



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра Прикладной информатики

БАКАЛАВРСКАЯ РАБОТА

На тему Информационная система тестового контроля знаний студентов
(на примере ГБПОУ КК «Туапсинский гидрометеорологический
техникум»)

Исполнитель Матрашилова Анастасия Николаевна
(фамилия, имя, отчество)

Руководитель
к.т.н., доцент, профессор кафедры «Прикладной информатики»
(ученая степень, ученое звание)

Попов Борис Николаевич
(фамилия, имя, отчество)

«К защите допускаю»
Заведующий кафедрой

(подпись)

кандидат технических наук
Слесарева Людмила Сергеевна

« 22 » 06 2016г.

Санкт-Петербург
2016



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра Прикладной информатики

БАКАЛАВРСКАЯ РАБОТА

На тему Информационная система тестового контроля знаний студентов
(на примере ГБПОУ КК «Туапсинский гидрометеорологический
техникум»)

Исполнитель Матрашилова Анастасия Николаевна
(фамилия, имя, отчество)

Руководитель
к.т.н., доцент, профессор кафедры «Прикладной информатики»
(ученая степень, ученое звание)

Попов Борис Николаевич
(фамилия, имя, отчество)

«К защите допускаю»
Заведующий кафедрой

(подпись)
кандидат технических наук
Слесарева Людмила Сергеевна

« ____ » _____ 2016г.

Санкт-Петербург
2016

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1. ОСНОВЫ ПРОВЕДЕНИЯ ТЕСТИРОВАНИЯ.....	7
1.1 Педагогический контроль и методы контроля знаний. Тестирование как форма контроля. Методика разработки тестов	7
1.2 Основные модели систем тестирования. Основные характеристики тестов.....	10
1.3 Теоретические аспекты разработки тестов. Классификация тестовых заданий	13
1.4 Анализ степени информатизации ГБПОУ КК ТГМТ и обоснование необходимости разработки системы автоматизированного тестирования.	19
2. ПРОЕКТИРОВАНИЕ ПОДСИСТЕМЫ КОНТРОЛЯ ЗНАНИЙ СТУДЕНТОВ	23
2.1 Анализ вариантов использования и потоки событий.....	23
2.2 Проектирование диаграммы классов	35
2.3 Проектирование базы данных.....	39
2.4 Проектирование графического пользовательского интерфейса	45
3. ОПИСАНИЕ ПРОГРАММНОГО ПРОДУКТА.....	49
3.1 Руководство пользователя.....	49
3.2 Руководство администратора.....	55
3.3 Результаты тестирования созданного программного продукта	55
4. ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ЭФФЕКТИВНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....	58
4.1 Оценка трудоемкости разработки программного продукта.....	58
4.2 Расчет показателей экономической эффективности внедрения программного продукта.....	63
ЗАКЛЮЧЕНИЕ.....	69
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	70
ПРИЛОЖЕНИЕ 1.....	72

ПРИЛОЖЕНИЕ 2.....	76
ПРИЛОЖЕНИЕ 3.....	77
ПРИЛОЖЕНИЕ 4.....	78

ВВЕДЕНИЕ

Одним из важнейших компонентов учебного процесса является систематический, хорошо организованный контроль качественного уровня знаний студентов. Формы проверки знаний могут быть самыми различными, например: устный опрос, контрольные работы, рефераты, курсовые работы, семинары. Перечисленные методы диагностирования успеваемости студентов имеют определенные недостатки: при проверке знаний большого числа студентов, наблюдается загруженность преподавателя работой, связанной с большим объемом информации, которую требуется подготовить, обработать, возможная небеспристрастность и списывание. Это искажает достоверность оценки знаний студентов и мешает преподавателю объективно оценивать качество своей педагогической работы. На современном этапе при оценке знаний студентов перечисленные проблемы в большей степени решаются использованием такой формы обучения и контроля, как тестирование. Этот метод позволяет измерять и интерпретировать результаты обучения с большой долей объективности, являясь оперативной, рациональной и удобной формой аттестации студентов.

Качество подготовки студентов во многом определяется качеством средств, используемых при обучении. В современной системе среднего образования возрастает роль информационных технологий, которые открывают новые возможности для повышения эффективности и качества подготовки студентов.

Актуальность работы связана с тем, что создание информационной образовательной среды системы образования предполагает наличие единых стандартов, форм, форматов данных, а также общей технологии взаимодействия всех структур и учреждений системы образования соответствующего уровня. Стратегической целью такого процесса является повышение эффективности функционирования структур и учреждений системы образования через внедрение ИКТ во все виды и формы образовательной

деятельности, трансформацию на этой основе существующих и формирование новых образовательных моделей. Вопросы проектирования информационной среды и организации управления стали повседневными вопросами менеджмента в любой сфере деятельности, не исключая образования.

Целью выпускной квалификационной работы является автоматизация процессов связанных с прохождением тестирования студентами.

Исходя из цели, можно выделить следующие задачи:

- создание тестов и тестовых вопросов различных типов;
- хранение в базе данных результатов тестирования студентов;
- автоматизация процесса прохождения тестирования;
- создание отчетов по результатам тестирования.

Пояснительная записка к выпускной квалификационной работе состоит из введения, четырех глав и заключения, изложенных на 66 страницах машинописного текста, списка использованных источников из 27 наименований, 4 приложений на 7 страницах, содержит 41 рисунок и 20 таблиц.

Во введении обоснована актуальность темы, сформулирована цель и задачи, которые необходимо решить в рамках данной выпускной квалификационной работы.

В первой главе рассмотрены и изучены основные понятия и виды тестирования, проведено исследование информационной системы ГБПОУ КК ТГМТ, приведены основания для разработки системы тестирования знаний.

Во второй главе выполнено проектирование ИС тестового контроля знаний студентов; составлена проектная документация, выполнена в соответствии с выбранным объектно-ориентированным подходом к проектированию декомпозиции системы на отдельные классы с применением case-средства Microsoft Visual Studio 2010 и языка UML; выполнено концептуальное, логическое и физическое проектирование БД, ее нормализация, создана логическая и физическая модель БД, выполнено проектирование графического интерфейса пользователя.

В третьей главе приведено руководство программиста, руководство

пользователя, а также приведены результаты тестирования программного обеспечения.

В четвертой главе приводится технико-экономическое обоснование эффективности разработанного ПО, а также оценка трудоемкости разработки программного комплекса.

1. ОСНОВЫ ПРОВЕДЕНИЯ ТЕСТИРОВАНИЯ

1.1 Педагогический контроль и методы контроля знаний.

Тестирование как форма контроля. Методика разработки тестов

Методы организации контроля знаний можно разделить на три класса:

- 1) неадаптивные методы;
- 2) частично адаптивные методы;
- 3) полностью адаптивные методы.

К неадаптивным методам контроля относятся:

Строгая последовательность. Набор заданий для контроля заранее подготавливается преподавателем или разработчиком контрольной работы. Как правило, это одинаковая последовательность вопросов для всех обучающихся. Недостатки данного метода очевидны: отсутствие разнообразия, понижение самостоятельности выполнения заданий и др. Этот метод считается наихудшим, поэтому и применяется крайне редко. Метод можно несколько улучшить, например, подготовив несколько вариантов контрольной работы и/или выдавая задания обучающимся в произвольной последовательности.

Случайная выборка. Набор заданий формируется непосредственно перед контролем на основе хранящихся заданий, т.е. вариант контрольной работы — это n случайно выбранных заданий. Преимущество данного метода состоит в том, что каждому обучающемуся предлагается индивидуальная последовательность вопросов.

Основным недостатком данного метода является то, что вариант контрольной работы генерируется без учёта трудности заданий. Таким образом, набор заданий для одного обучающегося может включать лишь самые трудные вопросы, а для другого — только лёгкие. Подобная ситуация может привести к искажению результатов контроля. Существуют различные модификации данного метода, позволяющие учитывать метаданные вопросов. Например, могут быть заданы тема и общее время контроля, время ответа на каждый вопрос, число попыток дать ответ и т.п.; дополнительно устанавливается число

вопросов разной степени трудности и/или из разных тем в каждом варианте контрольной работы.

Комбинированный метод, в основе которого лежит «случайная выборка», дополненная «строгой последовательностью». В этом случае преподаватель задаёт один или несколько вопросов, которые непременно должны быть включены в каждый вариант контрольной работы. Остальные задания генерируются случайным образом, как во втором методе.

Общим для всех неадаптивных методов является то, что вариант контрольной работы для каждого обучающегося формируется до времени контроля, что, с одной стороны, повышает скорость контроля (не требуется поиск задания), с другой — позволяет выдавать задания двумя способами: по одному или списком. В последнем случае обучающийся сам может выбрать последовательность выполнения заданий.

Достаточно очевидно, что наиболее простыми и универсальными методами контроля знаний являются неадаптивные методы. Частично адаптивные и особенно адаптивные методы требуют организации контроля при помощи ЭВМ. Также перед преподавателем возникает проблема определения трудности каждого вопроса. Вместе с этим при своей реализации они являются более эффективными по сравнению с неадаптивными методами, поскольку позволяют получить более точную оценку знаний испытуемого.

В настоящее время существует множество определений термина «тест». В самом общем виде тест можно определить как краткое стандартизированное испытание, допускающее количественную оценку результатов на основе их статистической обработки.

Под «испытанием» подразумевается система заданий, на основании анализа результатов выполнения которой предстоит провести количественную оценку измеряемым качествам и свойствам.

Каждое из заданий теста содержит описание некоторой «ситуации», взятой из природы, производства, педагогической деятельности и т.п. Это описание может быть представлено на различных «языках»: вербальном, языке

символов, графиков, рисунков и т.п. Любое описание всегда приблизительно, неполно, и поэтому перед испытуемым ставится «требование» сделать его ситуации более точными, полными, используя с этой целью «информацию к решению», содержащуюся в описании ситуации, тексте задания, а также привлекая «внешнюю по отношению к заданию информацию» — известные испытуемому научные факты, закономерности и т.п. В процессе выполнения задания ему приходится перемоделировать ситуацию путем введения ряда упрощающих решение допущений, абстракций, переходя в описании от одного языка к другому.

Среди заданий теста следует различать:

- задания информативного характера;
- задания, решение которых может быть осуществлено алгоритмическим, формализованным путем;
- проблемы, для решения которых необходим эвристический и не стандартизированный поиск.

Тест как «стандартизированное испытание» предполагает, что все выполняющие задания находятся в одинаковых, строго оговоренных условиях. Только это позволяет сравнивать результаты тестирования, доводить результаты измерения до числа.

Педагогический тест рассматривают как особую совокупность заданий, позволяющих дать объективную, сопоставимую и количественную оценку качества подготовки обучающегося в заданной области.

Условимся под педагогическим тестом понимать систему определенным образом расположенных заданий, позволяющих выявить структуру знаний, умений и измерить их.

Можно выделить основные характеристики педагогического теста, отличающие его от обычной контрольной работы:

- специфическое содержание и форма, определяемые целями тестирования;
- наличие эталона (ответа), т.е. правильного метода выполнения

каждого задания;

— указание метода обработки результатов (баллов) и критериев оценивания (шкала измерения);

— возможность оценки качества каждого тестового задания с помощью статистических методов обработки результатов тестирования;

— наличие инструкции по использованию широко применяемых тестов, в которой отражены цели тестирования, характеристика заданий и ожидаемых ответов, методика возможного использования и интерпретации тестовых результатов.

Результаты измерения заданий испытуемыми содержат самую разнообразную информацию в скрытой форме. Есть только один путь её извлечения — это сопоставление результатов выполнения достаточно большого числа заданий. Только это позволяет получить выводы, заслуживающие доверия.

1.2 Основные модели систем тестирования. Основные характеристики тестов

Педагогические тесты весьма разнообразны. По вопросу классификации тестов в литературе не существует единства. По характеру задач, решаемых с помощью тестов в процессе обучения, тесты могут подразделяться на входные, формирующие, диагностические и итоговые.

С точки зрения «уровня информации об объекте» можно различать, в частности:

1) *вербальные* тесты, которые проверяют знание обозначений, символов, знаков и т.п. Особое значение среди них имеет знание основных типов слов: синонимов, антонимов, паронимов, акронимов и т.п.

2) *концептуальные* тесты проверяют знание свойств понятий, отображающих объекты: случайные, конституирующие, фундаментальные, специфические и т.п.;

3) *интеллектуальные* тесты: противопоставление сущности и явления,

причин и следствий, рода и вида и т.п.

4) *идейные* тесты: содержание теорий по существу.

С точки зрения целей применения, можно выделить:

— тесты достижений;

— критериально-ориентированные тесты, позволяющие сопоставить уровень индивидуальных учебных достижений с полным объемом знаний, умений и навыков;

— нормативно-ориентированные тесты, сравнивающие испытуемых друг с другом по уровням и учебным достижениям;

— аттестационные тесты, определяющие степень обученности;

— тесты прогнозирования результатов обучения.

Как и любой педагогический тест, критериально-ориентированный тест представляет собой систему заданий, позволяющую измерить уровень учебных достижений. Но главная их особенность состоит в том, что критериально-ориентированные тесты сопоставляют уровень индивидуальных учебных достижений с полным объемом знаний, умений и навыков, которые должны быть усвоены обучающимися.

Обычно критериально-ориентированные тесты используются для разделения обучающихся на несколько групп по уровню их знаний. В самом простом случае это две группы — усвоившие и не усвоившие необходимый материал. В тесте образовательный стандарт выступает в виде критериального балла — минимального тестового балла, который должен получить студент, чтобы считаться усвоившим тот или иной материал. Для установления критериального балла используются экспертные методы анализа заданий.

Критериально-ориентированные тесты разрешают проблему соотношения тестовых баллов с традиционными педагогическими оценками. Устанавливаются такие баллы критериев, которые делят испытуемых на группы, соответствующие оценкам «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Такая организация теста позволяет проводить экзамены в тестовой форме.

Двумя главными характеристиками тестов являются *валидность* и *надёжность*.

Тест должен быть пригодным (*валидным*) с точки зрения поставленных перед ним целей измерения. *Валидность* означает соответствие формы и содержания теста тому, что он должен оценивать или измерять по замыслу его создателей. Задания теста должны соответствовать целям тестирования, содержанию проверяемого материала, и должна быть высока вероятность того, что испытуемый, успешно выполнивший тест, знает материал в соответствии с предъявляемыми требованиями. Таким образом, валидность свидетельствует о том, пригодна ли методика для измерения определённых параметров, и насколько эффективно она это делает.

Мера валидности теста определяется несколькими способами, например, сравнением с тестом-эталоном и сопоставлением с более объективными показателями, полученными другими методами.

Содержательная пригодность теста характеризуется определенными требованиями к его содержанию: все аспекты тестирования, отвечающие его целям, должны найти отражение в соответствующих заданиях, и наиболее значимым аспектам должно соответствовать большее количество заданий.

Содержательная пригодность педагогического теста может быть проверена либо с помощью экспертной оценки, либо сопоставлением оценок, полученных в результате выполнения заданий теста с оценками по той же теме, полученными другими методами (без использования тестов). Если при этом наблюдаются однотипные изменения общих результатов, то тест считается валидным (пригодным) с точки зрения поставленных перед ним целей измерения.

Надёжность теста характеризует степень устойчивости результатов тестирования каждого испытуемого. Фактически коэффициент надёжности показывает корреляционную связь между результатами измерений, проведенных в одинаковых условиях. При определении надёжности исходят из того, что в каждом измерении присутствуют истинный и искажающий

компоненты. Определение надёжности заключается в оценке соотношения этих компонентов в данных проведенного тестирования.

1.3 Теоретические аспекты разработки тестов. Классификация тестовых заданий

Для разработки и применения в учебном процессе рассмотренных выше педагогических тестов необходимо учитывать объективные характеристики подготовительного процесса. Для этого нужно:

- Выделить цели тестирования.
- Отобрать педагогические ситуации.
- Перевести (смоделировать) педагогические ситуации на язык учебных задач.

- Переконструировать учебные задачи в тестовые задания.
- Выбрать и оценить эталоны ответов.
- Разработать план теста.
- Выделить аспекты тестирования.
- Составить тест.
- Расположить задания в тесте с учетом системообразующих связей; следует отметить, что педагогический тест не может состоять из отдельных только лёгких или трудных заданий, он должен включать в себя задания различного уровня трудности, начиная с лёгкого и кончая трудным, на которые могут правильно ответить несколько испытуемых в тестируемой группе. С другой стороны, идея об обязательности расположения заданий по их уровню сложности в педагогическом тесте не является общепризнанной.

- Проверить экспериментально разработанный тест.

Процесс составления педагогических тестов является делом исключительно ответственным и трудоемким. Поскольку они являются основой контроля, диагностики в управлении процессом обучения, то становятся понятными исключительно высокие требования к содержанию и форме тестов.

Тестовое задание — это составная единица теста, отвечающая требованиям технологичности, формы, содержания и, кроме того, статистическим требованиям:

- 1) известной трудности;
- 2) достаточной вариации тестовых баллов;
- 3) положительной корреляции баллов задания с баллами по всему тесту.

Наличие достаточного числа тестовых заданий, полученных в результате обязательной эмпирической апробации заданий в тестовой форме, позволяет перейти к разработке теста как системы, обладающей целостностью, составом и структурой.

Требование известной трудности оказывается важнейшим системообразующим признаком тестового задания. Если придерживаться точки зрения, что тест — это система заданий возрастающей трудности, то в нем нет места заданиям с неизвестной мерой трудности.

Можно выделить следующие этапы конструирования теста:

- подготовительный этап, целью которого является чёткое, конкретное выделение целей тестирования;
- второй этап, связанный с подбором заданий для составления теста;
- третий этап, связанный с конструированием педагогического теста как логически и структурно оформленное целое, состоящее из заданий тестовой формы, находящихся в определенных связях между собой.

На подготовительном этапе, прежде всего, определяется назначение теста: для кого тест предназначен, на каком этапе обучения он будет использован, каким должен быть его уровень сложности. Иными словами, формулируются конкретные цели составления теста. При этом необходимо ориентироваться на конечный результат учебно-познавательной деятельности, соотнесенный с требованиями программы.

Заранее проектируется и необходимый уровень сложности теста, который предстоит составлять. По этому поводу можно сделать некоторые предварительные замечания. Сложность теста определяется пятью уровнями:

— *Информационный уровень*, требующий от испытуемого узнавания известной информации.

— *Репродуктивный уровень*, основными операциями которого являются воспроизведение информации, а также её преобразования алгоритмического характера.

— *Базовый уровень*, требующий от испытуемого понимания существенных сторон учебной информации, владения общими принципами поиска алгоритмов.

— *Повышенный уровень*, уровень сложности задания, требующий от испытуемого умения преобразовывать алгоритмы к условиям, отличающимся от стандартных условий, а также умение вести эвристический поиск.

— *Творческий уровень*, предполагающий наличие самостоятельного, критического оценивания учебной информации, умение решать нестандартные задания, владение элементами исследовательской деятельности.

Ответственной и сложной частью составления теста является подбор ответов к его заданиям. Прежде всего, эти ответы должны отображать наиболее распространенные ошибки обучающихся людей, что приобретает особое значение для постановки диагноза. Иными словами, подбор ответов должен помогать выявлению характера пробелов в знаниях испытуемых, давать достаточно обоснованную информацию о причинах их появления. Неправильные ответы, похожие на правильные ответы и потому правдоподобные, называются *дистракторами*. В идеале каждый дистрактор должен в равной мере использоваться всеми испытуемыми, выбирающими неправильный ответ. Дистрактор, который никто не выбирает в качестве правильного ответа, обычно называется неработающим и подлежит удалению для улучшения качества теста. Тестовое задание считается «хорошо работающим», если обучающиеся, дающие в основном правильные ответы, выполняют его правильно, а незнающие выбирают любой из ответов с равной вероятностью.

Поскольку в отдельных случаях не удастся подобрать все возможные

ответы испытуемого к заданию теста, то в их число включается вариант типа «ответ иной». Это может быть сделано и из иных соображений — дать возможность испытуемым самостоятельно вести поиск решений. При этом должна быть обеспечена возможность однозначно судить о правильности или ошибочности выводов испытуемого в случае его выбора ответа, отличающегося от ответов, предложенных в тесте. В то же время обосновано включение ответов типа: «ответить не представляется возможным».

Особенности постановки заданий, наличие ответов к ним несёт в себе информацию к решению, в определенной степени навязывает логику поиска решения, может ограничить в выборе способа решения. Всё это должно использоваться составителем теста оптимально и соответствовать целям тестирования.

В ответах к заданиям теста следует, как и при описании самих заданий, широко применять различные способы моделирования учебной информации; семантический и символический языки; языки рисунков, схем, графиков.

Согласно А. Г. Войтову, главная цель тестов — обеспечить систематическую учёбу обучающихся предъявлением им информации о достигнутом ими уровне знаний. Вместе с тем, тесты являются удобной формой оценки уровня знаний обучающихся преподавателем. Таким образом, тесты обеспечивают «единство прямых и обратных связей» между преподавателями и обучающимися, выступая «наиболее развитой формой контроля».

Гарантия контрольной функции тестов должна исходить из следующего положения, ведущего к составлению программ самоконтроля. Контроль никогда не бывает полным, всеобъемлющим. Он базируется на проверке части результатов процесса обучения. Выборочный характер контроля повышает для обучающихся значение контрольных программ. Ознакомление обучающихся людей с содержанием контролируемого приводит к более глубокому изучению ими данного раздела курса. Это имеет как положительный, так и отрицательный эффект. Отрицательный эффект состоит в «натаскивании на тест» (вместо изучения всего содержания курса): обучающийся зазубривает

программу контроля. Недопущение этого возможно различными методами. Эффективен метод доведения контрольных программ по курсу до максимально возможного их состояния. В таком случае обучающийся человек практически не сможет заучить путём зубрёжки всю контрольную программу из-за её большого объёма. Поскольку любая часть такой программы может быть включена в тест, то обучающийся должен будет понимать её всю.

По принятой в литературе классификации выделяют следующие типы тестовых заданий:

- 1) задания закрытой формы с множественным выбором — учащиеся выбирают правильный ответ из данного набора ответов;
- 2) задания на дополнения (открытые задания), требующие при выполнении самостоятельного получения ответов;
- 3) задания на установление соответствия, выполнение которых связано с выявлением соответствия между элементами двух множеств;
- 4) задания на установление правильной последовательности, в которых требуется указать порядок перечисленных действий или процессов.

Основные преимущества *закрытых заданий* связаны с быстротой тестирования и простотой подсчета итоговых баллов. В них можно выделить основную часть, содержащую постановку проблемы, и готовые ответы, сформулированные преподавателем.

Ответы к заданиям должны быть представлены в форме, удобной для их сравнения, и изложены лаконично. Оптимальным числом их к заданию следует признать 4—5. Нежелательно количество ответов менее 4-х и более 6-и.

Выбор формы задания во многом определяется спецификой содержания учебной дисциплины, целями создания и применения теста. Немало здесь зависит от технологии проверки, сбора и обработки эмпирических данных, от технического и материального обеспечения процесса применения теста.

Легко организовать компьютеризованный сбор и анализ результатов выполнения теста в том случае, когда все задания имеют закрытую форму. Результаты выполнения заданий на дополнение с ограниченным, а тем более с

открытым ответом требуют ручной обработки. Как правило, для оценки результатов их выполнения приходится привлекать экспертов, а это требует дополнительных материальных затрат и времени на проверку.

Конечно, в каждом тесте можно использовать несколько форм, однако желательно, чтобы этих форм было как можно меньше. Более того, при создании теста имеет смысл выбрать одну наиболее подходящую форму и выдерживать ее от первого до последнего задания. Требование единства формы соблюдается, как правило, в профессионально сделанных тестах.

Обилие форм в тесте затрудняет работу обучающегося и значительно усложняет статистическую обработку эмпирических результатов выполнения теста. К сожалению, требование моноформности не всегда выполнимо, поскольку не все знания и умения ученика можно проверить с помощью моноформного теста. Поэтому часто приходится идти на совмещение форм, что при прочих равных условиях всегда негативно отражается на точности измерений, обеспечиваемой тестом.

При создании теста правильный выбор формы не менее важен, чем продуманный отбор содержания теста. Если форма выбрана неудачно, то в некоторых заданиях обучающийся может найти не предусмотренный преподавателем более легкий путь решения поставленной задачи. Ошибки такого рода устранить наиболее трудно, поскольку даже после многократного использования теста, в группе обучающиеся вряд ли станут сообщать об этих ошибках преподавателю и, конечно, воспользуются недосмотром разработчика теста. Для отдельных заданий вопрос правильного выбора формы приобретает особую сложность, так как часто случается, что ни одна из форм не является в полной мере удачной для создаваемого задания теста.

1.4 Анализ степени информатизации ГБПОУ КК ТГМТ и обоснование необходимости разработки системы автоматизированного тестирования

Информационная система ТГМТ включает в себя локальную вычислительную сеть, построенную по технологии Ethernet. В качестве сетевой среды используются витая пара категории 5е и Wi-Fi. Уровень доступа обеспечивается коммутаторами и точками беспроводного доступа производства компании D-Link. На уровне распределения между техникумом и сетью Интернет находится маршрутизатор с точкой беспроводного доступа производства компании D-Link. В качестве сервера домена используется сервер производства фирмы IBM в следующей конфигурации:

- 2 центральных процессорных устройства Xeon;
- 16 ГБ ОЗУ, 2 жёстких диска с интерфейсом SAS ёмкостью по 250 гигабайт в конфигурации RAID-1 (зеркальное отображение информации на обоих дисках для повышения надёжности хранения данных);
- надёжное охлаждение и мощный блок питания, возможность горячей замены устройств (Hot Swap).

В качестве операционной системы сервера используется лицензионная сетевая серверная операционная система производства фирмы Microsoft — Windows 2008 R2. На сервере установлено программное обеспечение для защиты от несанкционированных вторжений в локальную сеть организации (сетевой экран). Для автоматизации учебного процесса используется программное обеспечение производства российской фирмы 1С — 1С:Колледж ПРОФ. Рабочие станции организации построены на оборудовании класса не ниже центральных процессоров Intel Pentium 4 или аналогичных им от фирмы AMD, объём ОЗУ не ниже 512 МБ, ёмкость накопителей на жёстких магнитных дисках не ниже 250 ГБ. В качестве программного обеспечения электронного документооборота используется локализованный лицензионный пакет прикладных программ производства фирмы Microsoft — Office 2007 Professional Plus.

По степени информатизации образовательного процесса ТГМТ удовлетворяет большинству современных требований, но все же некоторые компоненты информационной системы могут быть дополнительно разработаны, в частности информационная система тестового контроля знаний. В связи с этим создание такого компонента является актуальной задачей.

Большинство преподавателей ТГМТ положительно относятся к применению тестирования в учебном процессе. Они готовы использовать этот метод и считают его перспективным для оценки знаний студентов. Однако, тестовый контроль по мнению большинства преподавателей, не способен объективно и в полном объеме оценить знания студентов, а также дифференцировать их по этому показателю. Это свидетельствует о низкой осведомленности преподавательского состава в вопросах теории тестового контроля. Большинство преподавателей мало знакомы с компьютерными тестирующими программами и их возможностями.

В деятельности ТГМТ используются тестовые программы на основе MSExcel, MSWord, HTML и PHP, созданных самими преподавателями или привлеченными для этого студентами.

Недостатки такого тестового контроля следующие:

- затрата времени на изготовление пакета тестов по предмету очень большие;
- устаревание тестовой базы;
- возможность использования посторонних программ для помощи в ответах;
- плохой контроль результатов тестирования
- при бланковом тестировании тратится много бумаги;
- списывание.

Можно указать следующие главные требования к программам контроля знаний студентов, работающими с заданиями в тестовой форме.

- программа должна решать все задачи контроля знаний студентов в аудитории, при самоподготовке и работе с электронными учебными

материалами, при очной, вечерней, заочной или дистанционной формах обучения. Необходимо, чтобы все задания имели одинаковый формат и были на 100% совместимы. В этом случае они могут повторно использоваться целиком или в виде выборок из базы тестовых заданий.

— программы тестового контроля знаний должны использовать все основные типы заданий в тестовой форме: одиночный и множественный выбор, напоследовательность и соответствие, задания в открытой форме, а также давать возможность широко использовать компьютерную графику. Формат тестовых заданий должен быть универсальным, не связанным со спецификой конкретных дисциплин.

— результаты тестирования должны представляться как в числовой, так и в текстовой формах, с возможностью интерпретации результатов тестирования. Протоколы результатов должны быть легко экспортируемы в такие форматы, как MSExcel, MSAccess для сопряжения с другими программами. Программы должны иметь ряд режимов, необходимых при тестовом контроле знаний: таймер на каждое задание или сеанс тестирования целиком, перестановка вариантов ответов, шифрование заданий для исключения возможности несанкционированного доступа, режимы отображения результатов тестирования в текстовом виде и в виде методической графики, по которой можно наглядно оценить сложность тестовых заданий, уровень подготовки группы и др.

В связи с тем, что решение о внедрении централизованной системы тестирования знаний ещё не принято, а необходимость проведения таких тестирований существует, было решено разработать систему автоматизированного тестирования, которую могут использовать отдельные преподаватели, а при необходимости возможно создать централизованную систему, разместив её базу данных на сервере сети ТГМТ. Данная система решает две проблемы:

— экономия средств в условиях экономического кризиса;

— возможность произвольной адаптации системы под требования техникума силами специалистов самого техникума, обусловленная наличием исходных текстов программ.

2. ПРОЕКТИРОВАНИЕ ПОДСИСТЕМЫ КОНТРОЛЯ ЗНАНИЙ СТУДЕНТОВ

2.1 Анализ вариантов использования и потоки событий

Используя объектно-ориентированный подход и язык UML выполним проектирование диаграммы вариантов использования ИС тестового контроля знаний студентов (представлена на рисунке 2.1).

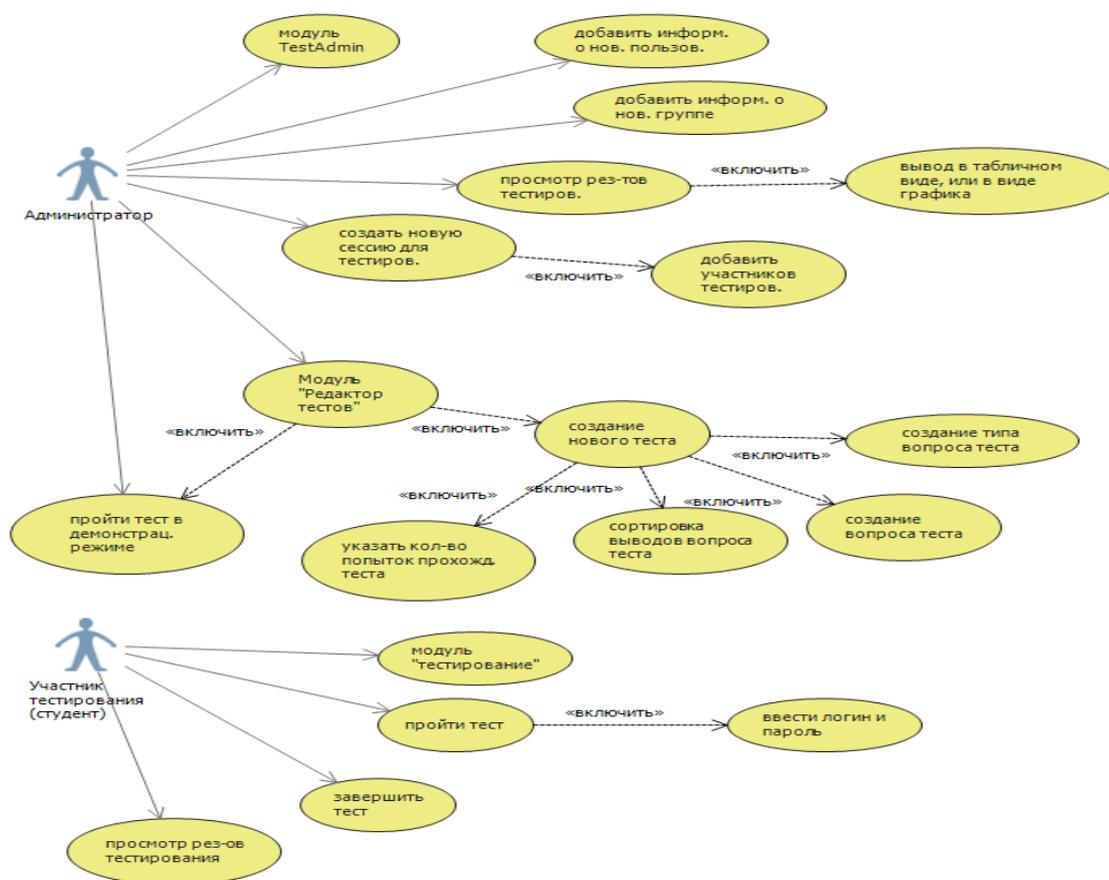


Рисунок 2.1 Диаграмма вариантов использования проектируемой ИС тестового контроля знаний студентов

При проектировании диаграммы вариантов использования, диаграмм последовательности действий, диаграммы классов и при разработке непосредственно программного продукта будем использовать инструментальную среду разработки Microsoft Visual Studio 2012.

Эффективная разработка интеллектуальных клиентов начинается с

качественного проектирования приложений. Visual Studio облегчает вовлечение дизайнеров пользовательских интерфейсов в процесс разработки за счет поддержки стандартных форматов (таких, как XAML) и предоставления им возможности непосредственно контролировать внешний вид, элементы управления и связывание данных с интерфейсами приложений. При этом они могут создавать интерфейсы с использованием знакомых инструментов, например Microsoft Expression Suite, а генерируемые ими файлы разработчики будут открывать прямо в Visual Studio.

На рисунке 2.1 изображены следующие действующие лица и варианты использования:

а) действующие лица:

– администратор (может создавать новые тесты, вопросы для тестов, выбирать типы вопросов, указывать сложность вопросов, выбирать порядок сортировки вопросов в тесте, указывать количество попыток для тестирования, создавать новых пользователей (участников тестирования), объединять участников тестирования в группы, назначать время прохождения тестирования);

– участник тестирования (студент) (может пройти тестирование, и просмотреть свои результаты);

б) варианты использования:

– «модуль TestAdmin» (модуль программы для создания новых пользователей, объединения их в группы и задания сессии прохождения тестов);

– «добавить информацию о новом пользователе» (вариант использования позволяет добавлять информацию о новом участнике тестирования);

– «добавить информацию о новой группе» (вариант использования позволяет создать новую группу, отредактировать существующую, удалить группу, добавить в группу участников тестирования);

– «просмотр результатов тестирования» (вариант использования позволяет просмотреть результаты тестирования тех студентов, которые

прошли данное тестирование);

– «вывод в табличном виде, или в виде графика» (вариант использования позволяет просмотреть результаты тестирования тех студентов, которые прошли данное тестирование соответственно в табличном виде, или в виде графика);

– «создать новую сессию для тестирования» (вариант использования позволяет создать новую сессию для тестирования, т.е. указать какой тест, и в какое время доступен);

– «добавить участников тестирования» (вариант использования позволяет добавлять участников тестирования, которым будет доступен определенный тест в определенное время);

– «модуль тестирование» (вариант использования позволяет пройти тестирование и просмотреть результаты тем участникам, для кого этот тест предназначен);

– «просмотр результатов тестирования»;

– «завершить тест»;

– «пройти тест» (вариант использования позволяет пройти тест, где можно указать логин и пароль для тех участников, для которых они были назначены);

– «ввести логин и пароль»;

– «модуль редактор тестов» (вариант использования позволяет создавать новый тест, создавать новые вопросы для текущего теста, выбирать типы сложности вопросов, указывать количество попыток для прохождения теста, пройти тест в демонстрационном режиме);

– «создание нового теста»;

– «пройти тест в демонстрационном режиме»;

– «указать количество попыток прохождения теста»;

– «сортировка выводов вопроса теста»;

– «создание вопроса теста»;

– «создание типа вопроса теста».

2.1.1 Анализ варианта использования «Модуль TestAdmin»

Краткое описание

Модуль программы для создания новых пользователей, объединения их в группы и задания сессии прохождения тестов.

Предусловия

Перед началом выполнения данного варианта использования пользователь должен запустить соответствующую программу.

Основной поток событий

1) Администратор должен создать новую группу, для этого ему надо будет выбрать в раскрывающемся списке «администрирование» – «пользователи и группы» – «группы».

2) Далее необходимо нажать кнопку «добавить».

3) В появившемся диалоговом окне необходимо ввести название группы и нажать кнопку «ОК», в случае если ранее были созданы пользователи, то их можно перетащить во вновь созданную группу, но при этом они не должны быть добавлены в других группах, т.к. один пользователь не может быть участником нескольких групп одновременно.

4) Далее администратор может создать нового пользователя, для этого ему необходимо выбрать в раскрывающемся списке «администрирование» – «пользователи и группы» – «пользователи».

5) После нажатия кнопки «добавить» в диалоговом окне администратор вводит:

а) Основные сведения о пользователе: фамилия, имя, отчество, логин, также может ввести пароль (если пароль не вводить, то участник группы будет иметь доступ к тестирующему модулю без пароля), также можно выбрать группу, к которой будет прикреплен вновь создаваемый участник тестирования.

б) Вкладка документ содержит сведения о документе, который будет предоставляться участником тестирования в случае необходимости, заполнять эту вкладку необязательно; поля для заполнения здесь: тип документа, серия и номер, кем выдан, и когда выдан.

в) Следующая вкладка, также не обязательна для заполнения – носит название «контакты», в ней следующие поля: адрес, телефон и электронная почта.

г) Вкладка «дополнительно» служит для заполнения какой-то дополнительной информации об участнике тестирования при необходимости.

После заполнения всех необходимых полей администратор должен нажать кнопку «ОК».

б) После того как новый участник тестирования создан, администратор может отредактировать о нем информацию, для этого необходимо выделить созданного участника тестирования в списке участников и нажать кнопку «редактировать», после чего будет доступно диалоговое окно с полями аналогичными тем, что перечислены выше в пункте 5.

7) Администратор может удалить любого созданного пользователя, для этого ему необходимо выбрать участника тестирования в списке и нажать кнопку «удалить».

8) Далее администратор может создать новую сессию для тестирования, для этого ему необходимо выбрать в раскрывающемся списке «администрирование» пункт «тестирование».

9) Администратор должен нажать кнопку «добавить» для создания новой сессии для тестирования, после чего на экране должно появиться диалоговое окно «параметры теста».

10) Администратор должен выбрать заранее созданный тест, он это может сделать путем нажатия кнопки «из файла», а может путем нажатия кнопки «из базы данных». Если заранее созданных тестов нет, то их вначале необходимо создать.

11) Если администратор открыл созданный заранее тест, то он может выбрать время, когда тест будет доступен для прохождения, соответственно с какого по какое время.

12) Далее необходимо выбрать участников тестирования, которым будет доступен этот тест.

13) Далее необходимо нажать кнопку «ОК», данная кнопка будет неактивна, если не выбран сам тест, или не один из участников тестирования не выбран для прохождения данного теста.

14) Администратор может отредактировать созданный тест, для этого ему необходимо выбрать этот тест из списка и нажать кнопку «редактировать», после чего выполнить аналогичные действия, описанные в пунктах 8-13.

15) Администратор может удалить созданный тест, для этого ему необходимо нажать на кнопку «удалить».

16) Администратор может просмотреть результаты прохождения теста, в том случае если соответствующий тест был создан, и кто-то из участников тестирования уже прошел данный тест.

17) Администратор может выбрать или всех участников тестирования, которые проходили данный тест, или выбрать определенную группу, к которой привязаны соответственно участники.

На рисунке 2.2 изображена диаграмма последовательности действий для варианта использования «TestAdmin».

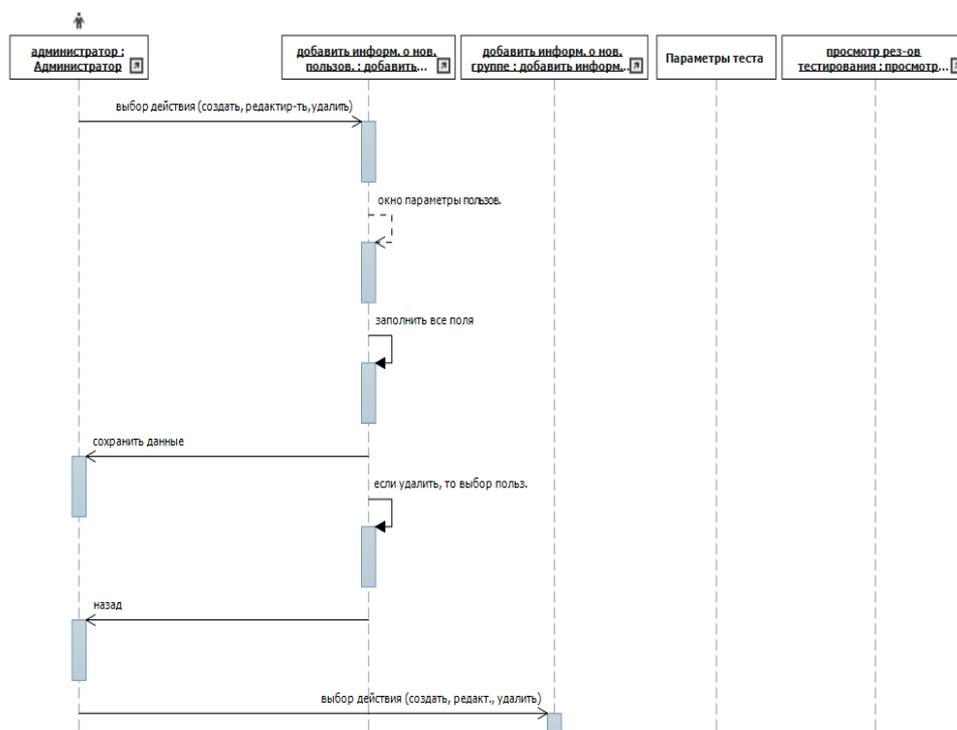


Рисунок 2.2 Диаграмма последовательности действий для «TestAdmin»

На рисунке 2.3 и рисунке 2.4 изображено продолжение диаграммы вариантов использования для «TestAdmin».

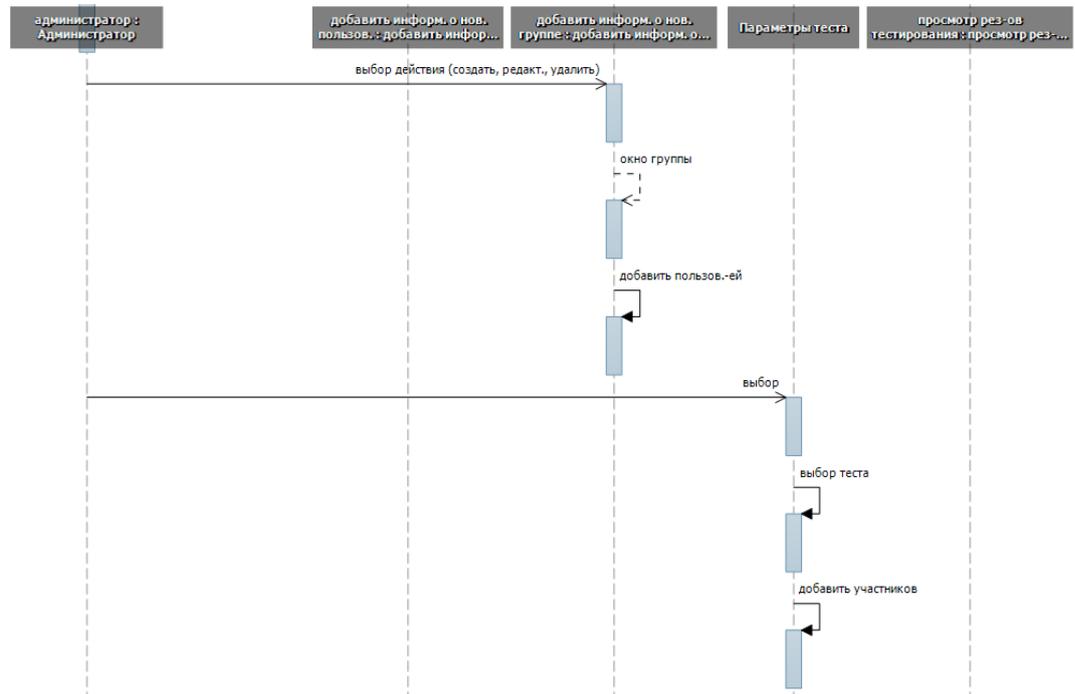


Рисунок 2.3 Продолжение диаграммы последовательности действий для «TestAdmin»

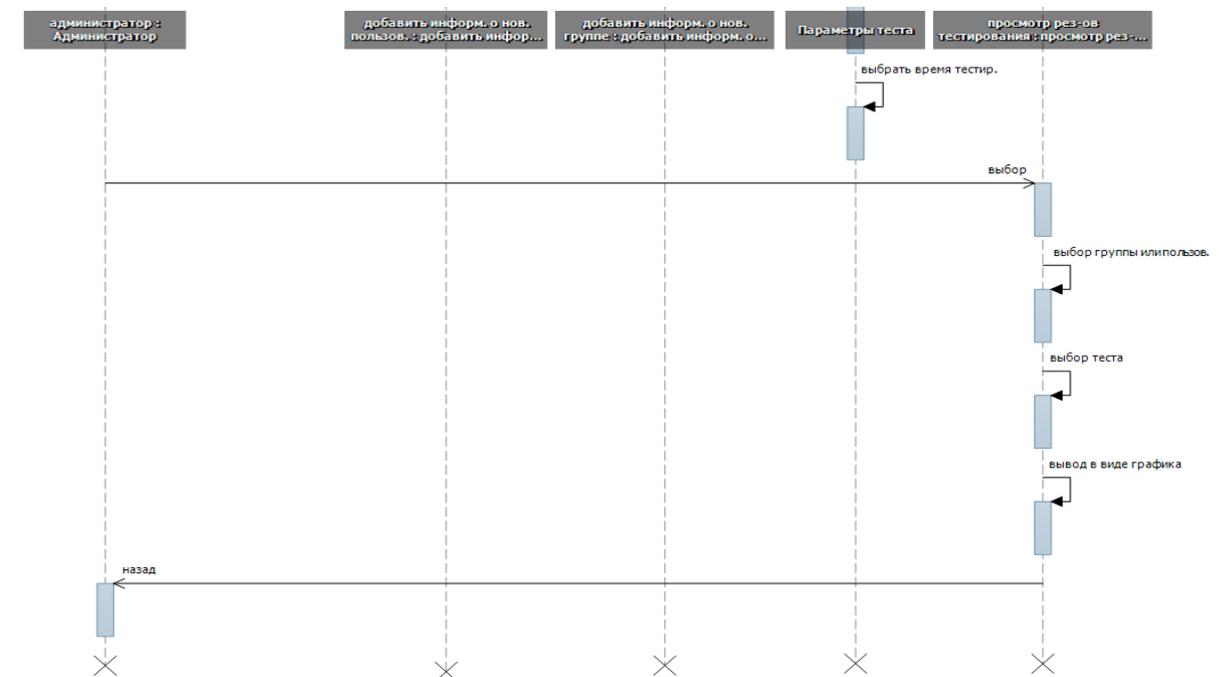


Рисунок 2.4 Продолжение диаграммы последовательности действий для «TestAdmin»

2.1.2 Анализ варианта использования «Модуль Тестирование»

Краткое описание

Модуль программы для прохождения теста участниками тестирования.

Предусловия

Перед началом выполнения данного варианта использования пользователь должен запустить соответствующую программу.

Основной поток событий

1) Участник тестирования может выбрать группу из списка, к которой он принадлежит, он может также выбрать из списка всех участников.

2) На вкладке «тестирование» участник тестирования (студент) выбирает тот тест, который ему доступен.

3) Если администратором был установлен для данного участника пароль, то студент должен ввести этот пароль.

4) После этих действий участник тестирования должен нажать кнопку «Тест».

5) В диалоговом окне выполнения теста участник тестирования должен ответить на вопросы теста различной сложности, студент может пропустить какой-то вопрос теста, или завершить досрочно тестирование, но в этом случае на те вопросы, на которые он не ответит будут засчитаны, как неверные.

6) После прохождения тестирования студент может просмотреть результаты тестирования, также может повторно пройти тест, если это задано администратором (количество попыток для тестирования устанавливается администратором).

Альтернативные потоки

Ввод некорректной информации. Если при выполнении Основного потока обнаружится, что пользователь ввёл некорректную информацию, система выводит сообщение об ошибке. Пользователь может вернуться к началу Основного потока или отказаться от ввода информации, при этом выполнение варианта использования завершается.

На рисунке 2.5 изображена диаграмма последовательности действий для

варианта использования «Модуль Тестирование».

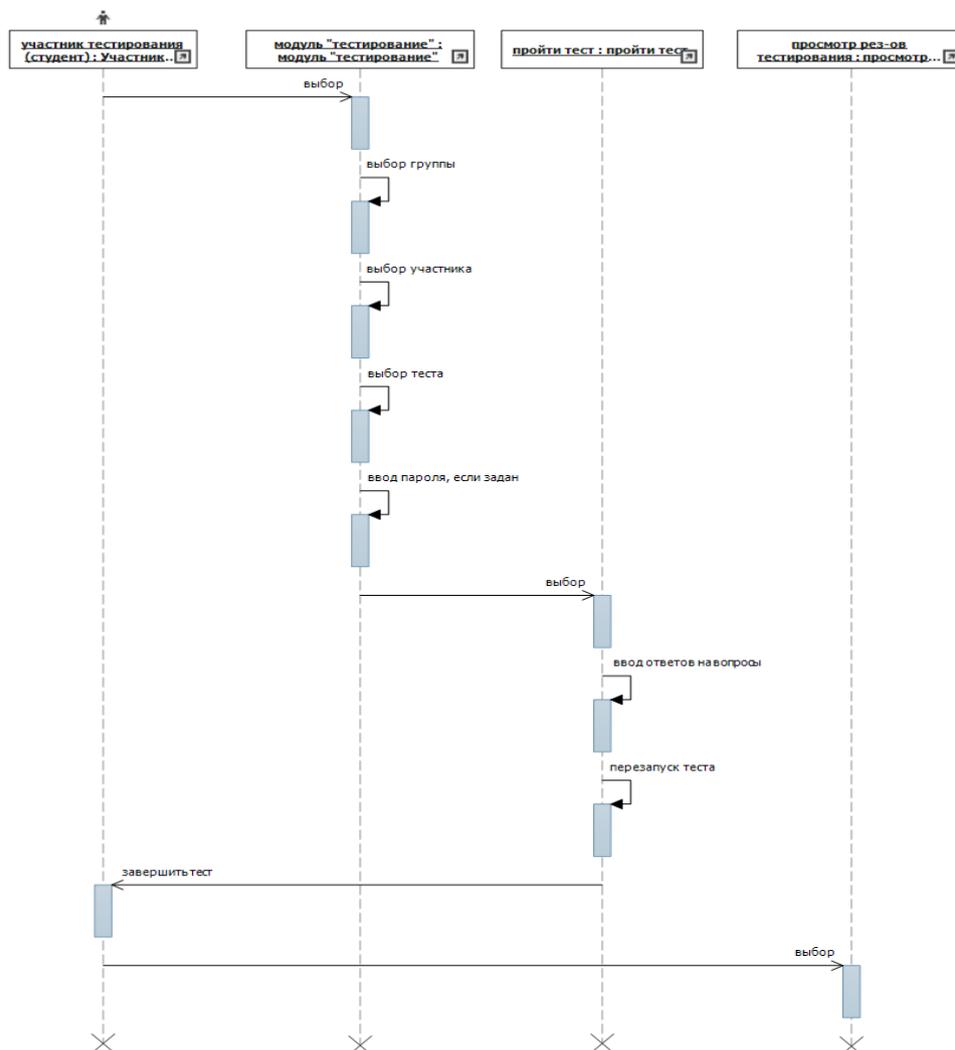


Рисунок 2.5 Диаграмма последовательности действий для варианта использования «Модуль Тестирование»

2.1.3 Анализ варианта использования «Модуль редактор тестов»

Краткое описание

Модуль программы «редактор тестов» позволяет создавать новый тест, создавать новые вопросы для текущего теста, выбирать типы сложности вопросов, указывать количество попыток для прохождения теста, пройти тест в демонстрационном режиме.

Предусловия

Перед началом выполнения данного варианта использования пользователь должен запустить соответствующую программу.

Основной поток событий

- 1) Администратор должен создать новый тест и ввести его название.
- 2) Также администратор может открыть уже существующий тест из базы данных, или из файла.
- 3) Администратор может добавить вопрос к тесту, изменить существующий вопрос, удалить вопрос и пройти тест в демонстрационном режиме.
- 4) Если администратор выберет пункт «добавить вопрос», тогда ему необходимо:
 - а) указать тип вопроса теста (единственный выбор ответа, множественный выбор ответа, ввод текстового ответа, ввод числового ответа, сопоставление);
 - б) выбрать режим вывода ответов (по порядку, в обратном порядке, случайным образом);
 - в) указать сложность вопроса (можно не задавать, а можно выбрать: легкий, средний, сложный);
 - г) ввести текст вопроса;
 - д) добавить ответы на вопрос теста и указать количество баллов за них.
- 5) Администратор может ограничить количество попыток для прохождения теста.
- 6) Администратор может указать способ вывода вопросов теста (случайным образом, по порядку, в обратном порядке).
- 7) Администратор может в «секциях сложности вопроса» указать к какой категории может быть отнесен вопрос (любой уровень, легкий уровень, средний уровень, сложный уровень), для этого ему необходимо выбрать любой вопрос из теста, и уже для него произвести изменения.
- 8) После того, как администратор создал новый вопрос теста, он может сохранить весь тест в базе данных подсистемы контроля знаний, или в файле.
- 9) Для того, чтобы создать новый вопрос, или изменить уже существующий необходимо пройти шаги с 4 по 7-ой.

10) Соответственно для создания нового теста, необходимо повторить все шаги, начиная с первого пункта данного потока событий.

Альтернативные потоки

Ввод некорректной информации. Если при выполнении Основного потока обнаружится, что пользователь ввёл некорректную информацию, система выводит сообщение об ошибке. Пользователь может вернуться к началу Основного потока или отказаться от ввода информации, при этом выполнение варианта использования завершается.

На рисунке 2.6 изображена диаграмма последовательности действий для варианта использования «Модуль редактор тестов».

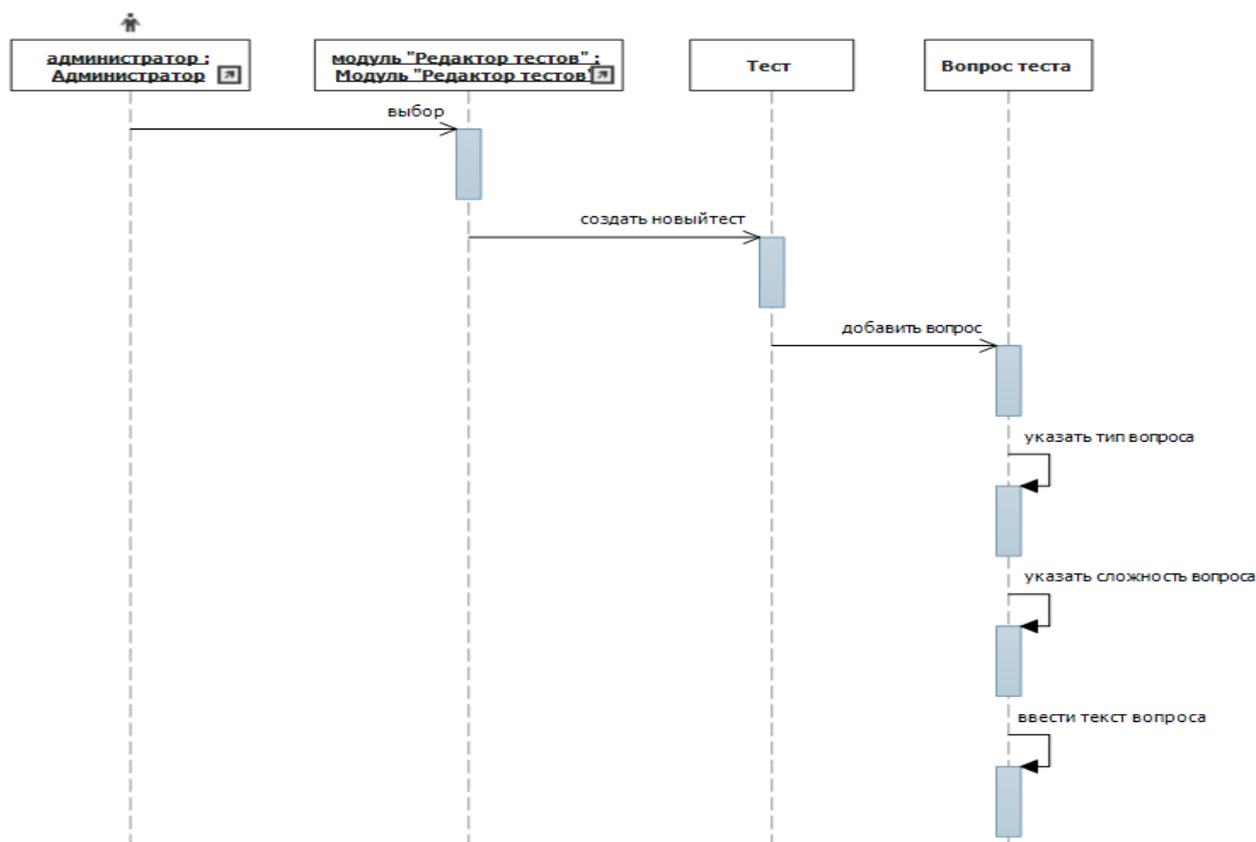


Рисунок 2.6 Диаграмма последовательности действий для варианта использования «Модуль Редактор Тестов»

На рисунке 2.7 изображено продолжение диаграммы последовательности действий для варианта использования «Модуль Редактор Тестов».

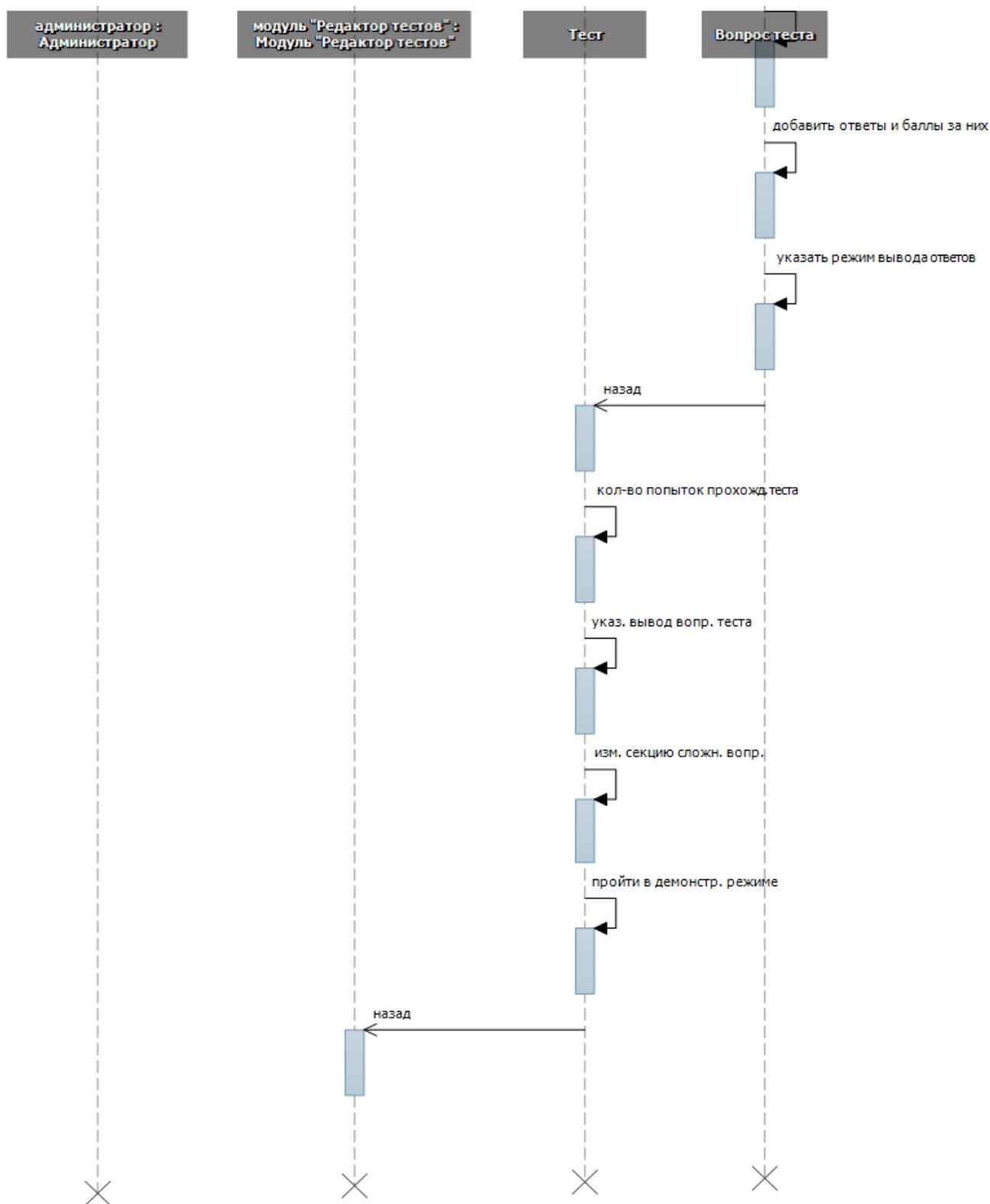


Рисунок 2.7 Продолжение диаграммы последовательности действий для варианта использования «Модуль Редактор Тестов»

2.2 Проектирование диаграммы классов

В приложении 2 изображена схема классов для модуля «TestAdmin», где изображены следующие классы:

1) CGroup – класс для создания группы, ее удаления, для ее переименования и сохранения, ниже на рисунке 2.8 приведен полностью.

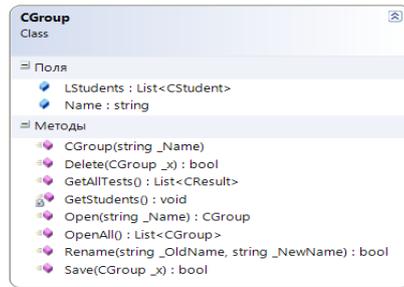


Рисунок 2.8 Класс CGroup

2) COptions – класс для хранения настроек подключения к базе данных (представлен на рисунке 2.9).



Рисунок 2.9 Класс COptions

3) CPassing – класс для хранения и установки времени проведения тестирования (представлен на рисунке 2.10).

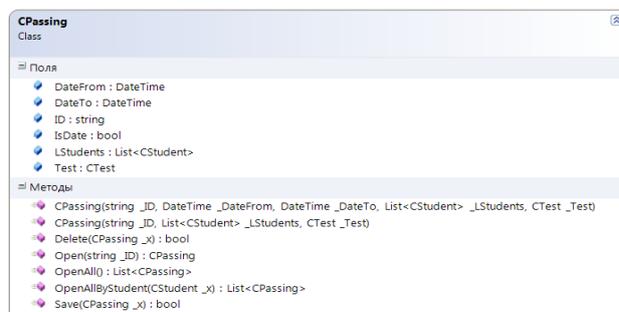


Рисунок 2.10 Класс CPassing

4) CResult – класс для хранения результатов прохождения теста (представлен на рисунке 2.11).

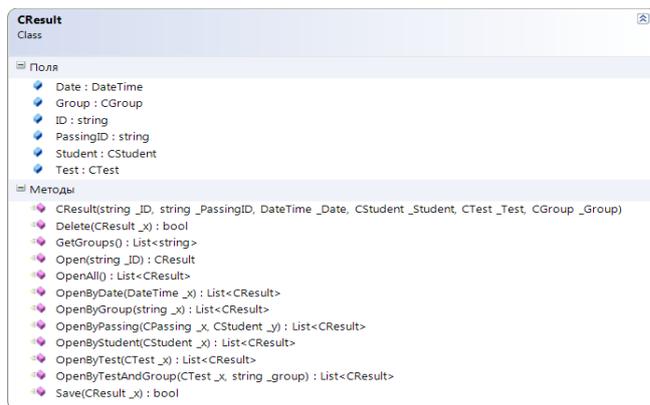


Рисунок 2.11 Класс CResult

5) CQuestion – класс для хранения и работы с вопросами теста (представлен на рисунке 2.12).

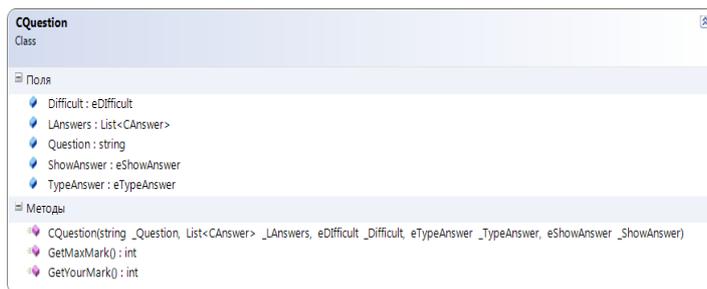


Рисунок 2.12 Класс CQuestion

6) CTest – класс для работы с тестом (для указания названия теста, подсчета баллов за тест и т.д., представлен на рисунке 2.13).

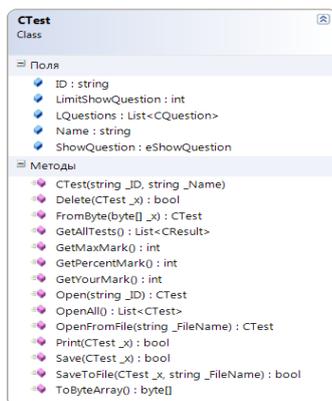


Рисунок 2.13 Класс CTest

7) CStudent – класс для работы с данными участников тестирования (представлен на рисунке 2.14).

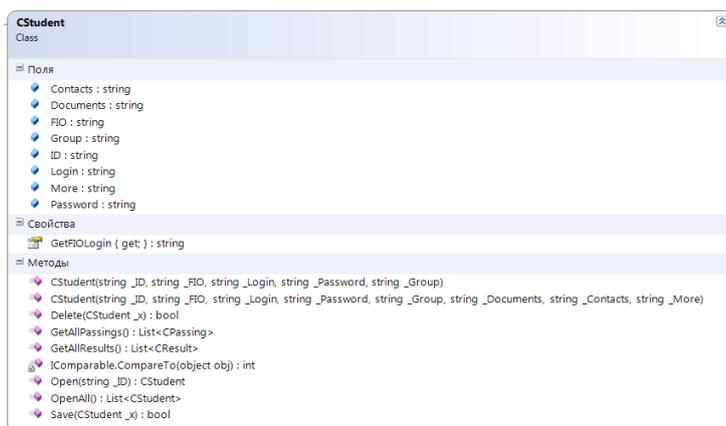


Рисунок 2.14 Класс CStudent

8) CAnswer – класс для хранения результатов ответа на вопросы теста (представлен на рисунке 2.15).

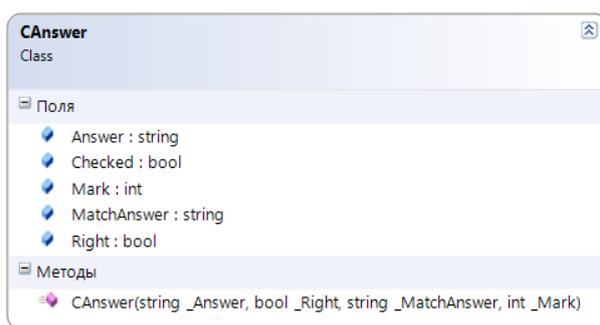


Рисунок 2.15 Класс CAnswer

Остальные классы, изображенные на диаграмме классов, создаются автоматически, т.к. описывают формы программы.

В приложении 3 изображена диаграмма классов модуля «Тестирование», где изображены следующие классы:

1) Класс CQuestion – класс для чтения вопросов из теста и подсчета результата ответа участника тестирования.

2) Класс CResult – класс для вывода результатов прохождения тестирования любым из участников.

3) Класс CStudent – класс для получения результатов прохождения

тестирования отдельным студентом (изображен на рисунке 2.16) .

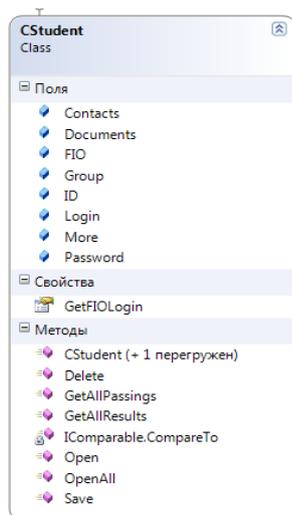


Рисунок 2.16 Класс CStudent

4) Класс CSettings – класс хранения настроек подключения к базе данных (изображен на рисунке 2.17).

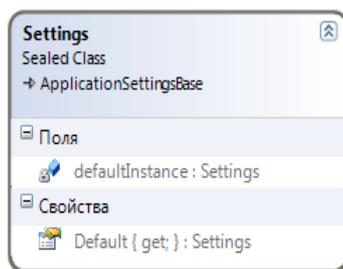


Рисунок 2.17 Класс Settings

Остальные классы модуля «Тестирование» не приведены на диаграмме классов, т.к. создаются автоматически и описывают визуальные элементы и методы диалоговых форм программы.

В приложении 4 изображена диаграмма классов модуля «Редактор тестов», где изображены выборочно некоторые классы основных форм модуля «Редактор Тестов»:

- frmAnswerMatch (класс для добавления ответа теста на сопоставление);
- frmDataBaseConnect (класс для подключения к базе данных при первоначальном запуске программы);
- frmTestFromDB (класс для открытия теста из базы данных);

- frmSection (класс для указания к какой секции отнести вопрос теста: легкий, сложный и т.д.);
- frmNewQuestion (класс для создания нового вопроса теста);
- frmTextNumAnswer (класс для создания открытого (текстового) ответа на вопрос теста);
- frmTestEditor (класс, описывающий главную форму модуля «Редактор Тестов»).

2.3 Проектирование базы данных

2.3.1 Этап концептуального проектирования

На данном этапе необходимо определиться с той информацией, которая будет храниться в базе данных проектируемой подсистемы контроля знаний студентов.

Необходимо хранить данные об участниках тестирования – студентах (Ф.И.О., логин, пароль, название группы, к которой прикреплен студент, документ удостоверяющий личность участника тестирования, контактные данные, дополнительную информацию). Также необходимо хранить вопросы теста и названия тестов, баллы за каждый правильный ответ, общую сумму баллов, результаты тестов, даты начала и окончания прохождения тестирования, промежуточные результаты тестирования. После того как общая информация, которая должна храниться в базе данных определена, можно приступить к следующей фазе логического проектирования.

2.3.2 Этап логического проектирования

На этапе логического проектирования проводят нормализацию базы данных с целью устранения избыточности данных, а также с целью оптимизации хранимых данных.

Нормальная форма – свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, которая потенциально может привести к логически ошибочным результатам выборки или изменения данных. Нормальная форма определяется как совокупность требований,

которым должно удовлетворять отношение.

Процесс преобразования базы данных к виду, отвечающему нормальным формам, называется нормализацией. Нормализация предназначена для приведения структуры базы данных к виду, обеспечивающему минимальную избыточность, то есть нормализация не имеет целью уменьшение или увеличение производительности работы или же уменьшение или увеличение объёма БД. Конечной целью нормализации является уменьшение потенциальной противоречивости хранимой в БД информации.

Устранение избыточности производится, как правило, за счёт декомпозиции отношений таким образом, чтобы в каждом отношении хранились только первичные факты (то есть факты, не выводимые из других хранимых фактов).

Отношение находится в первой нормальной форме тогда и только тогда, когда в любом допустимом значении отношения каждый его кортеж содержит только одно значение для каждого из атрибутов.

В реляционной модели отношение всегда находится в первой нормальной форме по определению понятия отношение. Что же касается таблиц в существующих реляционных СУБД (SQL-СУБД), то они могут не быть правильными отношениями и, соответственно, не находиться в первой нормальной форме.

В таблицах 2.1-2.4 представлены таблицы удовлетворяющие требованиям первой нормальной формы.

Таблица 2.1 – Данные об участниках тестирования (студентах)

FIO	Login	Password	Group	Document	Contacts	More
-----	-------	----------	-------	----------	----------	------

Таблица 2.2 – Данные о результатах прохождения тестирования

PassingID	Date	StudentID	Group	TestBinary	TestID
-----------	------	-----------	-------	------------	--------

Таблица 2.3 –Таблица для хранения самих тестов

TestBinary

Таблица 2.4 – Данные о времени прохождения тестирования

IsDate	DateFrom	DateTo	LStudentsID	TestBinary
--------	----------	--------	-------------	------------

Отношение находится во второй нормальной форме, если оно находится в первой нормальной форме, и при этом любой его атрибут, не входящий в состав потенциального ключа, функционально полно зависит от каждого возможного ключа. Функционально полная зависимость означает, что атрибут функционально зависит от всего составного потенциального ключа, но при этом не находится в функциональной зависимости от какой-либо из входящих в него частей. Или другими словами: во второй нормальной форме нет неключевых атрибутов, зависящих от части составного ключа.

Согласно требованию 2НФ необходимо задать первичный ключ, поэтому добавим в таблицы 2.1-2.4 первичный ключ, в результате чего они примут следующий вид (таблицы 2.5-2.8).

Таблица 2.5 – Данные об участниках тестирования (студентах)

ID	FIO	Login	Password	Group	Document	Contacts	More
----	-----	-------	----------	-------	----------	----------	------

Таблица 2.6 – Данные о результатах прохождения тестирования

ID	PassingID	Date	StudentID	Group	TestBinary	TestID
----	-----------	------	-----------	-------	------------	--------

Таблица 2.7 – Таблица для хранения самих тестов

ID	TestBinary
----	------------

Таблица 2.8 – Данные о времени прохождения тестирования

ID	IsDate	DateFrom	DateTo	LStudentsID	TestBinary
----	--------	----------	--------	-------------	------------

Согласно определению Кодда, таблица находится в третьей нормальной форме тогда и только тогда, когда выполняются следующие условия:

- отношение R (таблица) находится во второй нормальной форме;
- каждый непервичный атрибут R находится в нетранзитивной (то есть

прямой) зависимости от каждого ключа R.

Таким образом, отношение находится в третьей нормальной форме тогда и только тогда, когда оно находится во второй нормальной форме и отсутствуют транзитивные зависимости неключевых атрибутов от ключевых. Транзитивной зависимостью неключевых атрибутов от ключевых называется следующая: $A \rightarrow B$ и $B \rightarrow C$, где A - набор ключевых атрибутов (ключ), B и C – различные множества неключевых атрибутов.

В результате таблицы 2.5-2.8 удовлетворяют требованиям, как второй, так и третьей нормальной форме, так как не содержат составных атрибутов, и все неключевые атрибуты полностью (неприводимо) зависят от первичного ключа.

На рисунке 2.18 изображена логическая схема базы данных, созданная в среде Microsoft SQL Server Management Studio.

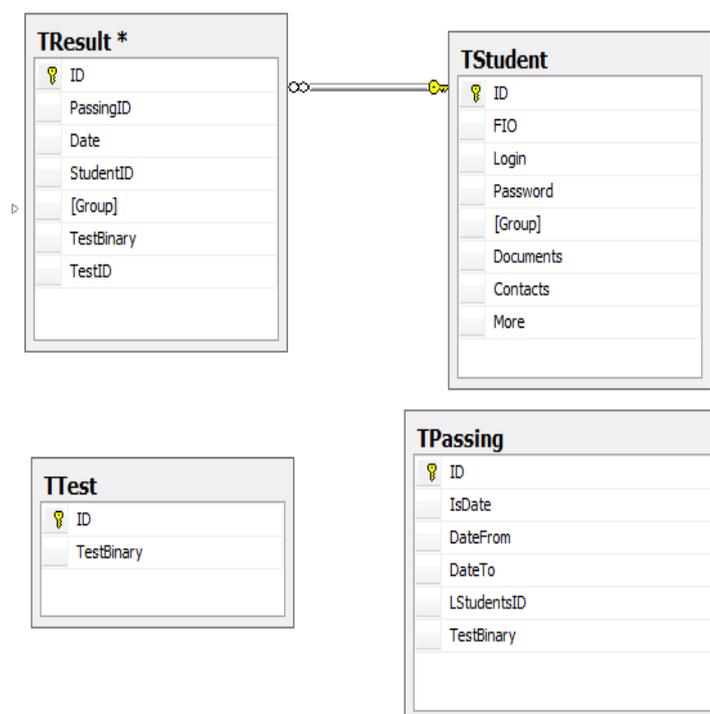


Рисунок 2.18 Логическая схема базы данных

На рисунке 2.18 отображены не все связи между таблицами, т.к. ограничения связанные со ссылочной целостностью логически связанных данных будут реализованы на уровне клиента.

2.3.3 Этап физического проектирования базы данных

На данном этапе происходит выбор СУБД в соответствии с оговоренными заранее требованиями заказчика, которые прописаны в техническом задании (отражено в приложении данной выпускной квалификационной работы).

Соответственно в качестве СУБД для создания базы данных и управления ею выбран Microsoft SQL Server 2008 R2 – современная на сегодняшний день система управления базами данных. На рисунке 2.19 изображена физическая схема базы данных, созданная в среде Microsoft SQL Server Management Studio.

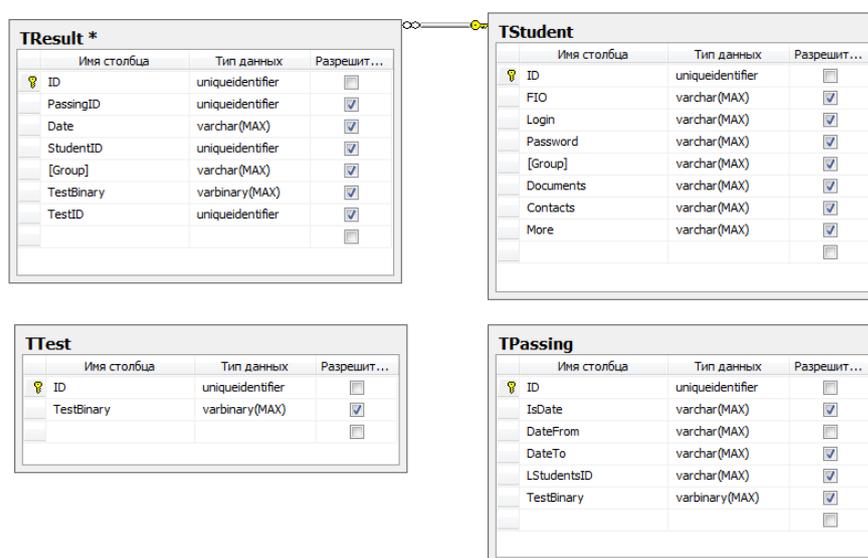


Рисунок 2.19 Физическая схема базы данных

В таблицах на рисунке 2.19 отмечены ключевые поля, типы данных полей, и возможность пустых (NULL) значений в некоторых полях.

Поля таблицы TResult (хранятся данные о тестировании студентов):

– ID (первичный ключ), PassingID (внешний ключ для связи с таблицей TPassing), StudentID (внешний ключ для связи с таблицей TStudent), TestID (внешний ключ для связи с таблицей TTest) имеют тип данных uniqueidentifier (тип данных позволяет создавать уникальные неповторяющиеся значения – используется для создания значений ключевых полей).

– Date (используется для хранения даты прохождения тестирования) и поле Group (используется для хранения названия группы, к которой прикреплен

студент) имеют символьный тип данных переменной длины –varchar(MAX).

– TestBinary (используется для хранения в двоичном виде файла с результатами ответов на вопросы теста) имеет тип данных varbinary(MAX).

В таблице TStudent поле ID (первичный ключ) имеет тип данных uniqueidentifier (тип данных позволяет создавать уникальные неповторяющиеся значения – используется для создания значений ключевых полей).

Поля таблицы TStudent, перечисленные ниже имеют символьный тип данных varchar(MAX):

- FIO (используется для хранения Ф.И.О. участников тестирования);
- Login (используется для хранения уникального логина участника тестирования);
- Password (используется для хранения пароля участника тестирования);
- Group (используется для хранения названия группы, к которой прикреплен участник тестирования);
- Documents (используется для хранения документов удостоверяющих личность участников тестирования);
- Contacts (используется для хранения контактных данных участников тестирования);
- More (используется для хранения дополнительной информации об участниках тестирования).

Поле таблицы TTest (в таблице хранятся данные в двоичном виде о тестах, включая все вопросы тестов, оценки за них, категории сложности вопросов) ID (первичный ключ) имеет тип данных uniqueidentifier.

Поле таблицы TTest TestBinary (в поле хранятся данные в двоичном виде о тестах, включая все вопросы тестов, оценки за них, категории сложности вопросов) имеет тип данных varbinary(MAX).

Поле таблицы TPassing (в таблице хранятся данные о периоде проведения тестирования по определенному тесту, определенными студентами) ID (первичный ключ) имеет тип данных uniqueidentifier.

Поля таблицы TPassing, перечисленные ниже имеют символьный тип

данных varchar(MAX):

- IsDate (дата прохождения теста определенным участником);
- DateFrom (дата, начиная с которой возможно пройти тест определенным участником);
- DateTo (дата, позже которой нельзя будет пройти определенный тест);
- LStudentsID (список студентов, которым доступен определенный тест).

Поле таблицы TPassing TestBinary (в поле хранятся данные в двоичном виде о тесте, включая все вопросы теста, оценки за них, категории сложности вопросов) имеет тип данных varbinary(MAX).

2.4 Проектирование графического пользовательского интерфейса

Одним из немаловажных этапов разработки программного продукта является процесс создания пользовательского интерфейса.

Интерфейс программы, прежде всего, определяет язык пользователя и язык сообщения компьютера, который организует с ним диалог. Расшифровать эти понятия можно очень просто: действия, которые пользователь проводит в отношении системы, путем использования различных технических средств и являются языком пользователя, а язык сообщений же – это информация, предназначенная для пользователя, которая отображается на экране.

Наиболее простой формой языка пользователя является создание форм входных и выходных документов. Важным измерителем эффективности используемого интерфейса является выбранная форма диалога пользователя с системой. Знание пользователя определяют круг вопросов, которые он должен знать при работе с системой.

Интерфейс должен обладать следующими возможностями:

- манипулировать различными формами диалога и изменяя их в процессе работы по выбору пользователя;
- передавать данные системе различными способами;
- получать данные от разных устройств системы в различных формах;
- гибко поддерживать запросы.

Так как в качестве среды разработки программного продукта была выбрана Microsoft Visual Studio 2012, то для создания графического интерфейса пользователя использовались стандартные визуальные компоненты, некоторые из которых показаны в таблице 2.9.

Таблица 2.9 – Компоненты графического интерфейса

Компонент	Описание
Form	Визуальный компонент, имеет свойства, методы и события общие для всех визуальных компонентов. Форма является контейнером для размещения на ней других визуальных компонентов
Menu Strip	Компонент служит для создания горизонтального меню
Label	Компонент служит для вывода текстовой информации
Button	Компонент типа «кнопка»
Tool Strip	Компонент типа горизонтального меню в виде кнопок
Combo Box	Компонент представляющий выпадающий список элементов
Компонент	Описание
Data Grid View	Компонент таблица служит для вывода данных хранящихся в БД, связан с набором данных с помощью компонента Data Source
Dlg Print Document	Компонент диалога вывода на печать
Dlg Open File	Компонент диалога открытия файла
Dlg Print Preview	Компонент предварительного просмотра перед выводом на печать

На рисунках 2.20 – 2.21 изображены некоторые формы модуля для создания теста (TestEditor) на этапе проектирования в среде Microsoft Visual Studio 2012.

Внешний вид диалогового окна «Редактор тестов» показан на рисунке 2.20.

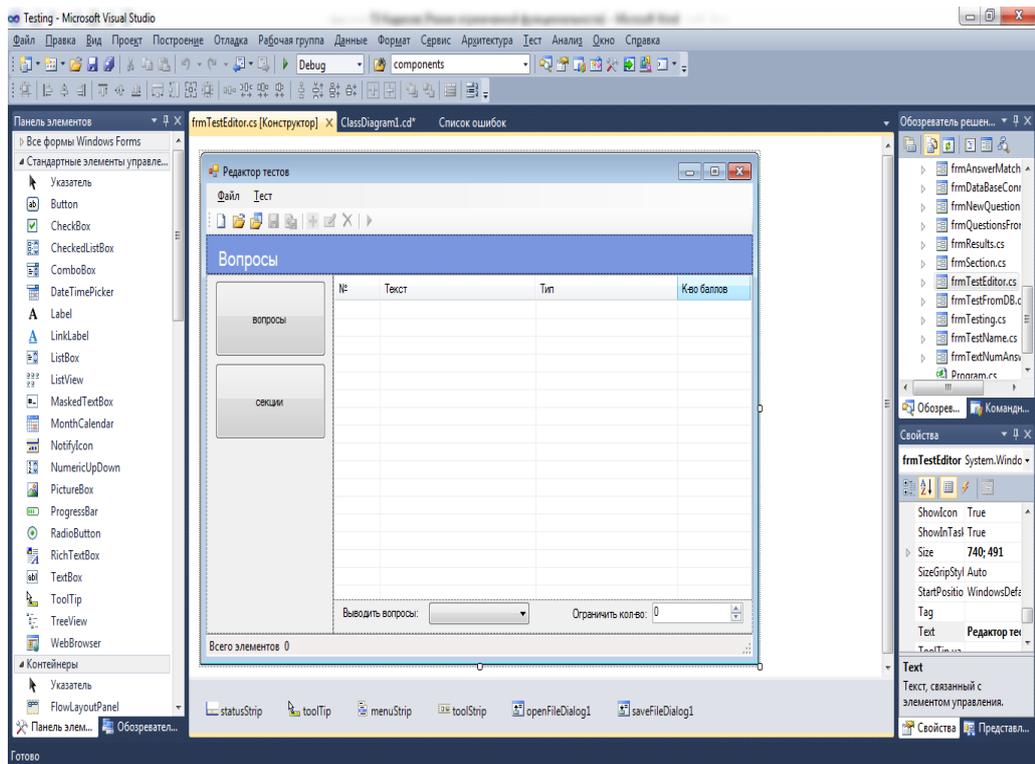


Рисунок 2.20 Внешний вид диалогового окна «Редактор тестов»

Окно выполнения теста в демонстрационном режиме представлено на рисунке 2.21.

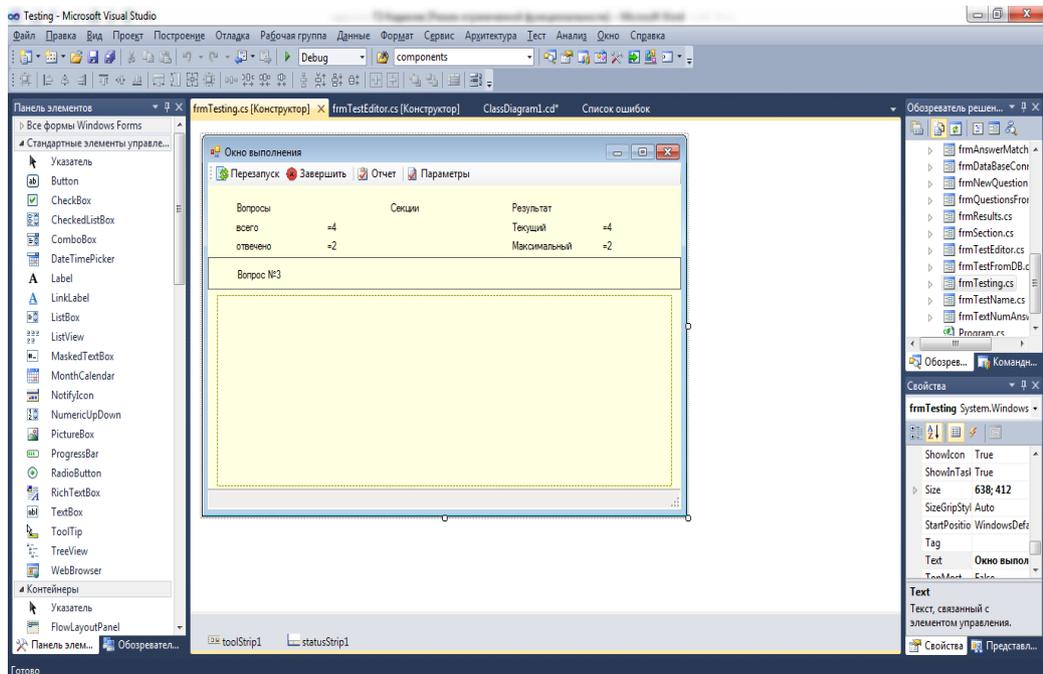


Рисунок 2.21 Окно выполнения теста в демонстрационном режиме

Окно для создания символьного ответа на вопрос теста представлено на рисунке 2.22.

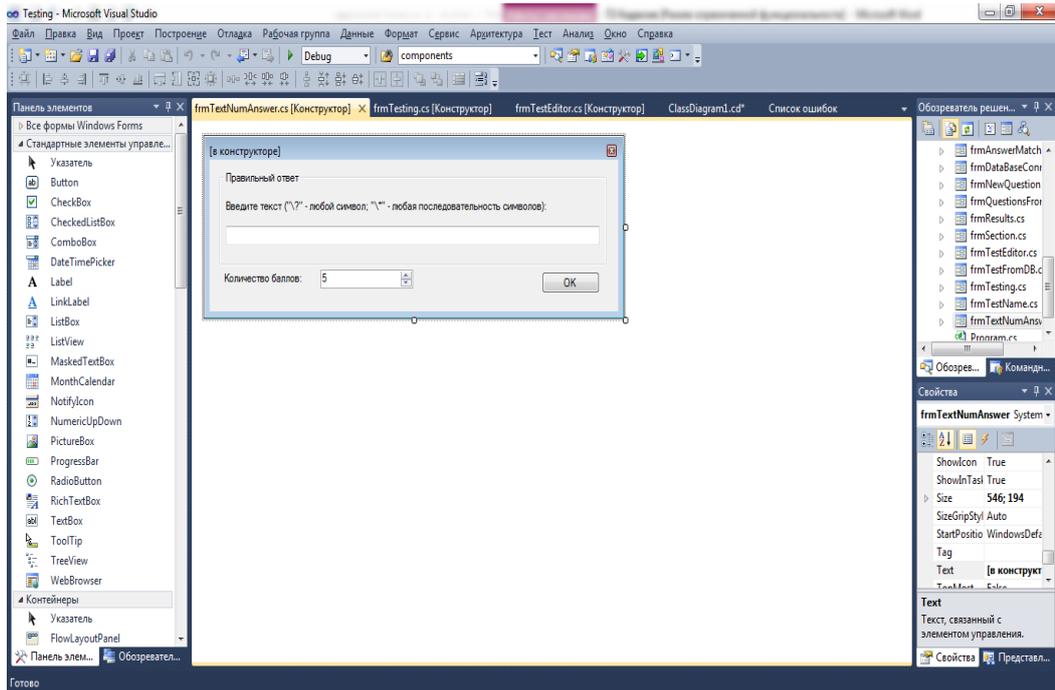


Рисунок 2.22 Окно для создания символьного ответа на вопрос теста

Окно для задания категорий сложности вопросам теста представлено на рисунке 2.23.

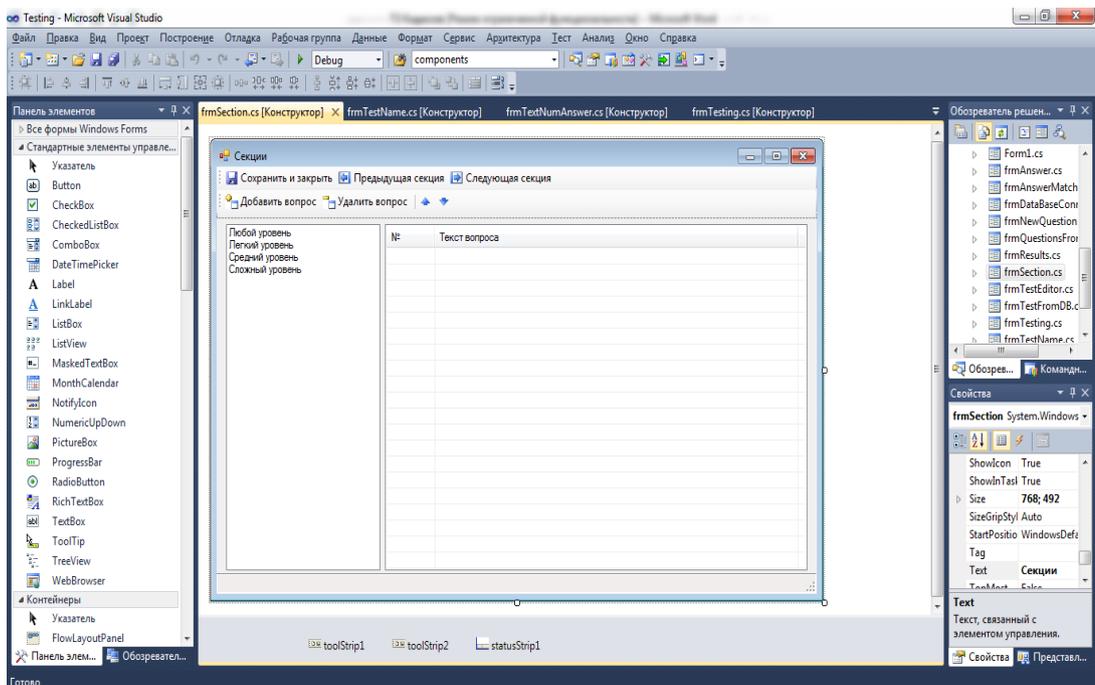


Рисунок 2.23 Окно для задания категорий сложности вопросам теста

3. ОПИСАНИЕ ПРОГРАММНОГО ПРОДУКТА

3.1 Руководство пользователя

Разработанный программный продукт состоит из трех модулей:

- Testing.exe (TestEditor);
- Testing.exe (TestAdmin);
- Testing.exe (Tester).

Первый модуль предназначен для создания тестов, второй модуль нужен для назначения тестов кому-либо, а третий для самого прохождения теста. Рассмотрим последовательно этих трех модулей, начав с рассмотрения Testing.exe (TestEditor).

Данный модуль Testing.exe (TestEditor) предназначен для создания тестов, вопросов и работой с ними.

Для того чтобы запустить данное приложение необходимо запустить файл Testing.exe (TestEditor), после этого пользователь увидит диалоговое окно, которое приведено на рисунке 3.1.

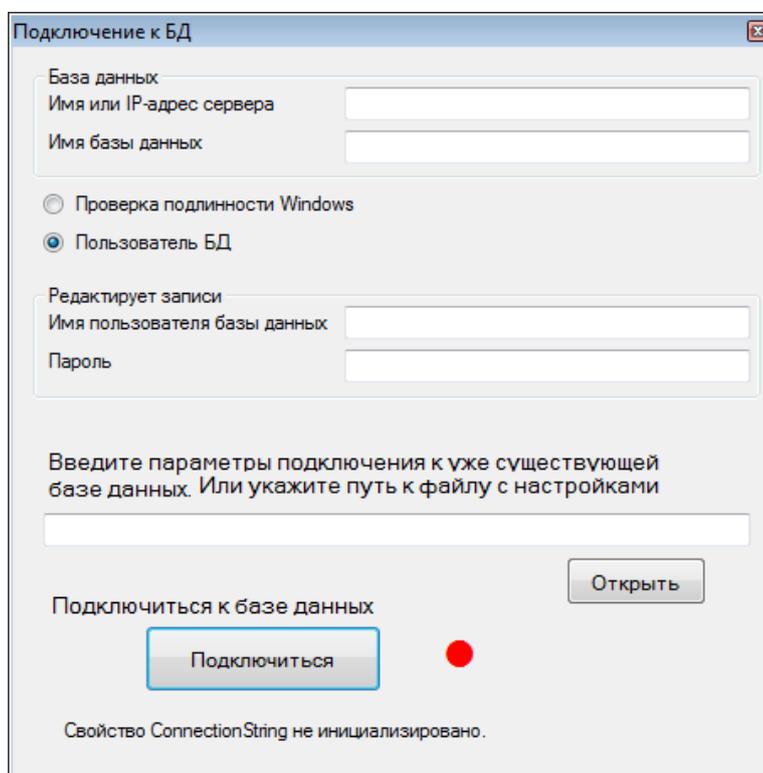


Рисунок 3.1 Подключение к БД

В данном окне пользователю необходимо выбрать базу данных или сервер. При правильном подключении к БД, система сообщит об успешном подключении к БД (рисунок 3.2).

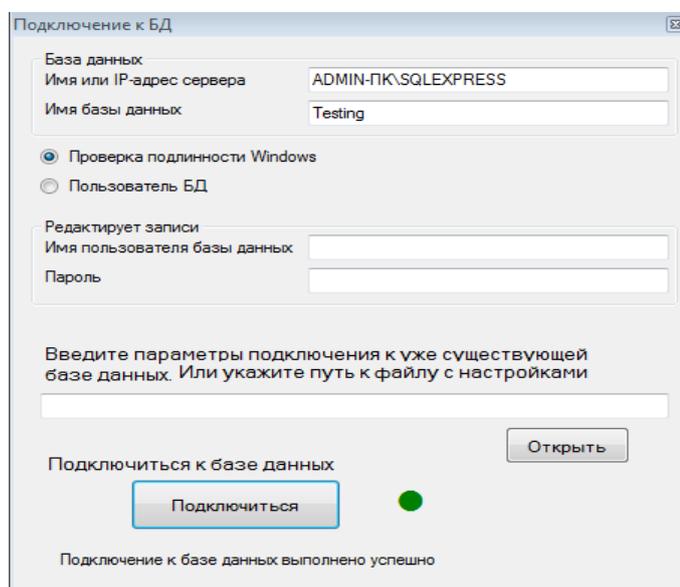


Рисунок 3.2 Подключение к БД выполнено успешно

Далее пользователь должен выйти из программы и заново зайти.

При повторном запуске Testing.exe (TestEditor) появится окно, в котором можно создавать, удалять и изменять тесты. А также создавать, изменять, удалять вопросы, выбирать порядок вывода вопросов и ограничить количество вопросов в определенном тесте (рисунок 3.3).

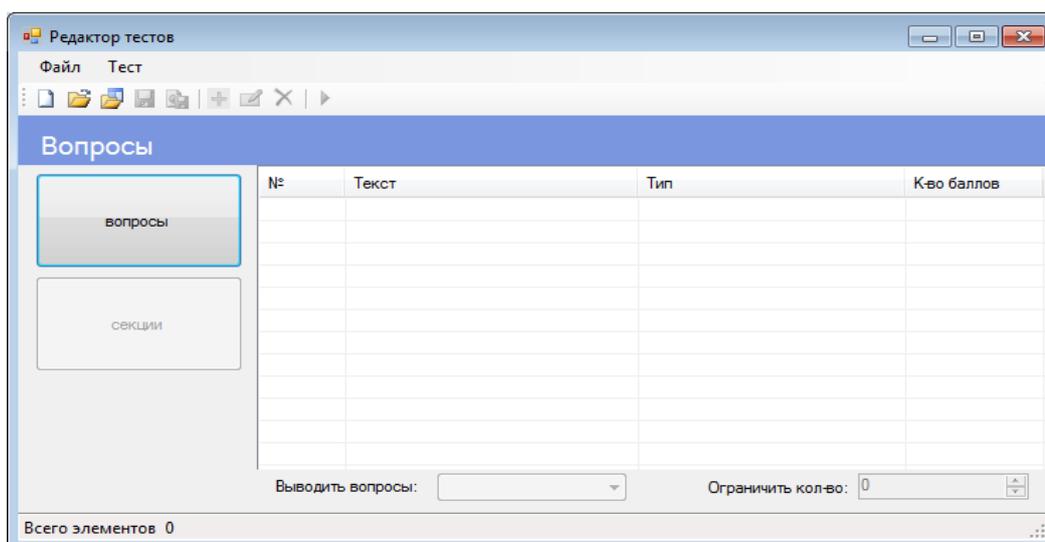


Рисунок 3.3 Редактор тестов

Для того что бы создать новый тест и вопросы надо зайти в «файл» и выбрать «создать». Перед пользователем откроется новое окно, в котором система попросит ввести название нового теста (рисунок 3.4).

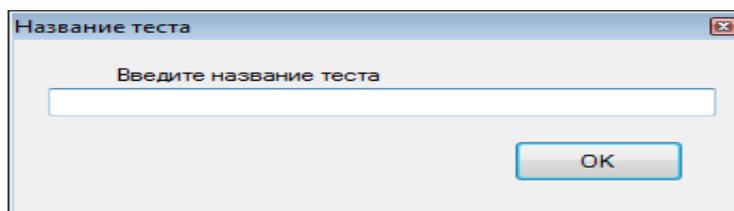


Рисунок 3.4 Создание нового теста

После того как новый тест создан, можно добавлять, изменять и удалять вопросы. Данные кнопки изображены на рисунке 3.5.



Рисунок 3.5 Кнопки работы с вопросами

Создание нового вопроса изображено на рисунке 3.6. В данном окне можно установить количество правильных ответов, режим вывода ответов, сложность вопроса, добавить вопрос, удалить вопрос.

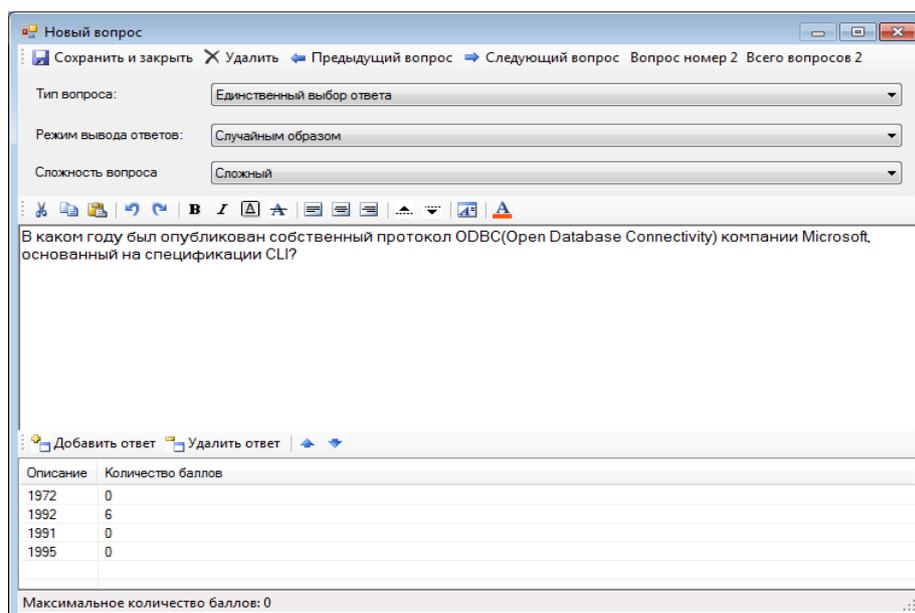


Рисунок 3.6 Создание нового вопроса

После того как вопрос создан, можно добавить ответы. В данном окне, изображенном на рисунке 3.7, указывается вариант ответа и выставляется количество баллов, если тестируемый выберет этот ответ.

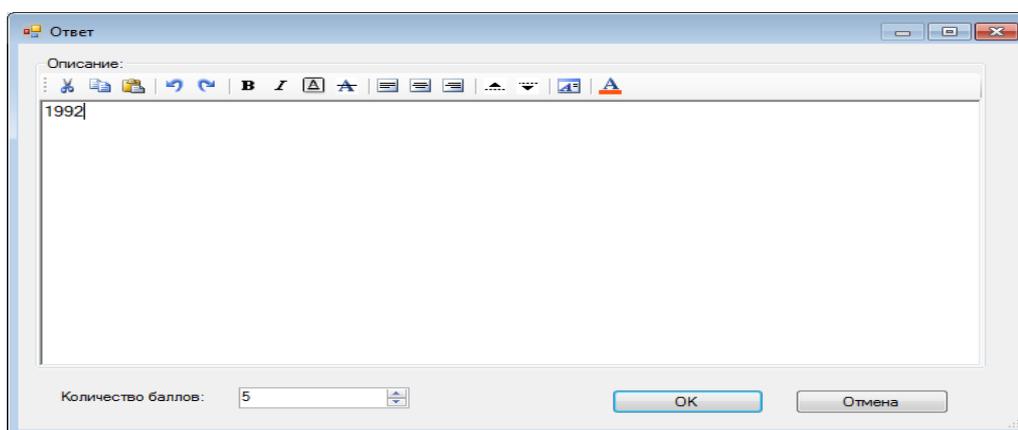


Рисунок 3.7 Создание ответов на вопрос

В секции можно также добавить, удалить вопрос, просмотреть секции и увидеть вопросы по уровню сложности (рисунок 3.8).

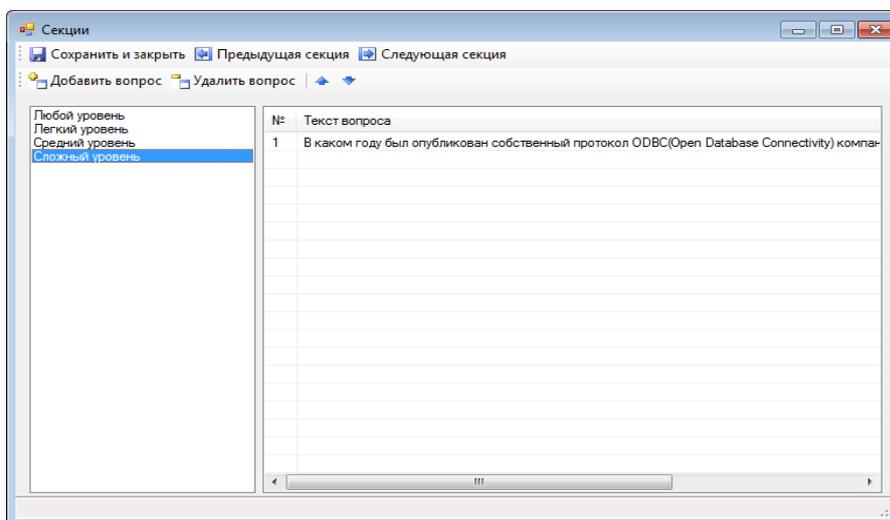


Рисунок 3.8 Вопросы сложного уровня

Для того что бы сохранить данный тест и вопросы надо выбрать «Сохранить и закрыть».

Модуль Testing.exe (TestAdmin) предназначен для администрирования. При первом запуске данного модуля появится форма для подключения БД. Пользователю дается два варианте подключения. Первый – это прописать имя

сервера и имя БД. А второй – это открыть файл с настройками подключения. Настройки подключения автоматически сохраняются при первом подключении БД к любому модулю из трех. Как БД подключится к данной программе, надо закрыть программу и заново зайти.

В данном модуле имеется возможность назначить кому-либо тест, добавить, удалить и заменить группу или пользователя. А также система позволяет увидеть результаты в виде графика (рисунок 3.9).

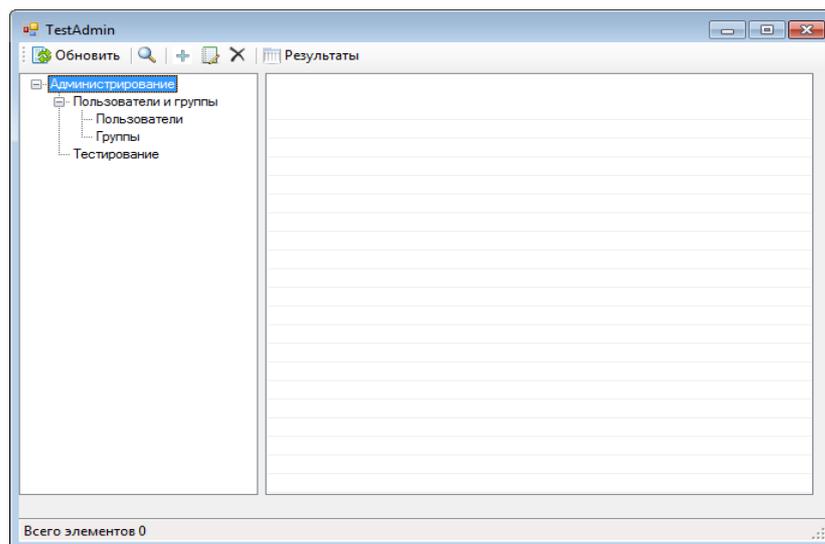


Рисунок 3.9 Главное окно Testing.exe (TestAdmin)

Для того что бы добавить нового пользователя, надо нажать на «пользователи», потом нажать «добавить». После всех этих действий откроется новое окно для добавления нового пользователя (рисунок 3.10).

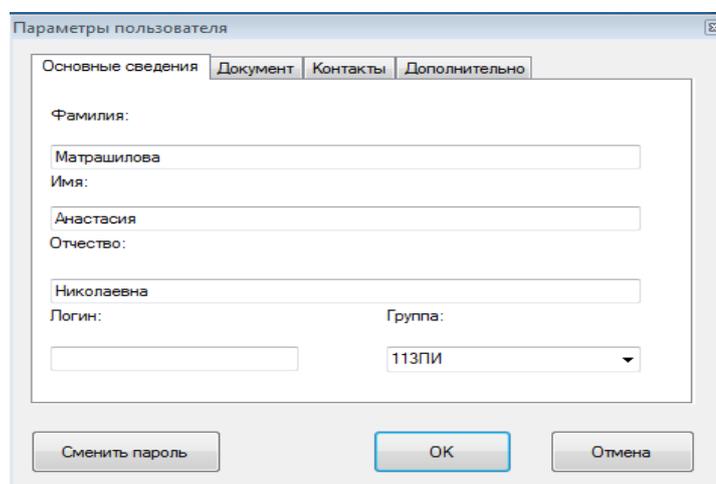


Рисунок 3.10 Добавление нового пользователя

Третий модуль это Testing.exe (Tester), он предназначен для прохождения тестов. При запуске система просит выбрать пользователя и тест. Если тесты недоступны, то данный пользователь не назначен ни на какой тест (рисунок 3.11).

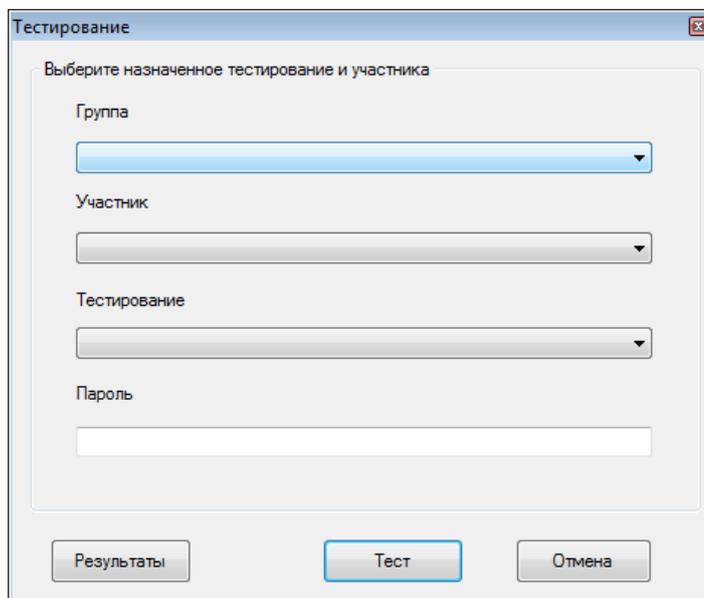


Рисунок 3.11 Выбор теста и участника тестирования

В окне выполнения пользователь может перезапустить данный тест, завершить, отчет о пройденном тесте, а также параметры.

Также участник может наблюдать сколько всего вопросов в данном тесте, на сколько он ответил, сколько баллов у него из максимального. Участник может пропустить вопрос (рисунок 3.12).

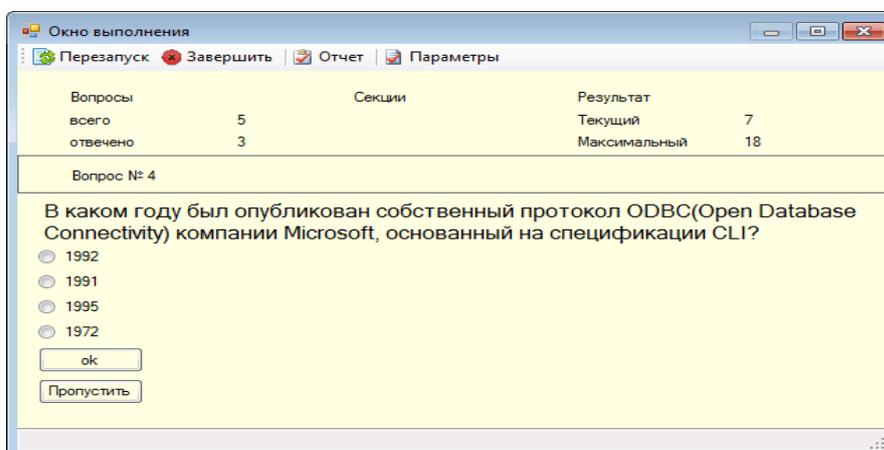


Рисунок 3.12 Тестирование

3.2 Руководство администратора

Для того чтобы программа корректно работала необходимо установить на компьютере СУБД Microsoft SQL Server и создать БД с именем Testing.

Процесс установки СУБД Microsoft SQL Server 2008 нами рассматриваться не будет, так как данный процесс является интуитивно понятным и простым. Главное, прежде чем приступить к установке СУБД Microsoft SQL Server 2008, необходимо убедиться соответствует ли конфигурация персонального компьютера, на который предстоит установка требованиям сервера данной версии 2008 express edition.

Данная версия сервера является свободно распространяемой и подходит для организации и разработки баз данных для малых предприятий и учреждений.

3.2.1 Аппаратные и программные требования СУБД MS SQL Server

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

- процессор Pentium-2.0Hz, не менее;
- оперативную память объемом, 1024 Мбайт, не менее;
- свободного пространства на жестком диске, 2 Гигабайт, не менее;
- операционную систему Windows 2003 SP2, SP3 или Windows XP Professional SP2, SP3, Windows Vista, Windows 7, Windows 8.

3.3 Результаты тестирования созданного программного продукта

Для тестирования работы программы запустим файл Testing.exe (Tester), на экране появляется следующая форма – рисунок 3.13.

В данной форме пользователю необходимо выбрать группу, участника и тест для прохождения.

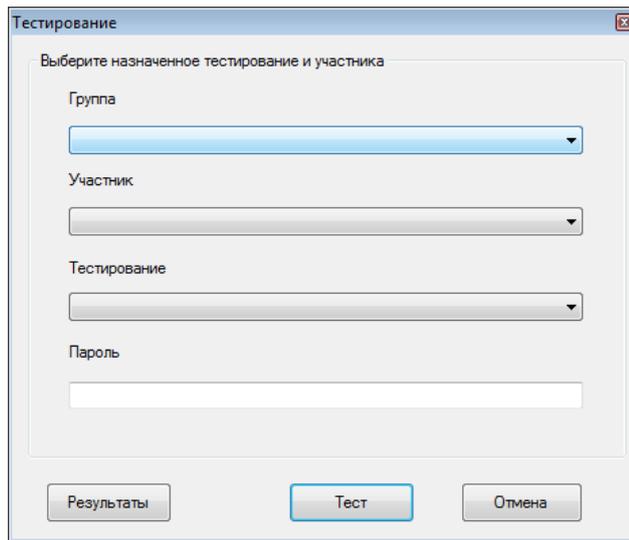


Рисунок 3.13 Выбор теста и участника для прохождения теста

В данной форме тестируемый может завершить данный тест, перезапустить, отчет о тесте и также ему доступны параметры (рисунок 3.14).

Тестируемый, выбрав ответ, должен нажать на кнопку «Ок» или, если затрудняется ответить на «пропустить». После этих действий программа автоматически перейдет на следующий вопрос.

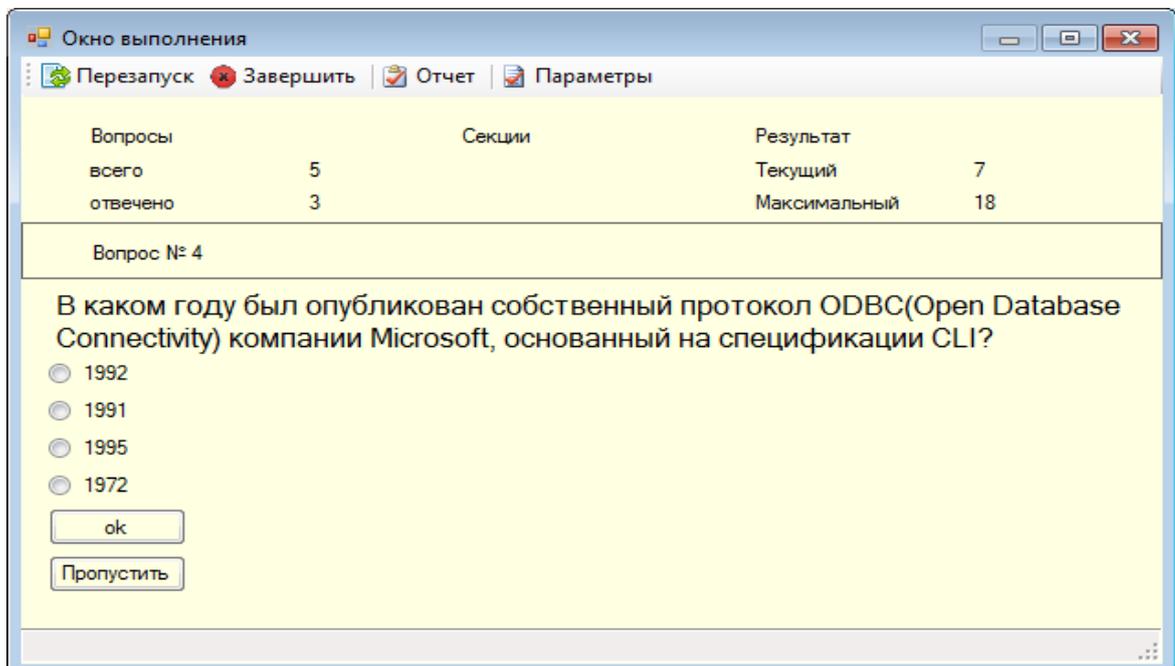


Рисунок 3.14 Тестирование

Когда пользователь ответит на все вопросы или нажмет завершить, в окне будет написано «Тест завершен!». Сверху будут написаны результаты тестирования, это всего, сколько вопросов в тесте, на сколько было дано ответов и количество набранных баллов из максимальных (рисунок 3.15).

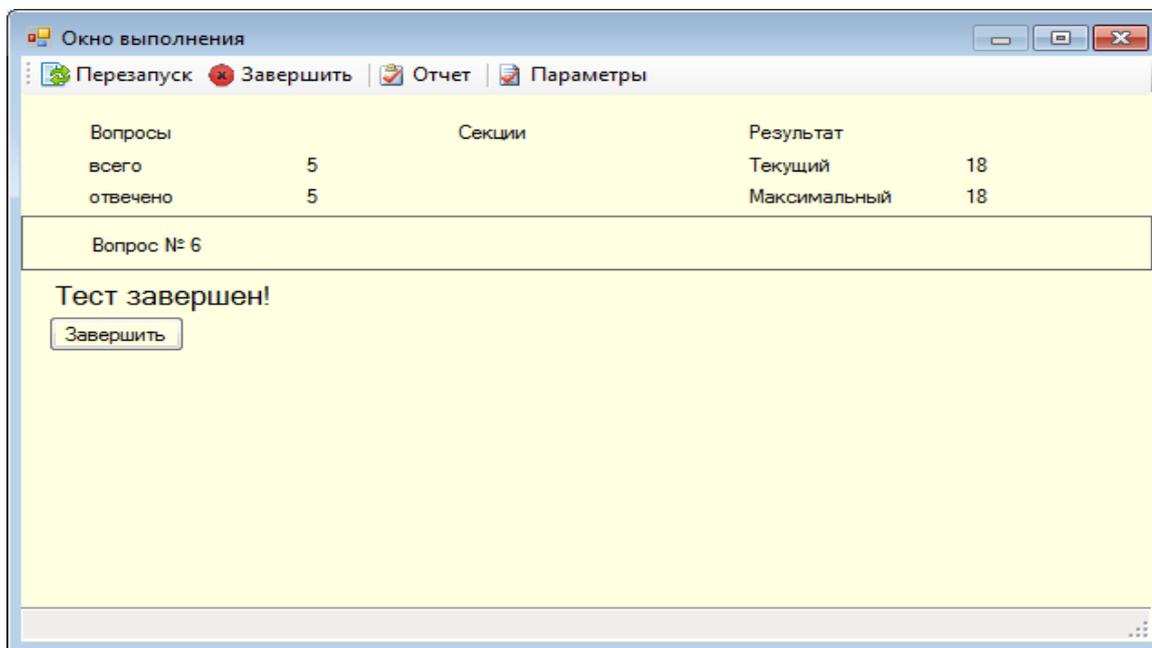


Рисунок 3.15 Результат пройденного теста

4. ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ЭФФЕКТИВНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

4.1 Оценка трудоемкости разработки программного продукта

Оценку трудоемкости разработки программного продукта мы выполним для ПФ на основе вариантов исследования и на материалах компании rational software. Для определения весовых показателей действующих лиц все действующие лица системы делятся на три типа: простые, средние и сложные (таблица 4.1). Простое действующее лицо представляет внешнюю систему с чётко определённым программным интерфейсом. Среднее действующее лицо представляет либо внешнюю систему, взаимодействующую с данной системой посредством протокола наподобие TCP/IP, либо личность, пользующуюся текстовым интерфейсом. Сложное действующее лицо представляет личность, пользующуюся графическим пользовательским интерфейсом (таблица 3.2).

Таблица 4.1 – Весовые коэффициенты действующих лиц

Тип действующего лица	Весовой коэффициент
Простое	1
Среднее	2
Сложное	3

Для разработанного программного обеспечения, действующие лица и их тип сложности приведен в таблице 4.2.

Таблица 4.2 – Типы действующих лиц

Действующее лицо	Тип
Администратор	Сложное
Преподаватель	Сложное
Студент	Сложное
Общий весовой показатель A	9

Для определения весовых показателей вариантов использования разделим их на три типа: простые, средние и сложные в зависимости от количества транзакций в потоках событий. Общее количество вариантов использования

каждого типа умножается на соответствующий весовой коэффициент, затем вычисляется общий весовой показатель (таблица 4.3).

Таблица 4.3 – Весовые коэффициенты вариантов использования

Тип варианта использования	Описание	Весовой коэффициент
Простой	3 или менее транзакций	5
Средний	От 4 до 7 транзакций	10
Сложный	Более 7 транзакций	15

Для разрабатываемой программы сложность вариантов использования приведена в таблице 4.4.

Таблица 4.4 – Сложность вариантов использования

Вариант использования	Тип
Инициализация параметров подключения к серверу	Простой
Авторизация	Простой
Создание новой сессии для тестирования	Сложный
Создание новых тестов и вопросов	Сложный
Проведение тестирования	Сложный
Общий весовой показатель UC	55

Рассчитаем показатель UUCP (Unadjusted Ted Use Case Points) по формуле 4.1:

$$UUCP = A + UC, \quad (4.1)$$

где A – общий весовой показатель действующих лиц;

UC – общий весовой показатель вариантов использования.

В итоге получаем $UUCP = 64$.

Техническая сложность проекта (TCF – Technical Complexity Factor) вычисляется с учетом показателей технической сложности (таблица 4.5).

Каждому показателю присваивается значение в диапазоне от 0 до 5 (0 означает отсутствие значимости показателя для данного проекта, 5 – высокую значимость).

Значение TCF вычисляется по формуле 4.2:

$$TCF = 0.6 + (0.01 \cdot (\sum T_i \cdot \text{Вес}_i)) \quad (4.2)$$

где T_i – показатель технической сложности проекта.

Для подсистемы учета и анализа состояния средств измерения, согласно весам (таблица 4.6), получим:

$$TCF = 0.6 + (0.01 \cdot 45) = 1.05$$

Таблица 4.5 – Показатели технической сложности проекта TCF

Показатель	Описание	Вес
T1	Распределенная система	2
T2	Высокая производительность	1
T3	Работа конечных пользователей в режиме онлайн	0
T4	Сложная обработка данных	2
T5	Повторное использование кода	1
T6	Простота установки	0,5
T7	Простота использования	0,5
T8	Переносимость	2
T9	Простота внесения изменений	1
T10	Параллелизм	1
T11	Специальные требования к безопасности	1
T12	Непосредственный доступ к системе со стороны внешних пользователей	1
T13	Специальные требования к обучению пользователей	1

Таблица 4.6 – Показатели технической сложности разработанной системы

Показатель	Вес	Значение	Значение с учетом веса
T1	2	4	8
T2	1	5	3
T3	1	2	5
T4	1	4	3
T5	1	1	3
T6	0,5	3	1,5
T7	0,5	5	1,5
T8	2	2	8
T9	1	4	3
T10	1	2	1
T11	1	2	2
T12	1	5	5
T13	1	1	1
Z			45

Уровень квалификации разработчиков (EF–Environmental Factor) вычисляется с учетом следующих показателей (таблица 4.7).

Таблица 4.7 – Показатели уровня квалификации разработчиков

Показатель	Описание	Вес
F1	Знакомство с технологией	1,5
F2	Опыт разработки приложений	0,5
F3	Опыт использования объектно-ориентированного подхода	1
F4	Наличие ведущего аналитика	0,5
F5	Мотивация	1
F6	Стабильность требований	2
F7	Частичная занятость	-1
F8	Сложные языки программирования	-1

Каждому показателю присваивается значение в диапазоне от 0 до 5.

Для показателей F1 – F4, 0 означает отсутствие, 3 – средний уровень, 5 – высокий уровень. Для показателя F5, 0 означает отсутствие мотивации, 3 – средний уровень, 5 – высокий уровень мотивации. Для F6, 0 означает высокую нестабильность требований, 3 – среднюю, 5 – стабильные требования. Для F7, 0 означает отсутствие специалистов с частичной занятостью, 3 – средний уровень, 5 – все специалисты с частичной занятостью. Для показателя F8, 0 означает простой язык программирования, 3 – среднюю сложность, 5 – высокую сложность.

Значение EF вычисляется по формуле 4.3:

$$EF = 1.4 + (-0.03 \cdot (\sum F_i \cdot \text{Вес}_i)) \quad (4.3)$$

где F_i – показатель уровня квалификации работников.

Вычислим для разработанной программы, данные для которой приведены в таблице 3.7 по формуле 4.3:

$$EF = 1.4 + (-0.03 \cdot 24,5) = 0.665$$

Результаты расчета показателей уровня квалификации разработчиков разработанной системы представлены в таблице 4.8.

Таблица 4.8 – Показатели уровня квалификации разработчиков в разработанной системе

Показатель	Вес	Значение F_i	Значение с учетом веса
F1	1,5	4	6
F2	0,5	5	2,5
F3	1	4	5
F4	0,5	1	1
F5	1	3	5
F6	2	4	8
F7	-1	2	0
F8	-1	3	-3
I			24,5

В результате получаем окончательное значение показателя UCP, которое рассчитывается по формуле 4.4:

В результате получаем по формуле 4.4 окончательное значение UCP (Use Case Points):

$$UCP = UUCP \cdot TCF \cdot EF, \quad (4.4)$$

где UUCP – общий весовой показатель;

TCF – техническая сложность проекта;

EF – уровень квалификации разработчиков.

В итоге получаем $UCP = 44,688$

В качестве начального значения предлагается использовать 20 чел./ч на одну UCP. Эта величина может уточняться с учетом опыта разработчиков. Приведем пример возможного уточнения.

Рассмотрим показатели F1 – F8 и определим, сколько показателей F1 – F6 имеют значение меньше 3 и сколько показателей F7 – F8 имеют значение больше 3. Если общее количество меньше или равно 2, следует использовать 20 чел./ч на одну UCP, если 3 или 4-28. Если общее количество равно 5 или более, следует внести изменений в сам проект, в противном случае риск провала слишком высок.

Для системы учета средств измерения получаем 20 чел./ч на одну UCP,

таким образом, общее количество чел./ч на весь проект равно 701,018, что составляет 22 недели при 40-часовой рабочей неделе. Команда разработчиков состоит из одного человека, и добавим 1 неделю на различные непредвиденные ситуации, тогда в итоге получим 23 недели на весь проект.

4.2 Расчет показателей экономической эффективности внедрения программного продукта

В результате внедрения программного продукта ожидается снижение трудоёмкости работ при системном контроле объекта и условно-годовая экономия фонда заработной платы в организации при использовании разработанной системы.

Для расчета экономического эффекта, который будет получен, при разработке программного компонента для информационной системы образовательного процесса в среде Microsoft Visual Studio 2012, необходимо рассчитать ряд показателей входящих в понятие экономической эффективности, таблица 4.9.

Таблица 4.9 – Показатели экономической эффективности

Наименование показателя	Обозначение	Значение	Ед. измерения
Срок разработки программы	tr	5	мес.
Основная заработная плата разработчиков	Взп	17000	руб./мес.
Количество пользователей	N	3	чел.
Количество рабочих дней за 20 недель	Кдн	100	день
Режим использования компьютера	P	8	час/сут.
Стоимость компьютера	Zэвм	20000	руб.
Время отладки, установки и внедрения	to	20	час.
Стоимость одного часа работы ЭВМ	Зчк	8	руб.

За время разработки программного обеспечения были израсходованы материалы, стоимость которых приведена в таблице 4.10.

Таблица 4.10 – Перечень израсходованных материалов

Наименование	Количество, шт.	Цена, руб.	Сумма, руб.
Тонер для картриджа принтера	1	250	250
Бумага (пачка 500 листов)	2	130	270
Диск CD-R	1	20	20
Итого, Вт:			540

Рассчитаем затраты на заработную плату, ЗП, руб. по следующей формуле 4.5:

$$ЗП = t_p \cdot Вэп, \quad (4.5)$$

где t_p – время разработки программы, мес.;

$Вэп$ – основная заработная плата разработчиков, руб.

Рассчитаем показатель ЗП по формуле (4.5)

$$ЗП = 5 \cdot 17000 = 85000 \text{ руб.}$$

Рассчитаем эксплуатационные затраты на разработку программного продукта, $Зэкс$, руб по следующей формуле 4.6:

$$Зэкс = T_{мв} \cdot Зчк, \quad (4.6)$$

где $T_{мв}$ – время использования ПК при разработке программного продукта, час./сут.;

$Зчк$ – стоимость одного часа работы ПК, руб.

Рассчитаем время использования ПК при разработке программного продукта, $T_{мв}$, час. по следующей формуле 4.7:

$$T_{мв} = P \cdot Кдн, \quad (4.7)$$

где P – режим использования компьютера, час/сут.;

$Кдн$ – количество рабочих дней за 20 недель, дней.

Рассчитаем показатель $T_{мв}$ по формуле 4.7:

$$T_{MB} = 8 \cdot 100 = 800 \text{ час.}$$

Рассчитаем показатель $Z_{\text{эксп}}$ по формуле 4.6:

$$Z_{\text{эксп}} = 800 \cdot 8 = 6400 \text{ руб.}$$

Рассчитаем полную себестоимость программного продукта, $C_{\text{пп}}$, руб. по следующей формуле 4.8:

$$C_{\text{пп}} = Z_{\text{зп}} + W_{\text{н}} + Z_{\text{эксп}} + Z_{\text{пр}}, \quad (4.8)$$

где $Z_{\text{зп}}$ – затраты, связанные с заработной платой, руб.;

$W_{\text{н}}$ – затраты на расходные материалы, руб.;

$Z_{\text{эксп}}$ – эксплуатационные затраты на разработку программного продукта, руб.;

$Z_{\text{пр}}$ – прочие затраты (амортизация, командировки и прочие расходы), руб.

Рассчитаем себестоимость программного продукта, $C_{\text{п}}$, руб. по следующей формуле 4.9:

$$C_{\text{п}} = (Z_{\text{зп}} + W_{\text{н}} + Z_{\text{эксп}} + Z_{\text{пр}}) / N, \quad (4.9)$$

где $Z_{\text{зп}}$ – затраты, связанные с заработной платой, руб.;

$W_{\text{н}}$ – затраты на расходные материалы, руб.;

$Z_{\text{эксп}}$ – эксплуатационные затраты на разработку программного продукта, руб.;

$Z_{\text{пр}}$ – прочие затраты (амортизация, командировки и прочие расходы), руб.;

N – количество пользователей программы, чел.

Налог на доходы физических лиц НДФЛ = 13% и отчисления в фонды ОФ = 28%, из них ПФР = 20%, ФСС = 2,9%, ФФОМС – 5,1%, $W_0 = 0.34$.

Рассчитаем затраты, связанные с заработной платой, $Z_{\text{зп}}$, руб. по следующей формуле 4.10:

$$Ззп = ЗП + (ЗП \cdot W_{дд}), \quad (4.10)$$

где ЗП – заработная плата, руб.

Рассчитаем прочие затраты (амортизация, командировки и прочие расходы), Зпр, руб. по следующей формуле 4.11:

$$Зпр = (Ззп + W_{н} + З_{\text{эксп}}) \cdot 0,3, \quad (4.11)$$

где Ззп – затраты, связанные с заработной платой, руб.;

$W_{н}$ – затраты на расходные материалы, руб.;

$З_{\text{эксп}}$ – эксплуатационные затраты на разработку ПО, руб.

Для расчета полной себестоимости программного продукта необходимо определить затраты связанные с заработной платой и прочие затраты.

Рассчитаем затраты связанные с заработной платой по формуле 4.10:

$$Ззп = 85000 + (40000 \cdot 0,13) = 96050 \text{ руб.}$$

Рассчитаем прочие затраты по формуле 4.11:

$$Зпр = (96050 + 540 + 64000) \cdot 0,3 = 30897 \text{ руб.}$$

Рассчитаем полную себестоимость программного продукта по формуле (4.8)

$$С_{пп} = 96050 + 540 + 30897 + 6400 = 133887 \text{ руб.}$$

Рассчитаем себестоимость программного продукта по формуле 4.9:

$$С_{п} = (96050 + 540 + 30897 + 6400) / 3 = 44629 \text{ руб.}$$

Рассчитаем цену программного продукта, рассчитывая по заданной норме рентабельности ($R=30\%$), Цпп, руб. по следующей формуле 4.12:

$$Ц_{пп} = С_{пп} + (R \cdot С_{пп}), \quad (4.12)$$

где $С_{пп}$ – полная себестоимость программного продукта, руб.;

R – норма рентабельности, %.

Рассчитаем цену программного продукта по формуле 4.12:

$$Ц_{пп} = 133887 + (133887 \cdot 0,3) = 174053,1 \text{ руб.}$$

Рассчитаем балансовую прибыль разработчика, БП, руб. по следующей

формуле:

$$БП = N \cdot (Цпп - Спп), \quad (4.13)$$

где N – количество пользователей программы, чел.;

$Цпп$ – полная цена программного продукта, руб.;

$Спп$ – полная себестоимость программного продукта, руб.

Рассчитаем балансовую прибыль разработчика по формуле 4.13:

$$БП = 3 \cdot (174053,1 - 133887) = 120498,3 \text{ руб.}$$

Рассчитаем чистую прибыль разработчика, ЧП, руб. по следующей формуле:

$$ЧП = БП - (N_{ст} \cdot БП), \quad (4.14)$$

где $N_{ст}$ – ставка налога на прибыль ($N_{ст} = 20\%$);

$БП$ – балансовая прибыль, руб.

Рассчитаем чистую прибыль разработчика по формуле 4.14:

$$ЧП = 120498,3 - (0,2 \cdot 120498,3) = 96398,64 \text{ руб.}$$

Рассчитаем капитальные затраты, связанные с разработкой программного продукта, КЗ, руб. по следующей формуле:

$$КЗ = (N \cdot Цпк) + КЗпр, \quad (4.15)$$

где $Цпк$ – цена компьютера, руб.;

N – количество компьютеров, шт.;

$КЗпр$ – прочие капитальные затраты руб.

Рассчитаем капитальные затраты, связанные с разработкой программного продукта по формуле 4.15:

$$КЗ = (1 \cdot 20000) + 3500 = 23500 \text{ руб.}$$

Рассчитаем срок окупаемости, Ток, по следующей формуле:

$$Ток = КЗ / ЧП, \quad (4.16)$$

где $КЗ$ – капитальные затраты, связанные с разработкой программного

продукта, руб.;

ЧП – чистая прибыль разработчика, руб.

Рассчитаем срок окупаемости для двух пользователей информационной системы по формуле 4.16:

$$\text{Ток} = 23500/96398,64 = 0,24 \text{ г.}$$

Т.о., срок окупаемости составляет 0,24 года или 2,88 месяца. Выделим основные рассчитанные экономические показатели в таблицу 4.11.

Таблица 4.11 – Основные экономические показатели для 3-х пользователей

Показатель	Значение
Цена программного продукта, руб.	174053,1
Полная себестоимость программного продукта, руб.	133887
Чистая прибыль, руб.	96398,64
Срок окупаемости, лет	0,24

В итоге стало известно, что затраты окупаются практически за полгода. Чистая прибыль (для 3-х пользователей) составляет 96398,64 руб. на 1 единицу программного продукта, а чистая прибыль (для 1-го пользователя) составляет 32132,88 руб. на 1 единицу программного продукта.

ЗАКЛЮЧЕНИЕ

В выпускной квалификационной работе проведен сравнительный анализ существующих автоматизированных обучающих систем включающих в себя подсистемы контроля знаний. Показана необходимость исследования сетевых технологий для создания распределенных автоматизированных обучающих систем, обеспечивающих удаленный доступ к системе и объединение сетевых ресурсов для решения стоящих перед системой задач.

Сформулированы и решены задачи, возникающие при создании ИС тестового контроля знаний студентов средних учебных заведений.

Выполнено проектирование с применением объектно-ориентированного подхода программного обеспечения позволяющего создавать новые тесты, вопросы для тестов различных типов (единственный выбор ответа, множественный выбор ответа, сопоставление, числовой ответ, текстовый ответ), присваивать вопросам теста различные категории сложности.

Также разработанное ПО позволяет автоматизировать процессы связанные с проведением тестирования, что особенно актуально в связи с введением ЕГЭ в школах, широким использованием компьютерного тестирования повсеместно в средних учебных заведениях.

В целом разработанное ПО позволяет хранить в базе данных информацию обо все тестах, обо всех участниках тестирования и их результатах в процессе прохождения тестирования.

В выпускной квалификационной работе также выполнены расчеты экономического эффекта от внедрения ПО, рассчитаны себестоимость программного продукта и его цена, а также срок окупаемости ПО.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Аляев Ю., Козлов О. Алгоритмизация и моделирование информационных систем. — М.: Финансы и статистика, 2011. — 435 с.
2. Бойко В. В., Савинков В. М. Проектирование баз данных информационных систем. — М.: Финансы и статистика, 2011. — 351 с.
3. Боуман Д, Эмерсон С., Дарновски М. Практическое руководство по SQL. — Киев: Диалектика, 2012. — 321 с.
4. Васкевич Д. Стратегии клиент/сервер. — Киев: Диалектика, 2012. — 516 с.
5. Гайдамакин Н. А. Автоматизированные информационные системы, базы и банки данных. Вводный курс — СПб: Гелиос АРВ, 2012. — 429 с.
6. Гилуа М. М. Множественная модель данных в информационных системах. — М.: Наука, 2011. — 243 с.
7. Голицына О. Л., Попов И. И. Основы моделирования информационных систем: Учебное пособие. — М.: Форум: Инфра-М, 2013. — 593 с.
8. Грабер М. Введение в SQL. — М.: Лори, 2014. — 379 с.
9. Грабер М. Справочное руководство по SQL. — М.: Лори, 2013. — 291 с.
10. Грейди Буч. Объектно-ориентированный анализ и моделирования информационных систем с примерами приложений на C++ /Пер. с англ. — 2-е изд. — М.: Бином, 2011. — 632 с.
11. Дейт К. Руководство по реляционной СУБД DB2. — М.: Финансы и статистика, 2012. — 320 с.
12. Дейт К. Введение в системы баз данных//6-издание. — Киев: Диалектика, 2012. — 784 с.
13. Джеймс Фокселл. Освой самостоятельно основы моделирования информационных систем за 24 часа. — М.: Вильямс, 2012. — 305 с.
14. Джексон Г. Проектирование реляционных баз данных для

использования с микроЭВМ. — М.: Мир, 2011. — 252 с.

15. Диго С. М. Проектирование и использование баз данных. — М.: Финансы и статистика, 2013. — 208 с.

16. Дэн Кларк. Основы моделирования информационных систем. — СПб.: Питер, 2014. — 317 с.

17. Емельянова Н. З., Партыка Т. Л., Попов И. И., Основы построения автоматизированных информационных систем: учебное пособие. — М.: ФОРУМ: ИНФА-М, 2012. — 291 с.

18. Зак Д. Моделирование информационных систем. — Киев: ВНУ; СПб.: Питер, 2011. — 291 с.

19. Злуф М. М. Query-by-Example: язык баз данных//СУБД. — 2011. — № 3. — С.149—160.

20. Кетков Ю., Кетков А. Практика моделирования информационных систем: Visual Basic, C++ Builder. — СПб.: ВНУ, 2012. — 572 с.

21. Кириллов В. В. Структурированный язык запросов (SQL). — СПб.: ИТМО, 2012. — 80 с.

22. Кузнецов С. Д. Введение в системы управления базами данных//СУБД. — 2011. — №1, 2, 3, 4. 2012. — №1, 2, 3, 4, 5.

23. Кузнецов С. Д. Стандарты языка реляционных баз данных SQL: краткий обзор//СУБД. — 2012. — №2. — С. 6—36.

24. Кузин. А. В., Демин В. М. Разработка БД в системе MS Access. — СПб.: Питер, 2013. — 473 с.

25. Пономарев В. Основы моделирования информационных систем: — Экспресс-курс. — СПб.: ВНУ — Санкт-Петербург, 2012. — 356 с.

26. Семакин И. Г., Шестаков А. П. Основы моделирования информационных систем. Учебник. — М.: Мастерство, 2011. — 483 с.

27. Соловьева М. Ф. Сетевые технологии. Учебник-практикум. — СПб.: БХВ-Петербург, 2012. — 476 с.

ПРИЛОЖЕНИЕ 1

Техническое задание на разработку ИС тестового контроля студентов

1.1 Введение

1.1.1 Наименование программы

Наименование программы: «контроль знаний студентов».

1.1.2 Назначение и область применения

Обеспечение автоматизации процессов связанных с прохождением компьютерного тестирования студентов.

Программа предназначена для:

- создания тестов и тестовых вопросов;
- хранения результатов тестирования в централизованной базе данных;
- прохождения тестирования студентами, управления пользователями и группами.

Цель данной разработки – автоматизация процессов прохождения тестирования студентами ГБПОУ КК «Туапсинского гидрометеорологического техникума».

1.2 Основания для разработки

1.2.1 Документы, на основании которых ведется разработка

Использованные ГОСТы:

ГОСТ 19.201 - 78. Техническое задание. Требования к содержанию и оформлению.

Наименование программы: «контроль знаний студентов»

1.2.2 Должностные обязанности:

- пользователь отвечает за введение информации в базу данных;
- администратор отвечает за целостность информации.

1.3 Требования к программе

1.3.1 Требования к функциональным характеристикам

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

- создание новых тестов и тестовых вопросов;
- создание вопросов теста различных типов (единственный выбор ответа, множественный выбор ответа, ответ на сопоставление, текстовый ответ, числовой ответ);
- добавление информации в базу данных новых участников тестирования;
- предоставление пользователю программы возможности редактировать информацию в базе данных (Ф.И.О. пользователей, названия тестов, названия вопросов теста, количество ответов в тестовом вопросе, ты сложности тестовых вопросов и т.д.);
- возможность администрирования процессов связанных с проведением тестирования (назначение времени прохождения теста, назначение пользователей которым будет доступен в определенное время данный тест, создание новых групп и привязка к ним пользователей);
- формирование результатов о прохождении тестирования;
- возможность многократного прохождения тестирования по одному и тому же тесту.

1.3.2 Требования к надежности

1.3.2.1 Требования к обеспечению надежного функционирования программы

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением Заказчиком совокупности организационно-технических мероприятий, перечень которых приведен ниже:

- а) организацией бесперебойного питания технических средств;
- б) использованием лицензионного программного обеспечения;
- в) ограничение доступа пользователей к базе данных с целью предотвращения несанкционированного доступа;
- г) обеспечение целостности базы данных;
- д) периодическое, не реже одного раза в неделю, резервирование информации, находящейся в базе данных.

1.3.2.2 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 40 минут при условии соблюдения условий эксплуатации технических и программных средств.

1.3.2.3 Отказы из-за некорректных действий пользователей системы

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу конечного пользователя без предоставления ему административных привилегий.

1.3.3 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 1 штатной единицы – пользователь.

Пользователь должен обладать основными навыками работы с компьютером, знать основы работы с Windows - приложениями.

1.3.4 Требования к составу и параметрам технических средств

В состав технических средств должен входить персональный компьютер, включающий в себя процессор Pentium IV с частотой не менее 1.0 GHz, оперативную память объемом не менее 512 Мегабайт, свободное место на жестком диске не менее 100 Мегабайт, операционную систему не ниже Windows XP.

1.3.5 Требования к информационной и программной совместимости

Информационная система должна работать на основе СУБД MS SQL Server 2008. Языком программирования является язык C#. Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы Windows.

1.3.6 Требования к эргономике

Информационная система должна обладать интуитивно понятным пользовательским интерфейсом, выполненным в спокойных тонах серо-голубого цвета. Оконный интерфейс в стиле Windows. Все надписи должны быть разборчивыми и соответствовать официальному стилю. Для удобства интерфейса должна быть предусмотрена панель навигации, включающая в себя инструменты фильтрации для разных режимов анализа. Электронная форма должна соответствовать формам реального документа.

1.4 Требования к программной документации

В комплект документации в обязательном порядке должны входить:

– техническое задание;

– программная документация в виде пакета диаграмм, оформленная в соответствии с ЕСПД (ГОСТ 19);

– текст программы;

– руководство пользователя;

– руководство программиста;

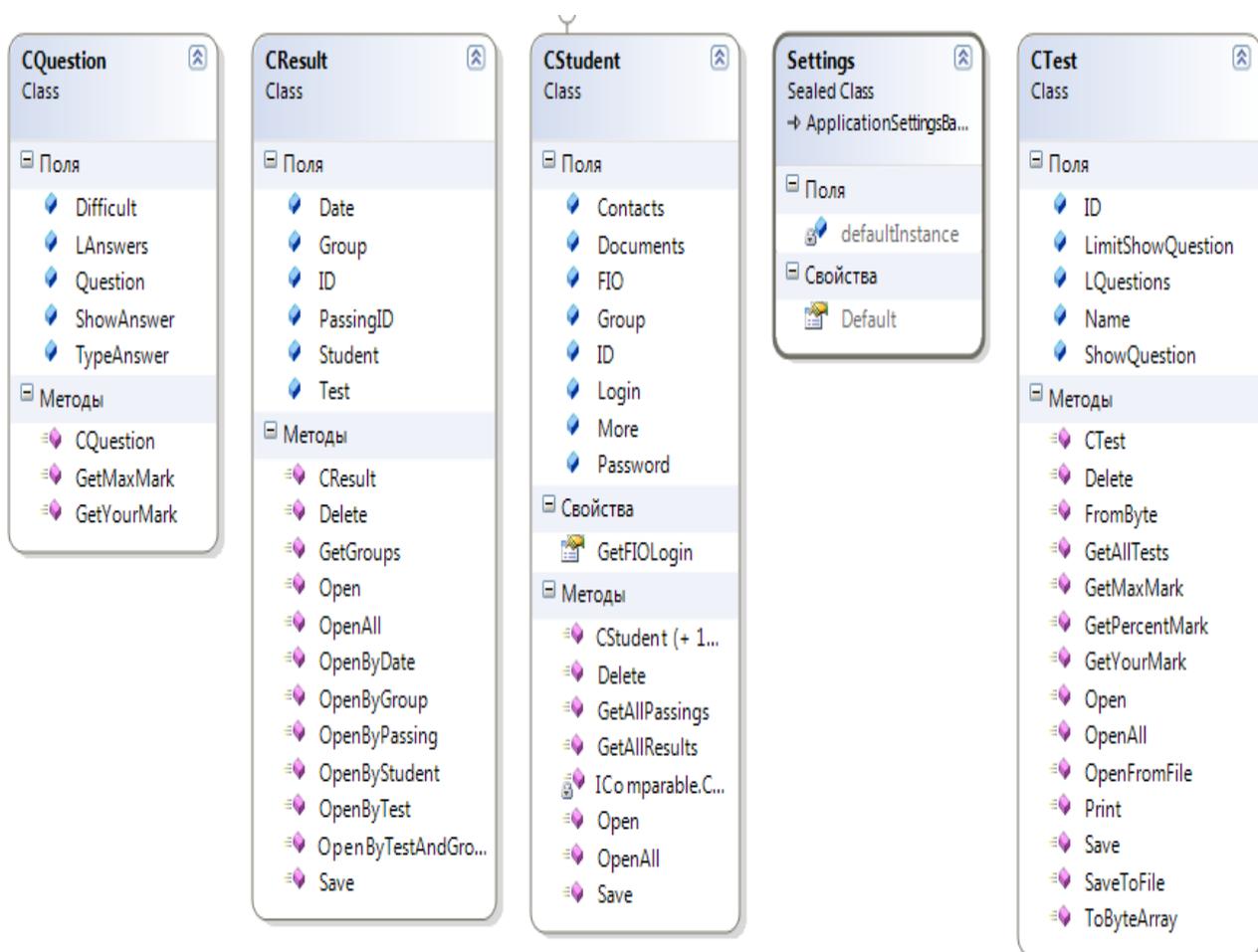
– программа и методика испытаний в виде тестирования.

1.5 Технико-экономические показатели

В разделе технико-экономического обоснования эффективности программного обеспечения дается техническое и экономическое обоснование целесообразности реализации данного проекта. Рассчитываются затраты на разработку программного обеспечения и приобретение аппаратных программных средств.

ПРИЛОЖЕНИЕ 3

Диаграмма классов модуля «Тестирование»



ПРИЛОЖЕНИЕ 4

Схема классов модуля «Редактор тестов»

The image displays five class windows from the Visual Studio IDE, showing the class hierarchy for the 'Test Editor' module. Each window lists the fields and methods for a specific class.

- frmAnswerMatch Class**
 - Поля: btnAlignCenter, btnAlignLeft, btnAlignRight, btnBold, btnCancel, btnCopy, btnCut, btnFontColor, btnFontDialog, btnItalic, btnOk, btnPaste, btnRedo, btnStrikeout, btnSubscript, btnSuperscript, btnUnderline, btnUndo, components, groupBox2, label1, label2, numMark, rxt, toolStrip1, toolStripSeparator1, toolStripSeparator2, toolStripSeparator3, toolStripSeparator4, toolStripSeparator5, toolStripSeparator6, txtMatch.
 - Методы: btnAlignCenter_Click, btnAlignLeft_Click, btnAlignRight_Click, btnBold_Click, btnCopy_Click, btnCut_Click, btnFontColor_Click, btnFontDialog_Click, btnItalic_Click, btnOk_Click, btnPaste_Click, btnRedo_Click, btnStrikeout_Click, btnSubscript_Click, btnSuperscript_Click, btnUnderline_Click, btnUndo_Click, Dispose, frmAnswerMatch, InitializeComponent.
- frmDataBaseConnect Class**
 - Поля: btnConnect, btnOpenFileCon, btnRestart, components, folderBrowserDialog1, groupBox1, groupBox2, label10, label11, label13, label4, label6, label7, label9, lblCheck, lblStatus, openFileDialog1, rbidUser, rbUser, toolTip1, txtDataSource, txtInitialCatalog, txtPassword, txtPathFileConfig, txtUser.
 - Методы: btn_OpenFolder_Click, btnConnect_Click, btnRestart_Click, Dispose, frmDataBaseConnect, InitializeComponent, lblStatus_TextChanged, rbUser_CheckedChanged.
- frmTestFromDB Class**
 - Поля: btnDelete, btnOk, columnHeader1, columnHeader2, columnHeader3, groupBox1, components, LTest.
 - Методы: btnDelete_Click, btnOK_Click, Dispose, frmTestFromDB, InitializeComponent, ShowAll.
- frmSection Class**
 - Поля: btnAddQuestion, btnDeleteQuestion, btnDown, btnNextSection, btnPreviousSection, btnSave, btnUp, columnHeader1, columnHeader2, components, Difficult, LQuestions, LQuestionsNow, lstLevel, lstVQuestions, panel1, statusStrip1, toolStrip1, toolStrip2, toolStripSeparator1.
 - Методы: btnAddQuestion_Click, btnDeleteQuestion_Click, btnDown_Click, btnNextSection_Click, btnPreviousSection_Click, btnSave_Click, btnUp_Click, Dispose, frmSection, GetAnswersByDifficult, InitializeComponent, listBox1_SelectedIndexChanged, listBox1_SelectedIndexChanged, ShowAll.
- frmNewQuestion Class**
 - Поля: btnAddAnswer, btnAlignCenter, btnAlignLeft, btnAlignRight, btnBold, btnCopy, btnCut, btnDelete, btnDeleteAnswer, btnDownAnswer, btnFontColor, btnFontDialog, btnItalic, btnNextQuestion, btnPaste, btnPreviousQu..., btnRedo, btnSaveAndClo..., btnStrikeout, btnSubscript, btnSuperscript, btnUnderline, btnUndo, btnUpAnswer, cmbDifficult, cmbShowAnswer, cmbTypeAnswer, columnHeader2, components, fontDialog1, label1, label3, label4, lblCountQ, lblNumQ, listView1, LQuestion, Num, panel1, panel2, panel3, panel4, panel5, panel6, quest, rxt, statusStrip1, toolStrip1, toolStrip2, toolStrip3, toolStripSeparator1, toolStripSeparator2, toolStripSeparator3, toolStripSeparator4, toolStripSeparator5, toolStripSeparator6, Описание.
 - Методы: btnAddAnswer..., btnAlignCenter..., btnAlignLeft_Cli..., btnAlignRight..., btnBold_Click, btnCopy_Click, btnCut_Click, btnDelete_Click, btnDeleteAnsw..., btnDownAnsw..., btnFontColor_C..., btnFontDialog..., btnItalic_Click, btnNextQuesti..., btnPaste_Click, btnPreviousQu..., btnRedo_Click, btnSaveAndClo..., btnStrikeout_Cl..., btnSubscript_Cl..., btnSuperscript..., btnUnderline_C..., btnUndo_Click, btnUpAnswer..., cmbDifficult_Se..., cmbShowAnsw..., cmbTypeAnsw..., Dispose, frmNewQuesti..., InitializeComponent, rbxt_TextChang..., ShowQuestion.
- frmTextNumAnswer Class**
 - Поля: btnOk, components, groupBox1, label1, label2, numMark, txtAnswer.
 - Методы: btnCancel_Click, btnOK_Click, Dispose, frmTextNumAnswer, InitializeComponent.
- frmTestEditor Class**
 - Поля: btnAddNewQuestion, btnAddNewQuestion2, btnDeleteQuestion, btnDeleteQuestion2, btnEditQuestion, btnEditQuestion2, btnGoTest, btnGoTest2, btnNewTest, btnNewTest2, btnOpenTestFromDB, btnOpenTestFromDB2, btnOpenTestFromFile, btnOpenTestFromFile2, btnSaveTestToDB, btnSaveTestToDB2, btnSaveTestToFile, btnSaveTestToFile2, btnSection, button1, cmbShowQuestion, columnHeader1, columnHeader2, columnHeader3, columnHeader4, components, exitToolStripMenuItem, fileMenu, ID, label1, label2, lblCountElements, lblWhat, lstVQuestions, menuStrip, numLimit, openFileDialog1, panel1, panel2, panel3, saveFileDialog1, statusStrip, Test, testMenu, toolStrip, toolStripSeparator1, toolStripSeparator2, toolStripSeparator3, toolStripSeparator4, toolStripSeparator6, toolStripStatusLabel, toolTip.
 - Методы: _IsMatch, btnAddNewQuestion2..., btnDeleteQuestion2_Click, btnEditQuestion2_Click, btnGoTest2_Click, btnNewTest_Click, btnOpenTestFromDB_Cli..., btnOpenTestFromFile_Cli..., btnSaveTestToDB_Click, btnSaveTestToFile_Click, btnSection_Click, cmbShowQuestion_Sele..., Dispose, ExitToolsToolStripMenuItem..., frmTestEditor, groupBox2_Enter, HaveTest, InitializeComponent, numLimit_ValueChanged, ShowAll.