.

Для создания тестов имеется очень удобный редактор тестов с дружественным интерфейсом. Любой преподаватель, даже владеющий компьютером на начальном уровне, может легко составить свои тесты и использовать их во время тестирования.

## 3.7.3 Системные требования к клиентской машине

Минимальные требования к оборудованию клиента следующие:

* процессор Intel Pentium III с частотой от 700 МГц или выше;
* оперативная память, доступная операционной системе, не менее 64Мб;
* монитор с поддержкой видеорежима 1024x768 при 256 цветах;
* манипулятор типа «мышь», устройство ввода - клавиатура.

Требования к программному обеспечению клиента следующие:

* любая операционная система (запуск из под Linux осуществляется с помощью Wine);
* сетевой доступ;
* .Net Framework 3.5 и выше.

## 3.7.4 Подключение и авторизация

При запуске программы на экране отображается окно авторизации (рисунок 7). В этом окне будет предложено заполнить поля логин и пароль для успешной авторизации в системе.

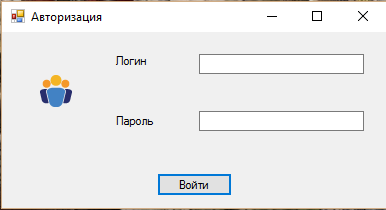


Рисунок 7 – Окно авторизации

## 3.7 Оценка экономических затрат на проект

Период проведения работ‎: c 7.05.2016 по 07.06.2016.

Работы проводились в одну смену продолжительностью 8 часов. Рассчитаем общую трудоемкость работ. Используем систему коэффициентов для отдельных этапов разработки. Определение условного количества операторов программы, трудоемкости (‎формула 1‎)‎:

|  |  |
| --- | --- |
| Т = То + Ти + Та + Тп + Тотл + Тд. | (‎1‎) |

Все составляющие определяем через условное число операторов – Q (‎формула 2‎)‎:

|  |  |
| --- | --- |
| Q = q \* c \* (‎1 + p‎). | (‎2‎) |

Коэффициент сложности c характеризует относительную сложность программы по отношению к так называемой типовой задаче, реализующей стандартные методы решения, сложность которой принята равной единице (‎величина с лежит в пределах от 1,25 до 2‎). Для разработанного программного продукта возьмем 1,3.

Коэффициент коррекции p – увеличение объема работ за счет внесения изменений в алгоритм по результатам уточнения постановок. С учетом того, что в данном случае заказчик, четко представлял, что он хочет получить, возьмем коэффициент = 0,9

В результате получим условное число операторов.

Q = q⋅c⋅(‎1 + p‎) = 700⋅1,3⋅(‎1+ 0,9‎) =1729.

Также используем следующие коэффициенты‎:

1. Коэффициент увеличения затрат труда, вследствие недостаточного описания задачи, в зависимости от сложности задачи принимается от 1,2 до 1,5, в связи с тем, что данная задача, не потребовала уточнения и больших доработок, примем B = 1,2.
2. Коэффициент квалификации разработчика k определяется в зависимости от стажа работы и составляет‎: для работающих до двух лет – 0,8; от двух до трех лет – 1,0; от трех до пяти лет – 1,1 – 1,2; от пяти до семи – 1,3 – 1,4; свыше семи лет – 1,5 – 1,6. Разработчик, которому было поручено это задание, имел опыт работы по специальности менее 1 года, поэтому примем k = 0,8.

Рассчитаем общую трудоемкость.

Затраты труда на подготовку описания задачи Тo точно определить невозможно, так как это связано с творческим характером работы. Примем

Тo = 16 чел.-‎ч.

Затраты труда на изучение описания задачи Ти с учетом уточнения описания и квалификации программиста могут быть определены по формуле 3‎:

|  |  |
| --- | --- |
| Ти = Q\*B / (‎75/85‎) k, | (‎3‎) |

Ти = 1729\*1,2 /75\*0,8 = 22,13 чел.-‎ч

Затраты труда на разработку алгоритма решения задачи Тa рассчитывается по формуле 4‎:

|  |  |
| --- | --- |
| Тa = Q / (‎20/25‎) k, | (‎4‎) |

Та = 1729/22\*0,8 = 62,8 чел.-‎ч.

Затраты труда на составление программы по готовой блок-‎схеме Тп определяется по формуле 5‎:

|  |  |
| --- | --- |
| Тп = Q / (‎20/25‎) k, | (‎5‎) |

Тп = 1729/22\*0,8 = 62,8 чел.-‎ч.

Затраты труда на отладку Тотл рассчитывается по следующей формуле 6‎:

|  |  |
| --- | --- |
| Тотл= Q / (‎4\*5‎) k, | (‎6‎) |

Тотл = 1729/5\*0,8 = 276,6 чел.-‎ч.

Затраты труда на подготовку документации по задаче Тд определяются по формуле 7‎:

|  |  |
| --- | --- |
| Тд = Тдр + Тдо, | (‎7‎) |

Затраты труда на подготовку материалов в рукописи формула 8‎:

|  |  |
| --- | --- |
| Тдр = Q / (‎15/20‎) k, | (‎8‎) |

Тдр = 1729/20\*0,8 = 69,16 чел.-‎ч.

Тдо – затраты труда на редактирование, печать и оформление документации (‎формула 9‎)‎:

|  |  |
| --- | --- |
| Тдо = 0,75 Тдр, | (‎9‎) |

Тдо = 0,75\*69,16 = 51,87 чел.-‎ч.

Тд = 69,16 +51,87 = 121,03 чел.-‎ч.

С учетом уровня языка программирования трудоемкость разработки программы может быть скорректирована следующим образом (‎формула 10‎)‎:

|  |  |
| --- | --- |
| Ткор = Е\*kкор , | (‎10‎) |

Таблица 2 – Изменение трудоемкости в зависимости уровня языка программирования

|  |  |  |
| --- | --- | --- |
| Уровень языка программирования | Характеристика языка программирования | Коэффициент изменения трудоемкости |
| 1 | Покомандный автокод-‎ Ассемблер | 1 |

Продолжение таблицы 2

|  |  |  |
| --- | --- | --- |
| 2 | Макроассемблер | 0,95 |
| 3 | Алгоритмические языки высокого уровня | 0,8 – 0,9 |
| 4 | Алгоритмические языки сверхвысокого уровня | 0,7 – 0,8 |

Выбранный для разработки язык C# относится к алгоритмическим языкам высокого уровня, с учетом этого примем kкор = 0,8

Подставив все полученные данные в формулу 1, получим полную трудоемкость разработки‎:

Т = 16+22,13+62,8+62,8+276,6 +121,03 = 561,36 чел.-‎ч.

С учетом корректировки из формулы 11 получим итоговую трудоемкость разработки‎:

Ткор = 0,8 \* 561,36 = 449,1 чел.-‎ч

Определение численности исполнителей (‎формула 11‎)‎:

|  |  |
| --- | --- |
| Ч = Т / Ф, | (‎11‎) |

При Ф = 240 часа найдем численность исполнителей‎:

Ч = 449,1 / 256 = 1,75 -‎ 1 исполнитель

В состав исполнителей входят‎: программист C#.

Распределение трудоемкости по стадиям разработки приведено в таблице 3.

Таблица 3 – Распределение трудоемкости по стадиям разработки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Этап разработки | Содержание работ | Трудоёмкость, ч. | Трудоёмкость работ исполнителя, ч. | Должность  исполнителя |
| Постановка задачи | 1. Предпроектное обследование.  2. Разработка ТЗ.  3. Разработка, согласование и утверждение технико-‎ экономического обоснования. | 36 | 36 | программист |
| Технический проект | Уточнение структуры и формы представления входных и выходных данных. Разработка алгоритма решения задачи. Разработка структуры программы. Разработка пояснительной записки. Согласование технического проекта и его утверждение. | 70 | 70 | программист |
| Рабочий проект | Отладка программы. Разработка дизайна программы. | 90 | 90 | программист |

Продолжение таблицы 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Документация и внедрение | Разработка программной документации. Подготовка и передача программы и программной документации для сопровождения и изготовления, оформления и утверждения акта о передаче программы на сопровождение. Передача программного продукта заказчику. | 60 | 60 | программист |
| ИТОГО‎: |  | 256 | 256 |  |

Расчет затрат на разработку

Основная заработная плата разработчика рассчитывается по формуле 12‎:

|  |  |
| --- | --- |
| ЗПосн = О \*1,15, | (‎12‎) |

где О – оклад, 1,15 – уральский коэффициент.

Его основная месячная заработная плата составит‎:

8000 \* 1,15 = 9200 рублей.

Основная заработная плата ‎программиста за весь период разработки программного продукта составит‎:

9200 руб. \* (‎256‎) ч. / (‎8ч. \* 32 дня‎) = 9200 руб

Дополнительная заработная плата рассчитывается в процентах от основной заработной платы и составляет 12%.

Дополнительная заработная плата разработчика за весь период разработки программного продукта составит‎:

9200\* 0,12 = 1104руб.

Суммарная заработная плата программиста за весь период разработки программного продукта составит‎:

9200 руб. + 1104 руб. = 10304 руб

Страховые взносы во внебюджетные фонды берутся в размере 30,2% от суммы основной и дополнительной заработной платы

Суммарные страховые взносы во внебюджетные фонды за весь период разработки программного продукта составят‎:

10304\*30,2%= 3111,808 руб

Содержание и эксплуатация вычислительного комплекса считается следующим образом (‎формула 13‎)‎:

|  |  |
| --- | --- |
| Свт = См-‎ч \* Число\_часов\_отладки, | (‎13‎) |

Число часов отладки составляет‎:

Тп + Тотл = (‎62,8ч.+ 276,6 ч.‎)\*0,7= 237,58 ч.

Стоимость машино-‎часа рассчитывается, как сумма составляющих‎:

|  |  |
| --- | --- |
| (‎Стть\_эл\_эн\_в\_год+Аморт\_в\_год+Затраты\_на\_ремонт\_за\_год‎)/Фвт, | (‎14‎) |
| Стоймось компьютера 32000руб.  А=32000\*0,25=8000 руб в год  А= (‎8000\*32‎)/365 = 701,37 рублей за период разработки  З=(‎32000\*0,03\*32‎)/365= 84,16 руб |  |

где Фвт – действительный фонд времени работы вычислительного комплекса.

Стоимость 1 КВТ/час электроэнергии составляет, например, 2,74 руб.

Один компьютер потребляет, например, 250 ВТ в час.

За месяц расходы на электроэнергию потребляемую одной ЭВМ составляет‎:

8 ч. \* 32 дня \* 0,25 КВТ/ч. \* 2,74 руб. \* ч. / КВТ = 175,36 руб

Действительный фонд времени работы вычислительного комплекса рассчитываем по следующей формуле 15‎:

|  |  |
| --- | --- |
| Фвт = Фном – Фпроф, | (‎15‎) |

Итак‎:

Фвт = 0,85 \* 256 час. = 217,6 часа.

Стоимость машино-‎часа составляет‎:

(‎175,36 руб. + 701,37 руб. + 84,16 руб.‎)/ 217,6 часа = 4,4руб.

Содержание и эксплуатация вычислительного комплекса составляет‎:

256ч. \* 4,4 руб. = 1126,4 руб.

Накладные расходы рассчитываются, как 60% от расходов на содержание и эксплуатацию вычислительного комплекса‎:

1126,4 руб.\*0,6 = 675,84 руб.

Смета затрат на разработку программного продукта приведена в таблице 4.

Таблица 4– «Смета затрат на разработку программного продукта»

|  |  |  |
| --- | --- | --- |
| № | Наименование статьи расходов | Затраты, руб. |
| 1 | Основная заработная плата производственного персонала. | 8000 |
| 2 | Дополнительная заработная плата производственного персонала. | 1104 |
| 3 | Страховые взносы во внебюджетные фонды | 3111,8 |
| 4 | Содержание и эксплуатация вычислительного комплекса | 1126,4 |
| 5 | Накладные расходы. | 675,84 |
| ИТОГО‎: |  | 14018,04 |

Применение данной разработки позволит в самые кратчайшие сроки компенсировать затраты на разработку и эксплуатацию, получить экономический эффект от использования данного комплекса.

В ходе вышеприведённых вычислений были получены следующие результаты‎:

* была определена трудоёмкость разработки программного продукта, которая составила‎: 449,1 чел.-‎час.;
* рассчитано, что для выполнения данной разработки в планируемый период понадобится 1 человек;
* была рассчитана смета затрат на разработку программного продукта, итоговая сумма которой‎: 14018,04 руб.

# ВЫВОД ПО ГЛАВЕ 3

В данной главе была представлена разработка автоматизированной информационной системы, приведены рисунки форм интерфейса, описаны используемые при разработке элементы управления, а также их функциональное применение.

Была проведена оценка автоматизированной информационной системы по таким параметрам, как:

* надежность;
* сопровождаемость;
* удобство применения и т.д.

Была описана процедура внедрения программы на предприятии, которая сводится к включению двух баз данных в SQL server и изменению двух строковых параметров, отвечающая за инициализацию подключения к источнику баз данных.

Вывод главы заключается в том, что разработанное программное обеспечение полностью соответствует требованиям.

# ЗАКЛЮЧЕНИЕ

Основная цель данной работы заключалась в автоматизации работы преподавателей по тестированию студентов. Задача автоматизации состояла в минимизации взаимодействия конечного пользователя с различным программным обеспечением путем консолидации необходимого функционала в разработанном программном продукте.

# Список литературы

Нормативно-правовые акты

1. ГОСТ 19.102-77 (СТ СЭВ 1626-79). Единая система программной документации. Стадии разработки : Гос. стандарт. – Введ. 01.07 1978 (С учетом изменения №1, утвержденным в июне 1981 г.) //Утвержден постановлением государственного совета стандартов Совета Министров СССР от 20 мая 1977 г. № 1268.
2. ГОСТ 34.602-89. Техническое задание на создание автоматизированной системы: Гос. стандарт. – Введ. 01.01.1990 // утверждён приказом роспрома 16.09.2004 №95.
3. ГОСТ 28195-89. Оценка качества программных средств: Гос. стандарт. – Введ. 01.07.1990 //Утвержден постановлением государственного комитета СССР по стандартам от 28.07.89 №2507.
4. ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов. Гос. стандарт. – Введ. 14.07.1998 //Утвержден постановлением Госстандарта РФ от 14.07.1998 №295.
5. ГОСТ Р 34.10-2012. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи: Гос. стандарт. – Введ. 07.08.2012 //утвержден приказом федерального агентства по техническому регулированию и метрологии от 07.08.2012 №215;

Основная литература

1. Абрамов Г. В., Медведкова И. Е., Коробова Л. А. Проектирование информационных систем [Электронный ресурс]: учебное пособие/Г. В. Абрамов, И. Е. Медведкова, Л. А. Коробова, Воронеж гос. ун-т инж. технол. – Воронеж : ВГУИТ, 2012. - 172 с. URL: biblioclub.ru/index.php?page=book\_view \_red&book\_id=141626 (15.10.2015)
2. Котов, О.М. Язык C#: краткое описание и введение в технологии программирования [Электронный ресурс]: учебное пособие / О.М. Котов; Министерство образования и науки Российской Федерации, Уральский федеральный университет имени первого Президента России Б. Н. Ельцина. - Екатеринбург: Издательство Уральского университета, 2014. - 209 с. URL: http://biblioclub.ru/index.php?page=book&id=275809 (15.10.2015).
3. Ликнесс, Д. Приложения для Windows 8 на C# и XAML / Д. Ликнесс. – СПб.: Питер, 2013. – 368 с.
4. Методология IDEF0 [Электронный ресурс], 2014. – Режим доступа: http://itteach.ru/bpwin/metodologiya-idef0.
5. Пайлон, Д. UML 2 для программистов / Д. Пайлон, Н. Питмен. – СПб.: Питер, 2012. -240 с.
6. Петцольд, Ч. Программирование для Microsoft Windows 8. 6-е изд. / Ч. Петцольд. – СПб.: Питер, 2013. – 1008 с.
7. Рихтер, Д. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е изд. СПб.: Питер, 2014. – 896 с.
8. Смоленцев, Н.К. MATLAB: Программирование на Visual C#, Borland C#, JBuilder, VBA / Н.К. Смоленцев. - М. [Электронный ресурс]: ДМК Пресс, б.г. - 453 с. URL: http://biblioclub.ru/index.php?page=book&id=86284 (15.10.2015).
9. Суханов, М.В. Основы Microsoft .NET Framework и языка программирования C# [Электронный ресурс]: учебное пособие / М.В. Суханов, И.В. Бачурин, И.С. Майоров; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования Северный (Арктический) федеральный университет им. М.В. Ломоносова. - Архангельск : ИД САФУ, 2014. - 97 с. URL: http://biblioclub.ru/index.php?page=book&id=312313 (15.10.2015).
10. Стиллмен, Э. Изучаем C#. 3-е изд. /Э. Стиллмен. – СПб.: Питер, 2014. – 816 с.
11. Швецов, В.И. Базы данных / В.И. Швецов. – М. : Интернет-Университет Информационных Технологий, 2012. - 195 с. ; То же [Электронный ресурс]. – URL: http://biblioclub.ru/index.php?page=book&id=234676 (16.12.2014).

Дополнительная литература

1. Биллиг, В.А. Основы программирования на C# [Электронный ресурс]: учебный курс / В.А. Биллиг. - М.: Интернет-Университет Информационных Технологий, 2006. - 485 с. URL: http://biblioclub.ru/index.php?page=book&id=233695 (15.10.2015).
2. Бизнес – процесс [Электронный ресурс] - <http://www.hr-portal.ru/varticle/biznes>-process - словарь по управлению персоналом.
3. Гаврилов, М.В. Информатика и информационные технологии‎: Учебник для бакалавров / М.В. Гаврилов, В.А. Климов; Рецензент Л.В. Кальянов, Н.М. Рыскин. -‎ М.‎: Юрайт, 2013. -‎ 378 c.
4. Голицына, О.Л. Системы управления базами данных / О. Л. Голицына, Т. Л. Партыка, И.И. Попов. –М.: Инфра-М, 2011. – 432 с.
5. Грабауров, В.А. Информационные технологии [Электронный ресурс] - http://www.directmedia.ru/?page=book\_view&book\_id=225542: В.А. Грабауров – Москва: финансы и статистика, 2012 с. 87 – 94.
6. Жданов С.А. Информационные системы: для студ. Учреждений высш. Образования / С.А. Жданов, М.Л.Соболева, А.С.Алфимова. – М.:ООО «Прометей», 2015. – 320 с. то же [электронный ресурс].-URL: http://biblioclub.ru/index.php?page=book\_red&id=426722&sr=1 (22.03.2016).
7. Кариев, Ч.А. Разработка Windows-приложений на основе Visual C# [Электронный ресурс]: учебное пособие / Ч.А. Кариев. - М. : Интернет-Университет Информационных Технологий, 2007. - 768 с. URL: http://biblioclub.ru/index.php?page=book&id=233307 (15.10.2015).
8. Методология проектирования [Электронный ресурс] - <http://www.monographies.ru/141-4641> - сайт российской академии естествознания.
9. MicrosoftVisio [Электронный ресурс] - <http://office.microsoft.com/ru-ru/visio/> - сайт производителя Microsoft.
10. Милезина О.В. информационные системы: теоретические предпосылки к построению: учебное пособие/ О.В. Милехина, Е.Я. Захарова, В.А.Титова-2-е изд-е. – Новосибирск: Изд-во НГТУ, 2014. – 283 с. то же [электронный ресурс].-URL: http://biblioclub.ru/index.php?page=book\_red&id=258420&sr=1 (22.03.2016).
11. Персианов, В.В. Информационные системы: учебно-методическое пособие / В.В. Персианов, В.И. Логвинова. – М.-Берлин: Директ-Медиа 2016. – 191 с.; То же [электронный ресурс].-URL: http://biblioclub.ru/index.php?page=book\_view\_red&book\_id=96968 (22.03.2016).
12. Федорова, Г.Н. Информационные системы‎: Учебник для студ. учреждений сред. проф. образования / Г.Н. Федорова. -‎ М.‎: ИЦ Академия, 2013. -‎ 208 c.
13. Хейлсберг, А. Язык программирования C#. Классика Computers Science. 4-е изд. / А. Хейлберг. – СПб.: Питер, 2011. – 784 с.
14. Шмуллер, Д. Освой самостоятельно UML 2 за 24 часа [Текст]: Практическое руководство/М.Вильямс, 2011. — 416 с.

# ПРИЛОЖЕНИЕ А

Декомпозиция блока «Создание теста»



# ПРИЛОЖЕНИЕ Б

Декомпозиция блока «Редактирование теста»



# ПРИЛОЖЕНИЕ В

Декомпозиция блока «Тестирование студентов»

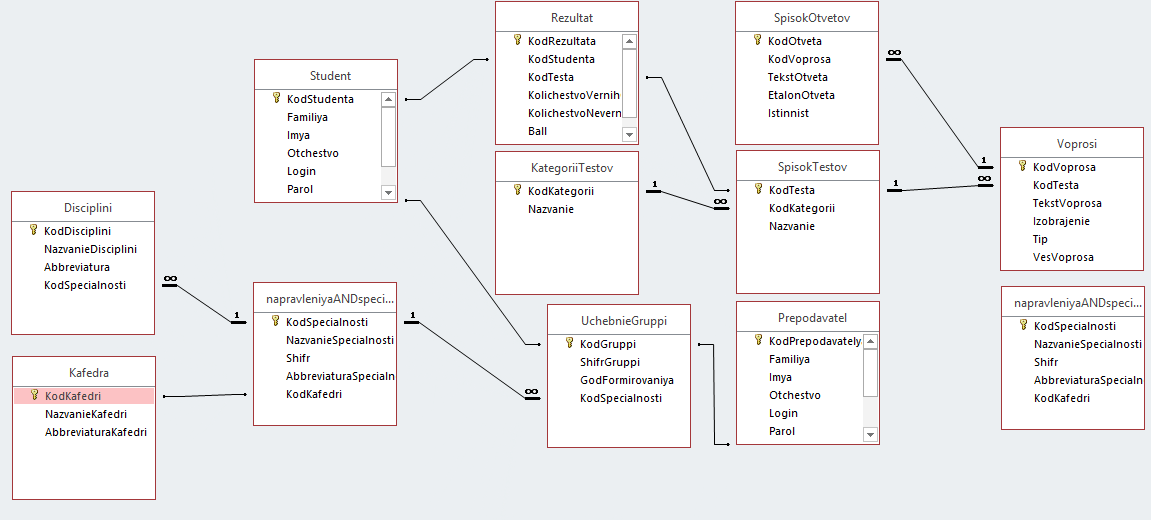


# ПРИЛОЖЕНИЕ Г

Декомпозиция блока «Просмотр результатов»

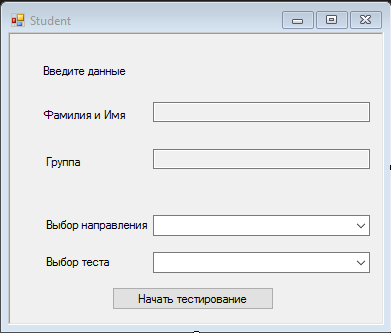


# ПРИЛОЖЕНИЕ Д

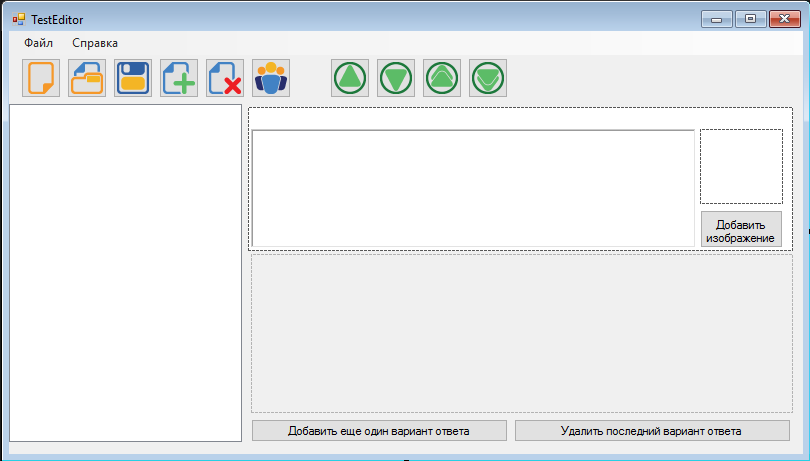
Схема базы данных

# ПРИЛОЖЕНИЕ Е

Скриншот окна ввода данных студента перед тестированием



# ПРИЛОЖЕНИЕ Ж

Скриншот окна создания теста

# ПРИЛОЖЕНИЕ З

Листинг программы

Main Form.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

using System.Security.Cryptography;

namespace TestRed

{

public partial class MainForm : Form

{

OleDbConnection conn;

OleDbDataReader rd;

int[] student\_id = new int[999];

string[] student\_login = new string[999];

string[] student\_pass = new string[999];

int[] teacher\_id = new int[999];

string[] teacher\_login = new string[999];

string[] teacher\_pass = new string[999];

public MainForm()

{

conn = new OleDbConnection("Provider=Microsoft.Ace.OLEDB.12.0;" + "Data Source= Database.accdb;" + "Persist Security Info=False;");

DBTable();

InitializeComponent();

Продолжение приложения Ж

}

void DBTable()

{

conn.Open();

OleDbCommand cmd = new OleDbCommand("SELECT Login, Parol, KodStudenta FROM Student",conn); //login:13100234 pass: 7512244551

rd = cmd.ExecuteReader();

int i = 0;

while (rd.Read())

{

student\_login[i] = rd[0].ToString();

student\_pass[i] = rd[1].ToString();

student\_id[i] = (int)rd[2];

i++;

}

rd.Close();

cmd.CommandText = "SELECT Login, Parol, KodPrepodavatelya FROM Prepodavatel"; //login:LebedevaTN pass:12345678

rd = cmd.ExecuteReader();

i = 0;

while (rd.Read())

{

teacher\_login[i] = rd[0].ToString();

teacher\_pass[i] = rd[1].ToString();

teacher\_id[i] = (int)rd[2];

i++;

}

rd.Close();

conn.Close();

}

//string a, b;

private void button1\_Click(object sender, EventArgs e)

{

if (textBox1.Text == String.Empty)//Проверка на пустоту текстбокса

MessageBox.Show("Введите логин");

else

{

Продолжение приложения Ж

if (textBox1.Text == Array.Find(student\_login, str => str == textBox1.Text))

{

if (CalculateMD5Hash(textBox2.Text) == Array.Find(student\_pass, str => str == CalculateMD5Hash(textBox2.Text)))

{

Student newForm = new Student(student\_id[Array.IndexOf(student\_login, textBox1.Text)]);

newForm.Show();

this.Hide();

}

else

{

MessageBox.Show("Пароли не совпадают.");

}

}

else

{

if (textBox1.Text == Array.Find(teacher\_login, str => str == textBox1.Text))

{

if (CalculateMD5Hash(textBox2.Text) == Array.Find(teacher\_pass, str => str == CalculateMD5Hash(textBox2.Text)))

{

TestEditor newForm = new TestEditor();

newForm.Show();

this.Hide();

}

else

{

MessageBox.Show("Пароли не совпадают.");

}

}

else

{

MessageBox.Show("Логин не найден.");

}

}

}

}

public string CalculateMD5Hash(string input)

{

Продолжение приложения Ж

// step 1, calculate MD5 hash from input

MD5 md5 = System.Security.Cryptography.MD5.Create();

byte[] inputBytes = System.Text.Encoding.ASCII.GetBytes(input);

byte[] hash = md5.ComputeHash(inputBytes);

// step 2, convert byte array to hex string

StringBuilder sb = new StringBuilder();

for (int i = 0; i < hash.Length; i++)

{

sb.Append(hash[i].ToString("x2"));

}

return sb.ToString();

}

private void MainForm\_FormClosed(object sender, FormClosedEventArgs e)

{

Application.Exit();

}

}

}

Test.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

Продолжение приложения Ж

using System.Windows.Forms;

using System.Data.OleDb;

using System.IO;

namespace TestRed

{

public partial class Test : Form

{

int test\_id;

int student\_id;

int tek\_vopr = 0;

int otv\_et = 0;

int kol\_vern\_otv = 0;

int kol\_nevern\_otv = 0;

int[] last\_vopr = new int[100];

OleDbConnection conn;

OleDbDataReader rd;

List<int> vopr\_id = new List<int>();

string[] vopr\_tekst = new string[100];

string[] vopr\_img = new string[100];

List<int> true\_otv = new List<int>();

Dictionary<string, int> etalon = new Dictionary<string, int>();

Random rnd = new Random();

public Test(int id\_test, int id\_st)

{

test\_id = id\_test;

student\_id = id\_st;

conn = new OleDbConnection("Provider=Microsoft.Ace.OLEDB.12.0;" + "Data Source= Database.accdb;" + "Persist Security Info=False;");

InitializeComponent();

}

private void Test\_Load(object sender, EventArgs e)

{

conn.Open();

OleDbCommand cmd = new OleDbCommand("SELECT KodVoprosa, TekstVoprosa, Izobrajenie FROM Voprosi WHERE KodTesta = " + test\_id, conn);

rd = cmd.ExecuteReader();

int i = 0;

while(rd.Read())

Продолжение приложения Ж

{

vopr\_id.Add((int)rd[0]);

vopr\_tekst[i] = rd[1].ToString();

vopr\_img[i] = rd[2].ToString();

i++;

}

rd.Close();

conn.Close();

i = tek\_vopr;

tek\_vopr++;

label1.Text = vopr\_tekst[i];

label2.Text = "Вопрос " + tek\_vopr + " из " + vopr\_id.Count + ".";

pictureBox1.ImageLocation = vopr\_img[i];

otv(vopr\_id[i]);

progressBar1.Maximum = vopr\_id.Count;

progressBar1.Step = 1;

}

private void otv(int id\_vopr)

{

flowLayoutPanel2.Controls.Clear();

conn.Open();

OleDbCommand cmd = new OleDbCommand("SELECT TekstOtveta, EtalonOtveta FROM SpisokOtvetov WHERE KodVoprosa = " + id\_vopr, conn);

rd = cmd.ExecuteReader();

int i = 0;

while (rd.Read())

{

RadioButton rb = new RadioButton();

rb.Text = "";

rb.Size = new Size(14, 13);

rb.Tag = i;

TextBox tb = new TextBox();

tb.Size = new Size(600, 20);

tb.Text = rd[0].ToString();

tb.ReadOnly = true;

tb.Tag = i;

tb.Click += textBox\_Click;

etalon.Add(rd[0].ToString(),Convert.ToInt32(rd[1]));

Продолжение приложения Ж

flowLayoutPanel2.Controls.Add(rb);

flowLayoutPanel2.Controls.Add(tb);

i++;

}

rd.Close();

conn.Close();

}

private void seachEtalonOtv()

{

int i = 0;

foreach (var item in flowLayoutPanel2.Controls.OfType<RadioButton>())

{

if (item.Checked)

{

i = (int)item.Tag;

}

}

foreach (var item in flowLayoutPanel2.Controls.OfType<TextBox>())

{

if ((int)item.Tag == i)

{

otv\_et += etalon[item.Text];

if (etalon[item.Text] == 0)

kol\_nevern\_otv++;

else

kol\_vern\_otv++;

}

}

}

private void button1\_Click(object sender, EventArgs e)

{

true\_otv.Add(tek\_vopr);

seachEtalonOtv();

if (vopr\_id.Count != true\_otv.Count)

{

int i = tek\_vopr;

tek\_vopr++;

Продолжение приложения Ж

label1.Text = vopr\_tekst[i];

label2.Text = "Вопрос " + tek\_vopr + " из " + vopr\_id.Count + ".";

pictureBox1.ImageLocation = vopr\_img[i];

otv(vopr\_id[i]);

progressBar1.PerformStep();

}

else

{

Finish();

}

}

private void Finish()

{

progressBar1.Value = progressBar1.Maximum;

flowLayoutPanel2.Controls.Clear();

pictureBox1.ImageLocation = "";

label1.Text = "Вы закончили тестирование. Нажмите 'Закончить тестирование' снизу окна.";

label2.Text = "";

Label lb = new Label();

lb.Dock = DockStyle.Fill;

lb.Font = label1.Font;

lb.AutoSize = true;

lb.Text = "Вы получили " + otv\_et + " баллов из " + vopr\_id.Count \* 3;

lb.Text += "\nВаша оценка: " + Ocenka();

flowLayoutPanel2.Controls.Add(lb);

button1.Enabled = false;

pictureBox1.Enabled = false;

}

private void button2\_Click(object sender, EventArgs e)

{

if (progressBar1.Value != progressBar1.Maximum)

{

if (MessageBox.Show("Вы действительно хотите закончить тестирование? \nВ случае продолжения вы не получите баллы за не пройденные вопросы.", "Внимание", MessageBoxButtons.OKCancel) == DialogResult.OK)

{

Finish();

kol\_nevern\_otv += vopr\_id.Count - kol\_vern\_otv;

Продолжение приложения Ж

}

}

else

{

conn.Open();

OleDbCommand cmd = new OleDbCommand("INSERT INTO Rezultat (KodStudenta, KodTesta, KolichestvoVernihOtvetov, KolichestvoNevernihOtvetov, Ball, Ocenka) VALUES ('" + student\_id + "', '" + test\_id + "', '" + kol\_vern\_otv + "', '" + kol\_nevern\_otv + "', '" + otv\_et + "', '" + Ocenka() + "')", conn);

rd = cmd.ExecuteReader();

rd.Close();

conn.Close();

this.Close();

}

}

private int Ocenka()

{

if (otv\_et >= (vopr\_id.Count \* 3) \* 0.5)

{

if (otv\_et >= vopr\_id.Count \* 3 \* 0.7)

{

if (otv\_et >= vopr\_id.Count \* 3 \* 0.7)

{

return 5;

}

else

return 4;

}

else

{

return 3;

}

}

else

return 2;

}

private void Test\_FormClosed(object sender, FormClosedEventArgs e)

{

Application.Exit();

}

Продолжение приложения Ж

private void pictureBox1\_Click(object sender, EventArgs e)

{

Form1 frm = new Form1(pictureBox1.ImageLocation);

frm.Show();

}

private void textBox\_Click(object sender, EventArgs e)

{

TextBox tb = sender as TextBox;

foreach (var item in flowLayoutPanel2.Controls.OfType<RadioButton>())

{

if ((int)item.Tag == (int)tb.Tag)

{

item.Checked = true;

}

}

}

}

}

Student.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace TestRed

{

public partial class Student : Form

{

OleDbConnection conn;

OleDbDataReader rd;

int student\_id;

Продолжение приложения Ж

string student\_f = "";

string student\_i = "";

string student\_g = "";

Dictionary<string, int> Test = new Dictionary<string, int>();

Dictionary<string, int> test\_id = new Dictionary<string, int>();

int selected\_test\_id = 0;

public Student(int id)

{

student\_id = id;

conn = new OleDbConnection("Provider=Microsoft.Ace.OLEDB.12.0;" + "Data Source= Database.accdb;" + "Persist Security Info=False;");

InitializeComponent();

DBCon();

textBox1.Text = student\_f + " " + student\_i;

textBox2.Text = student\_g;

}

private void button1\_Click(object sender, EventArgs e)

{

this.Visible = false;

if (selected\_test\_id != 0)

{

Test newForm = new Test(selected\_test\_id, student\_id);

newForm.Show();

}

}

void DBCon()

{

string id\_group;

conn.Open();

OleDbCommand cmd = new OleDbCommand("SELECT Familiya, Imya, KodGruppi FROM Student WHERE KodStudenta = " + student\_id, conn);

rd = cmd.ExecuteReader();

rd.Read();

student\_f = rd[0].ToString();

student\_i = rd[1].ToString();

id\_group = rd[2].ToString();

rd.Close();

Продолжение приложения Ж

cmd.CommandText = "SELECT ShifrGruppi FROM UchebnieGruppi WHERE KodGruppi = " + id\_group;

rd = cmd.ExecuteReader();

rd.Read();

student\_g = rd[0].ToString();

rd.Close();

cmd = new OleDbCommand("SELECT KodKategorii, Nazvanie FROM KategoriiTestov", conn);

rd = cmd.ExecuteReader();

while (rd.Read())

{

Test.Add(rd[1].ToString(), (int)rd[0]);

comboBox1.Items.Add(rd[1].ToString());

}

rd.Close();

rd.Close();

conn.Close();

}

private void Student\_Load(object sender, EventArgs e)

{

}

private void comboBox1\_SelectionChangeCommitted(object sender, EventArgs e)

{

ComboBox cbox = sender as ComboBox;

conn.Open();

OleDbCommand cmd = new OleDbCommand("SELECT KodTesta, Nazvanie FROM SpisokTestov WHERE KodKategorii = " + Test[cbox.SelectedItem.ToString()], conn);

rd = cmd.ExecuteReader();

comboBox2.Items.Clear();

while (rd.Read())

{

test\_id.Add(rd[1].ToString(), (int)rd[0]);

comboBox2.Items.Add(rd[1].ToString());

}

rd.Close();

Продолжение приложения Ж

conn.Close();

}

private void comboBox2\_SelectionChangeCommitted(object sender, EventArgs e)

{

ComboBox cbox = sender as ComboBox;

selected\_test\_id = test\_id[cbox.SelectedItem.ToString()];

}

private void Student\_FormClosed(object sender, FormClosedEventArgs e)

{

Application.Exit();

}

}

}

Test Editor.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

using System.IO;

namespace TestRed

{

public partial class TestEditor : Form

{

int kod\_otveta = 0;

int number = 0;

int current\_test;

string current\_path\_img;

//int current\_vopr;

Продолжение приложения Ж

OleDbConnection conn;

OleDbDataReader rd;

Dictionary<string, int> test = new Dictionary<string, int>();

List<int> vopr\_id = new List<int>();

List<string> vopr\_text = new List<string>();

List<string> vopt\_img = new List<string>();

int count\_otv = 0;

public TestEditor()

{

conn = new OleDbConnection("Provider=Microsoft.Ace.OLEDB.12.0;" + "Data Source= Database.accdb;" + "Persist Security Info=False;");

InitializeComponent();

}

private void TestEditor\_Load(object sender, EventArgs e)

{

conn.Open();

OleDbCommand cmd = new OleDbCommand("SELECT KodTesta, Nazvanie FROM SpisokTestov", conn); //

rd = cmd.ExecuteReader();

while (rd.Read())

{

treeView1.Nodes.Add(rd[1].ToString());

test.Add(rd[1].ToString(), Convert.ToInt32(rd[0]));

}

rd.Close();

conn.Close();

}

private void выходToolStripMenuItem\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void button1\_Click(object sender, EventArgs e)

{

RadioButton rb = new RadioButton();

rb.Text = "";

rb.Size = new Size(14, 13);

Продолжение приложения Ж

rb.Tag = count\_otv;

TextBox tb = new TextBox();

tb.Size = new Size(500, 20);

tb.Tag = count\_otv;

count\_otv++;

flowLayoutPanel1.Controls.Add(rb);

flowLayoutPanel1.Controls.Add(tb);

}

private void LoadQuestion()

{

flowLayoutPanel1.Controls.Clear();

richTextBox1.Text = vopr\_text[number];

pictureBox1.ImageLocation = vopt\_img[number];

if (vopt\_img[number] != "")

{

pictureBox1.ImageLocation = vopt\_img[number];

}

label1.Text = "Вопрос "+(number + 1).ToString()+" из "+vopr\_id.Count;

conn.Open();

OleDbCommand cmd = new OleDbCommand("SELECT KodOtveta, TekstOtveta, EtalonOtveta FROM SpisokOtvetov WHERE KodVoprosa = " + vopr\_id[number], conn); //

rd = cmd.ExecuteReader();

count\_otv = 0;

while (rd.Read())

{

RadioButton rb = new RadioButton();

rb.Text = "";

rb.Size = new Size(14, 13);

rb.Tag = count\_otv;

if (rd[2].ToString() == "3")

rb.Checked = true;

else

rb.Checked = false;

TextBox tb = new TextBox();

tb.Size = new Size(500, 20);

tb.Text = rd[1].ToString();

tb.ReadOnly = false;

tb.Tag = count\_otv;

Продолжение приложения Ж

if (kod\_otveta < Convert.ToInt32(rd[0]))

kod\_otveta = Convert.ToInt32(rd[0]);

flowLayoutPanel1.Controls.Add(rb);

flowLayoutPanel1.Controls.Add(tb);

count\_otv++;

}

rd.Close();

conn.Close();

}

private void treeView1\_AfterSelect(object sender, TreeViewEventArgs e)

{

TreeNode nd = treeView1.SelectedNode;

label2.Text = "";

number = 0;

conn.Open();

current\_test = test[nd.Text];

OleDbCommand cmd = new OleDbCommand("SELECT KodVoprosa, TekstVoprosa, Izobrajenie FROM Voprosi WHERE KodTesta = " + current\_test, conn); //

label2.Text = nd.Text;

rd = cmd.ExecuteReader();

vopr\_id.Clear();

vopr\_text.Clear();

while (rd.Read())

{

vopr\_id.Add(Convert.ToInt32(rd[0]));

vopr\_text.Add(rd[1].ToString());

vopt\_img.Add(rd[2].ToString());

}

rd.Close();

conn.Close();

LoadQuestion();

}

private void button9\_Click(object sender, EventArgs e)

{

if (number == vopr\_id.Count - 1)

number = 0;

else

number++;

Продолжение приложения Ж

LoadQuestion();

}

private void button10\_Click(object sender, EventArgs e)

{

if (number > 0)

number--;

else

number = vopr\_id.Count - 1;

LoadQuestion();

}

private void button11\_Click(object sender, EventArgs e)

{

number = vopr\_id.Count - 1;

LoadQuestion();

}

private void button12\_Click(object sender, EventArgs e)

{

number = 0;

LoadQuestion();

}

private void button8\_Click(object sender, EventArgs e)

{

Form mainForm = new MainForm();

mainForm.Show();

this.Hide();

}

private void TestEditor\_FormClosed(object sender, FormClosedEventArgs e)

{

Application.Exit();

}

private void button5\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Вы действительно хотите сохранить задание в базу данных?","Внимание", MessageBoxButtons.OKCancel) == DialogResult.OK)

{

Продолжение приложения Ж

conn.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = conn;

int number\_vopros;

if (number != vopr\_id.Count)

{

cmd.CommandText = "DELETE FROM SpisokOtvetov WHERE KodVoprosa = " + vopr\_id[number];

rd = cmd.ExecuteReader();

rd.Close();

cmd.CommandText = "DELETE FROM Voprosi WHERE KodVoprosa = " + vopr\_id[number];

rd = cmd.ExecuteReader();

rd.Close();

}

cmd.CommandText = "INSERT INTO Voprosi (KodTesta, TekstVoprosa, izobrajenie, VesVoprosa) VALUES ('" + current\_test + "', '" + richTextBox1.Text + "', '"+pictureBox1.ImageLocation+"', 3)";

rd = cmd.ExecuteReader();

rd.Close();

for (int i = 0; i < flowLayoutPanel1.Controls.OfType<TextBox>().Count(); i++)

{

string textOtv = "";

string etalonOtv = "0";

bool istinnost = false;

foreach (var item in flowLayoutPanel1.Controls.OfType<TextBox>())

{

if (item.Tag.ToString() == i.ToString())

textOtv = item.Text;

}

foreach (var item in flowLayoutPanel1.Controls.OfType<RadioButton>())

{

if (item.Tag.ToString() == i.ToString())

if (item.Checked)

{

etalonOtv = "3";

Продолжение приложения Ж

istinnost = true;

}

else

{

etalonOtv = "0";

istinnost = false;

}

}

cmd.CommandText = "SELECT KodVoprosa FROM Voprosi WHERE TekstVoprosa = " + "'" + richTextBox1.Text + "'";

rd = cmd.ExecuteReader();

rd.Read();

number\_vopros = Convert.ToInt32(rd[0]);

rd.Close();

cmd.CommandText = "INSERT INTO SpisokOtvetov (KodVoprosa, TekstOtveta, EtalonOtveta, Istinnist) VALUES ('" + number\_vopros + "','" + textOtv + "', '" + etalonOtv + "', '"+istinnost+"' )";

rd = cmd.ExecuteReader();

rd.Close();

}

conn.Close();

ReloadBase();

}

MessageBox.Show("Задание успешно сохранено");

}

private void button13\_Click(object sender, EventArgs e)

{

flowLayoutPanel1.Controls.Remove(flowLayoutPanel1.Controls.OfType<RadioButton>().Last());

flowLayoutPanel1.Controls.Remove(flowLayoutPanel1.Controls.OfType<TextBox>().Last());

}

private void button6\_Click(object sender, EventArgs e)

{

conn.Open();

richTextBox1.Text = "Введите описание задания:";

flowLayoutPanel1.Controls.Clear();

Продолжение приложения Ж

label1.Text = "Вопрос " + (vopr\_id.Count + 1) + " из " + (vopr\_id.Count + 1);

count\_otv = 0;

for (int i = 0; i < 3; i++)

{

RadioButton rb = new RadioButton();

rb.Text = "";

rb.Size = new Size(14, 13);

rb.Tag = count\_otv;

TextBox tb = new TextBox();

tb.Size = new Size(500, 20);

tb.Text = "";

tb.ReadOnly = false;

tb.Tag = count\_otv;

flowLayoutPanel1.Controls.Add(rb);

flowLayoutPanel1.Controls.Add(tb);

count\_otv++;

}

number = vopr\_id.Count;

conn.Close();

}

private void ReloadBase()

{

conn.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = conn;

cmd.CommandText = "SELECT KodVoprosa, TekstVoprosa FROM Voprosi WHERE KodTesta = " + current\_test;

rd = cmd.ExecuteReader();

vopr\_id.Clear();

vopr\_text.Clear();

while (rd.Read())

{

vopr\_id.Add(Convert.ToInt32(rd[0]));

vopr\_text.Add(rd[1].ToString());

}

rd.Close();

conn.Close();

Продолжение приложения Ж

}

private void button7\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Вы действительно хотите удалить задание из базы данных?", "Внимание", MessageBoxButtons.OKCancel) == DialogResult.OK)

{

conn.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = conn;

cmd.CommandText = "DELETE FROM SpisokOtvetov WHERE KodVoprosa = " + vopr\_id[number];

rd = cmd.ExecuteReader();

rd.Close();

cmd.CommandText = "DELETE FROM Voprosi WHERE KodVoprosa = " + vopr\_id[number];

rd = cmd.ExecuteReader();

rd.Close();

conn.Close();

}

MessageBox.Show("Задание удалено.");

vopr\_text.Remove(richTextBox1.Text);

number = 0;

ReloadBase();

LoadQuestion();

}

private void button3\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Вы действительно хотите сохранить задание в файл?", "Внимание", MessageBoxButtons.OKCancel) == DialogResult.OK)

{

StreamWriter sw = new StreamWriter("test/"+current\_test+ "\_" + number +".txt");

sw.WriteLine(richTextBox1.Text);

for (int i = 0; i < flowLayoutPanel1.Controls.OfType<TextBox>().Count(); i++)

{

string textOtv = "";

string etalonOtv = "0";

Продолжение приложения Ж

foreach (var item in flowLayoutPanel1.Controls.OfType<TextBox>())

{

if (item.Tag.ToString() == i.ToString())

textOtv = item.Text;

}

foreach (var item in flowLayoutPanel1.Controls.OfType<RadioButton>())

{

if (item.Tag.ToString() == i.ToString())

if (item.Checked)

etalonOtv = "3";

else

etalonOtv = "0";

}

sw.WriteLine(textOtv);

sw.WriteLine(etalonOtv);

}

sw.Close();

}

MessageBox.Show("Задание успешно сохранено.");

}

private void button4\_Click(object sender, EventArgs e)

{

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

StreamReader sr = new StreamReader(openFileDialog1.FileName);

richTextBox1.Text = sr.ReadLine();

flowLayoutPanel1.Controls.Clear();

label1.Text = "Вопрос " + (vopr\_id.Count + 1) + " из " + (vopr\_id.Count + 1);

count\_otv = 0;

while(sr.EndOfStream == false)

{

TextBox tb = new TextBox();

tb.Size = new Size(500, 20);

tb.Text = sr.ReadLine();

tb.ReadOnly = false;

tb.Tag = count\_otv;

Продолжение приложения Ж

RadioButton rb = new RadioButton();

rb.Text = "";

rb.Size = new Size(14, 13);

rb.Tag = count\_otv;

string t = sr.ReadLine();

if (t == "3")

rb.Checked = true;

else

rb.Checked = false;

flowLayoutPanel1.Controls.Add(rb);

flowLayoutPanel1.Controls.Add(tb);

count\_otv++;

}

sr.Close();

number = vopr\_id.Count;

}

}

private void button2\_Click(object sender, EventArgs e)

{

if (openFileDialog2.ShowDialog() == DialogResult.OK)

{

current\_path\_img = "image/" + openFileDialog2.SafeFileName;

if (openFileDialog2.FileName != current\_path\_img)

File.Copy(openFileDialog2.FileName, current\_path\_img);

pictureBox1.ImageLocation = current\_path\_img;

}

}

}

}