

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Институт естественных и точных наук
Факультет математики, механики и компьютерных технологий
Кафедра прикладной математики и программирования
Направление подготовки Программная инженерия

РАБОТА ПРОВЕРЕНА

Рецензент

_____ 2018 г.
« ____ » _____

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой д.ф.-
м.н., доцент

_____ /А.А. Замышляева
« ____ » _____ 2018 г.

Разработка редактора электронного учебного пособия для учебных заведений

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУрГУ – 090404.2018.199. ПЗ ВКР

Руководитель работы

_____ / М.Ю. Сартасова
« ____ » _____ 2018 г.

Автор работы

Студент группы ЕТ-223
_____ /Ф.Р. Хурматуллин
« ____ » _____ 2018 г.

Нормоконтролер, к.э.н.,
доцент кафедры ПМиП

_____ / Д.А. Дроздин
« ____ » _____ 2018 г.

АННОТАЦИЯ

Хурматуллин Ф.Р. Разработка редактора электронного учебного пособия для учебных заведений. – Челябинск: ЮУрГУ, ЕТ-223, 98 с., 27 ил., 6 табл., библиогр. список – 30 наим., 1 прил.

В работе рассмотрена общая структура информационной среды образовательного учреждения, определены направления, которые не автоматизированы или уровень автоматизации мог бы быть улучшенным. Проведен анализ существующих разработок и подходов к автоматизации контроля знаний с использованием компьютерных программ. Разработана модель элемента системы для информационной поддержки процесса контроля знаний. На основе модели, был спроектирован и реализован практически отдельный модуль – элемент информационной системы образовательного учреждения. Проведено тестирование и проверка на работоспособность разработанного модуля.

Оглавление

ВВЕДЕНИЕ	6
----------------	---

1. СИСТЕМА ТЕСТИРОВАНИЯ В ОБРАЗОВАТЕЛЬНОМ УЧЕРЕЖДЕНИИ	8
1.1 Краткая характеристика образовательного учреждения. Выбор направления автоматизации	8
1.2 Классификация информационных технологий в образовании	11
1.3 Тестирование как система проверки и контроля знаний	13
1.4 Обзор программ аналогов	15
1.5 Постановка задачи	21
Выводы по разделу	25
2. РАЗРАБОТКА РЕДАКТОРА ЭЛЕКТРОННОГО УЧЕБНОГО ПОСОБИЯ	24
2.1 Модель разрабатываемого программного комплекса	24
2.2 Физическая схема базы данных	30
2.3 Программное обеспечение задачи	33
2.3.1 Общие положения (дерево функций и сценарий диалога)	33
2.3.2 Описание программных модулей	32
2.4 Характеристика интерфейса, эргономические решения, инструкция пользователя ..	37
Выводы по разделу	45
ЗАКЛЮЧЕНИЕ	44
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	48
ПРИЛОЖЕНИЕ 1. КОД ПРОГРАММЫ	50

ВВЕДЕНИЕ

Роль средств новых информационных технологий в образовании сегодня трудно переоценить. Ускорение научно - технического прогресса, основанное на внедрении в производство гибких автоматизированных систем, микропроцессорных средств и устройств программного управления, роботов и обрабатывающих центров, поставило перед современной педагогической наукой важную задачу - воспитать и подготовить подрастающее поколение, способное активно включиться в качественно новый этап развития современного общества, связанный с информатизацией.

Массовое внедрение инновационных технологий, методик с использованием компьютерных технологий и программ наблюдается как в секторе среднего образования, так и в ВУЗах.

Именно современные информационные технологии сделали возможным доступ каждого специалиста к огромному количеству различных видов информации. В условиях развития национальной системы образования, важное значение приобретает инновационная деятельность общеобразовательных учебных заведений, которая характеризуется системным экспериментированием инноваций в образовательном процессе.

Применение информационных технологий в учебной деятельности особенно необходимо и актуально, поскольку именно следование современным потребностям и своевременное внедрение ИТ способны изменить всю систему в целом, а от их правильности и своевременности зависит эффективность системы образования.

Современные технологии позволяют оптимизировать процесс обмена информацией, уменьшить объем работы участников системы образования, позволяет принимать эффективные управленческие решения и решения связанные с учебным процессом непосредственно.

Объектом исследования являются специальные информационные технологии в образовании, позволяющие улучшить, как протекание

образовательного процесса, так и повысить уровень конечного результата – уровень образования обучаемых. В то же время, как объект стоит рассматривать и образовательное учреждение (ОУ) для которого проектируется программное обеспечение.

Предметом исследования есть сектор программных и программно-технических средств для использования в сфере образовательной деятельности, в частности программы контроля и оценки знаний, как элемент единой информационной сферы ОУ.

Цель проекта – разработать программное обеспечение позволяющее автоматизировать процесс предоставления учебных материалов и проведения контроля знаний, посредством тестирования. Продукт должен быть разработан с учетом универсальности применения. Продукт ориентирован на использование в учебном процессе, пользователями продукта являются:

- преподаватель, как администратор системы;
- студенты (ученики), как пользователи системы.

Задачи проекта:

1. рассмотреть общую структуру информационной среды образовательного учреждения, определить направления, которые не автоматизированы или уровень автоматизации мог бы быть улучшенным;
2. провести анализ существующих разработок и подходов к автоматизации контроля знаний с использованием компьютерных программ;
3. разработать модель элемента системы для информационной поддержки процесса контроля знаний;
4. на основе модели, спроектировать и реализовать практически отдельный модуль – элемент информационной системы образовательного учреждения;
5. провести тестирование и проверить работоспособность разработанного модуля.

1. СИСТЕМА ТЕСТИРОВАНИЯ В ОБРАЗОВАТЕЛЬНОМ УЧЕРЕЖДЕНИИ

1.1 Краткая характеристика образовательного учреждения. Выбор направления автоматизации.

Идеей исследования и практической реализации в работе является подсистема тестирования знаний. Разработка этой системы должна предусматривать должный уровень универсальности, такая система не должна быть привязана к конкретному ВУЗу или другому ОУ. Исходя из этого - нет смысла анализировать конкретный ВУЗ, стоит рассмотреть структуру абстрактного учебного заведения и определить место и роль проектируемой в работе системы.

В качестве предметной области выступает высшее учебное заведение.

ВУЗ это организация, предназначение которой – обеспечить учебный процесс для студентов с целью получения высшего образования в конкретной специальности[3]. ВУЗ это огромный комплекс, который включает в себя множество различных компонентов и бизнес-процессов. С точки зрения тематики данной работы стоит рассмотреть структуру ВУЗа с точки зрения обеспечения учебного процесса и научной деятельности.

Процесс подготовки специалиста является композицией целого класса бизнес процессов организации, в которые вовлечены следующие базовые категории персон и подразделений, принятые в большинстве вузов РФ.

Большинство ВУЗов РФ сегодня придерживаются иерархической структуры (рисунок 1.1)

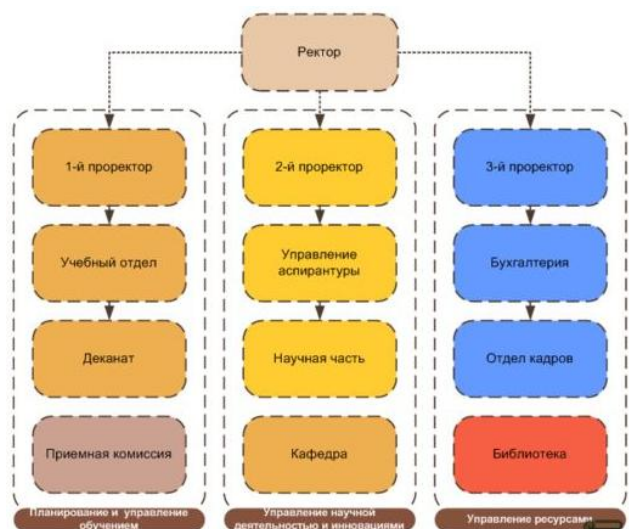


Рисунок 1.1 – Общая иерархическая структура

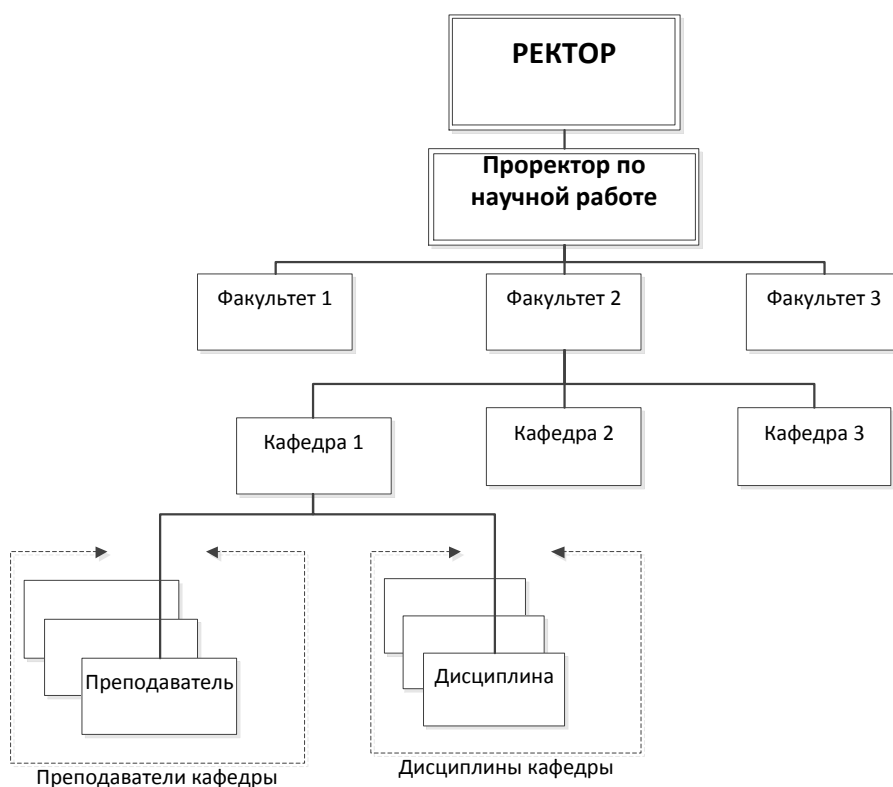


Рисунок 1.2 – Иерархическая организационная структура учебной ветки ВУЗа

Далее определим основные понятия.

Ректор – руководитель учебного заведения ему подчинены прямо или косвенно все структурные подразделения и сотрудники. Ректор подчинен непосредственно министерству образования.

Проректор по научной работе – заместитель ректора в вопросах обеспечения учебной и научной деятельности.

Факультет - учебно-научное и административное структурное подразделение вуза, осуществляющее подготовку студентов и аспирантов по одной или нескольким родственным направлениям подготовки.

Кафедра - подразделение вуза, осуществляющее выполнение учебных курсов в рамках специализаций кафедры, осуществляющее подготовку студентов в рамках своих специализаций. Кафедра структурно относится к определённому факультету. Существуют так называемые обще факультетные кафедры, например - в технических вузах это могут быть кафедры, обеспечивающие социальные дисциплины (социология, философия, культура и т.д.). Такие кафедры обеспечивают дисциплины на разных кафедрах.

Основными клиентами (потребителями образовательной услуги) являются студенты, студентом становится выпускник средней школы, пройдя этап «Абитуриент»

Абитуриенты - персоны участвующие в конкурсе для поступления в вуз.

Студенты - персоны, проходящие обучения по некоторому направлению подготовки, осуществляемом в рамках факультета вуза.

Профессорско-преподавательский состав (ППС) - преподаватели, непосредственно проводящие учебные курсы в рамках направлений подготовки, а также осуществляющие промежуточную и итоговую аттестацию студентов. Преподаватели могут являться сотрудниками кафедр, силами которой выполняется некоторое множество учебных курсов в рамках специализации кафедр.

Даже такой беглый анализ дает возможность разделить исследуемый объект на: организация (глобально ВУЗ), составные структурные элементы организации, персоны (акторы) – люди, принимающие непосредственное активное участие в процессе.

Для определения места проектируемой системы рассмотрим структурно-функциональную модель «Предоставление образовательных услуг»



Рисунок 1.3 – Контекстная диаграмма IDEF0

Диаграмма наглядно иллюстрирует предпосылки и механизмы предоставления образовательной услуги.

Предпосылки (входные данные и управляющие документы) достаточно подробно описаны на схеме и пояснений не требуют. Результаты тоже достаточно очевидны.

Механизмы: сюда относится в первую очередь профессорско-преподавательский состав (ППС) как главный механизм обеспечения образовательного процесса. Естественно стоит отдельно вынести материальную базу (здания, мебель, компьютеры и техника и т.д). Кроме преподавателей деятельность обеспечивает и ряд других подразделений: юристы, документоводы, секретари, технический персонал.

Задача «обеспечение образовательного процесса» сформулирована слишком широко и объем работы не позволяет разработать и описать полную модель. Поэтому ограничимся некоторыми выбранными областями.

Для детализации процесса используется диаграмма декомпозиции. Так диаграмма декомпозиции предоставления образовательных услуг может выглядеть следующим образом (рисунок 1.4)

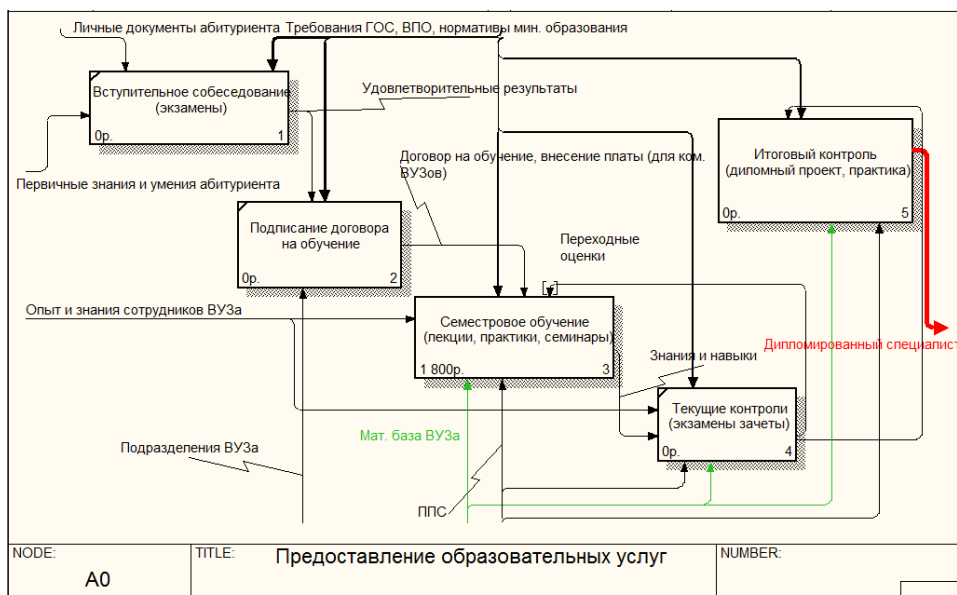


Рисунок 1.4 – Декомпозиция первого уровня

Уже на этой диаграмме декомпозиции виден процесс «Текущие контроли». Именно этот процесс и подлежит автоматизации посредством использования проектируемой системы

Обучение в семестре (возможны и другие разделения: триместры, это не влияет кардинально на протекание процессов) можно в свою очередь детализировать следующей диаграммой (рисунок 1.5)



Рисунок 1.5 – Декомпозиция второго уровня

Система тестирования с дополнением теоритическими материалами, для обучения по конкретной тематике, может частично совместить и процесс «Выполнение заданий» и процесс «Контроль и оценка знаний». Частично

потому, что нельзя строить задания только на тестировании. Задания должны быть комплексными и включать различные виды работ, среди которых тестирование также уместно[8]. Система имеет преимущества в оценивании, так как оценивание осуществляется программой и является абсолютно объективным, полностью, исключая субъективный фактор и влияние отношений между «Студент» - «Преподаватель».

При создании системы с наперед продуманным уровнем универсальности может использоваться не только для тестирования студентов. Так, например, многие ВУЗы на своей базе проводят различные курсы переподготовки, повышения квалификации, курсы от служб занятости и других организаций. Разрабатываемая система должна предполагать и такую переориентацию. Это существенно расширит спектр применения системы. Как пример можно рассматривать тестирование в направлении «Охрана труда и безопасность жизнедеятельности»

1.2 Классификация информационных технологий в образовании

Внедрение информационно-коммуникационных технологий в информационное обеспечение образовательных процессов осуществляется эпизодично и недостаточно. Постепенно формируется рынок таких систем, где предлагаются как их коммерческие версии, созданные независимыми производителями, так и системы, разработанные непосредственно в подразделениях образовательной сферы, где эти средства и эксплуатируются, ведь научно - технический потенциал сотрудников образовательных учреждений, особенно, если это учебное заведение технического профиля, вполне позволяет разработать действующую и всеобъемлющую систему автоматизации учебного процессом самостоятельно. В зависимости от способа создания, подобные системы имеют ряд характерных особенностей, как положительных, так и отрицательных. Так, к недостаткам информационных систем, созданных независимыми разработчиками, можно

отнести их чрезмерную коммерциализацию, избыточную универсальность, сложность настройки, громоздкий и непонятный интерфейс. Как положительные признаки можно отметить хорошую документированность и сервисную поддержку, широкий набор функциональных возможностей. Информационные системы индивидуальной разработки, как правило, менее устойчивы, больше зависят от персонального сопровождения своими разработчиками, хуже документированы. Однако системы, разработанные непосредственно силами сотрудников учебного заведения, точнее соответствуют его специфическим условиям и требованиям, гибко и оперативно реагируют на изменения в законодательстве и общей экономической ситуации, они удобнее для использования персоналом определенного конкретного учебного заведения. Из этого следует, что подобные системы нужно обязательно разрабатывать сборной командой и непременно привлекать в процесс разработки специалистов из образовательной сферы, очень желательно, чтобы эти специалисты были практикующими учителями, преподавателями, менеджерами образовательных учреждений.

Информационное обеспечение образовательных процессов, построенное на создании информационных систем дает возможность снизить трудности в обработке информации и расширить сферу ее использования. При этом каждая информационная система представляется как совокупность обеспечивающих подсистем: информационной, организационной, правовой и программной.

Для классификации ИС и ПП обеспечения образовательных процессов можно предложить ряд разнообразных критериев, в своем исследовании предложим две классификации - по критерию глобальности, ширины охвата задач (рисунок 1.6) и по критерию функциональной направленности (рисунок 1.7).

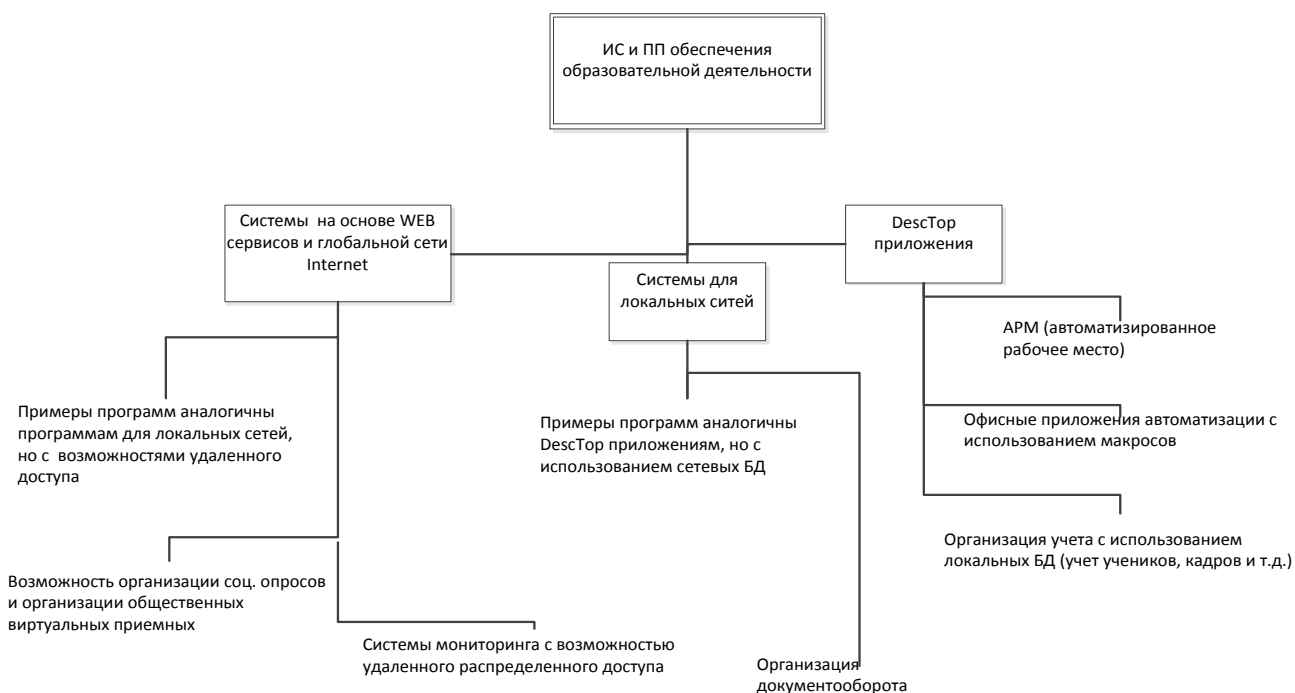


Рисунок 1.6 – Классификация систем по критерию глобальности (уровня проектирования)

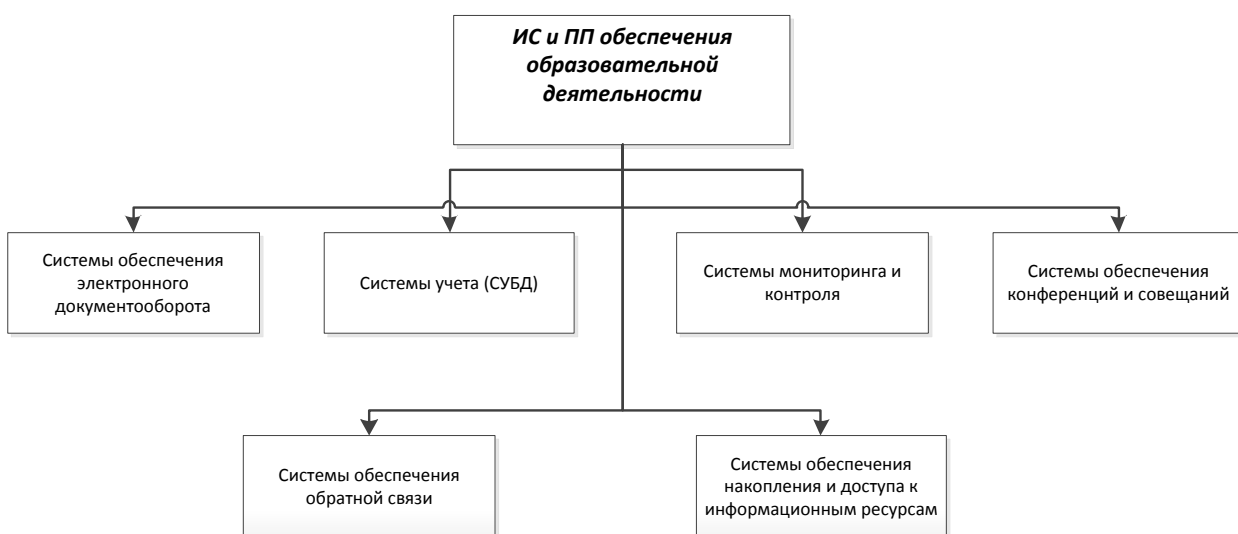


Рисунок 1.7 – Классификация систем по функционалу

1.3 Тестирование как система проверки и контроля знаний

Тестирование широко применяется для оценки уровня знаний в учебных заведениях, при приеме на работу, для оценки квалификации персонала учреждений, т.е. практически во всех сферах деятельности

человека. Испытуемому предлагается ряд вопросов (тест), на которые он должен ответить.

Обычно к каждому вопросу дается несколько вариантов ответа, из которых надо выбрать правильный. Каждому варианту ответа соответствует некоторая оценка. Суммированием оценок за ответы выходит общий балл, на основе которого делается вывод об уровне подготовленности испытуемого.

Основное отличие теста заключается в том, что он всегда предполагает измерение. Оценка, выставляемая по итогам тестирования, отличается большей объективностью и независимостью от возможного субъективизма того кто проводит собеседование или тестирование устно. Главная отличительная черта теста - объективность, гарантируемая измерением, функция которого заключается в том, чтобы давать количественную (бальную) оценку[8].

Максимально повысить надежность измерения и валидность использования теста можно, если следовать трем основным этапам его создания:

- дать ясное и недвусмысленное теоретическое определение умений, которые надо проверить;
- точно установить условие и операции, которые должны соблюдаться при проведении теста;
- количественно определить результаты наблюдений, с тем, чтобы убедиться, что используемые мерные шкалы обладают всеми необходимыми качествами.

Под компьютерным тестом будем понимать реализованную с помощью персонального компьютера систему взаимосвязанных предметным содержанием заданий стандартной формы, имеющие определенную шкалу значений, выполнение которых обнаруживает наличие знаний, умений и навыков и уровня подготовки специалиста. Формирование умений и навыков целесообразно проверять, учитывая такие взаимодополняющие критерии [15]:

1. правильность: результат любой деятельности соответствует цели выполнения;
2. осмысленность: готовность объяснить каждую из применяемых операций;
3. осознанность: ориентация на существенные связи и отношения, заданные в условии при осуществлении деятельности;
4. рациональность: выбор такого способа выполнения задач, который быстрее (оптимально) приведет к цели;
5. обобщение: способность студента выполнять необходимые действия в вариативных условиях;
6. абстрактность: возможность описания оперативного состава выполненного действия и последовательности выполнения операций, входящих в нее, воспроизведения необходимого правила;

1.4 Обзор программ аналогов

На рынке предлагается несколько подобных (разрабатываемому продукту) программ, и в этом пункте представим обзор некоторых из них.

SunRav TestOfficePro

Разработчик: SunRav Software (<http://www.sunrav.ru>)

Способ распространения: shareware

Цена: 5900 руб. (для всех образовательных учреждений предоставляется 20% скидка, то есть цена для них - 4720 руб).

В пакет SunRav TestOfficePro входят программы для создания тестов, проведения тестирования и обработки результатов тестирования. С его помощью можно организовать и провести тестирования и экзамены в образовательных учреждениях (вузы, колледжи, школы), а предприятия и организации могут осуществлять аттестацию и сертификацию своих сотрудников.

Пакет позволяет создавать тесты по предметам школьной программы, вузовским дисциплинам, тесты для профессионального тестирования, психологические тесты и т.д.

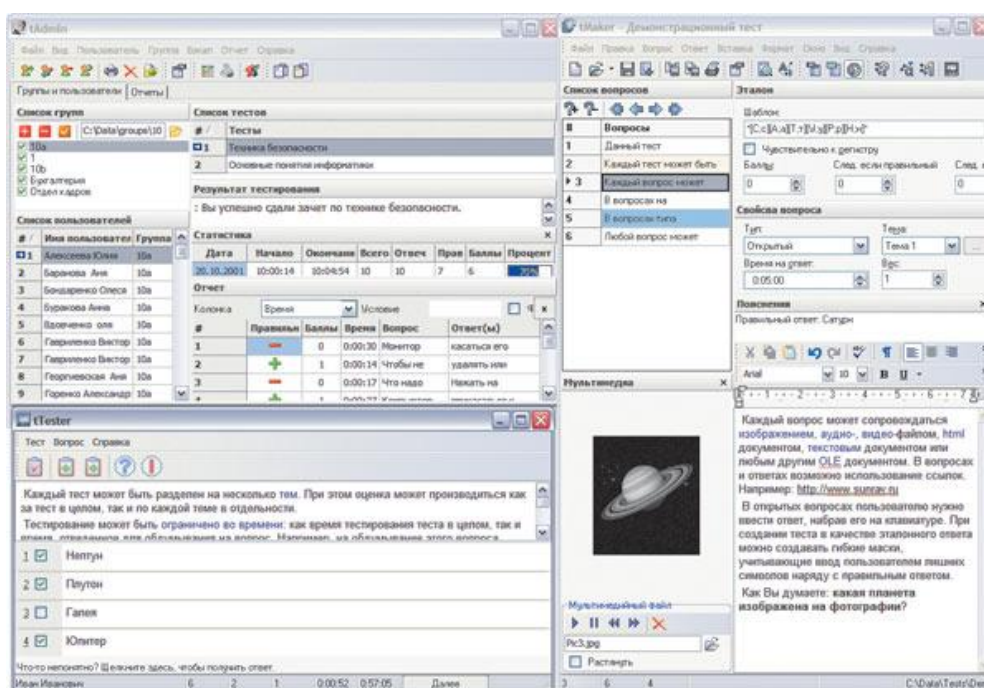


Рисунок 1.8 – Система тестирования **SunRay TestOfficePro**

Все тесты и результаты тестирования шифруются методами стойкой криптографии, что исключает возможность подделки результатов. Кроме того, на тест можно установить пароли: на редактирование - защищает тест от просмотра его структуры, правильных ответов и т.д.; на просмотр - предотвращает пробное тестирование с целью выяснения правильных ответов.

Вопросы и варианты ответа можно форматировать, используя для этого встроенный текстовый редактор, близкий по своим функциям к MS WORD. В редакторе можно вставлять изображения, формулы, схемы, таблицы, аудио- и видеофайлы, HTML-документы и любые OLE-документы.

В тестах возможно использование пяти вышеперечисленных типов вопросов. Тест может быть разделен на несколько тем. При этом возможно оценивать знания тестируемого как по каждой теме в отдельности, так и по тесту в целом.

Конструктор тестов Keepsoft

Разработчик: Keepsoft

Сайт программы: <http://www.keepsoft.ru/simulator.htm>

Способ распространения: shareware

Цена: корпоративная лицензия на неограниченное число компьютеров - 3000 руб (существует ряд других типов лицензий с ограничением по количеству ПК).

«Конструктор тестов» - это универсальная программа для проверки знаний. Приложение можно применять для проведения тестирования дома и в учебных заведениях. Программа позволяет использовать неограниченное количество тем, вопросов и ответов.

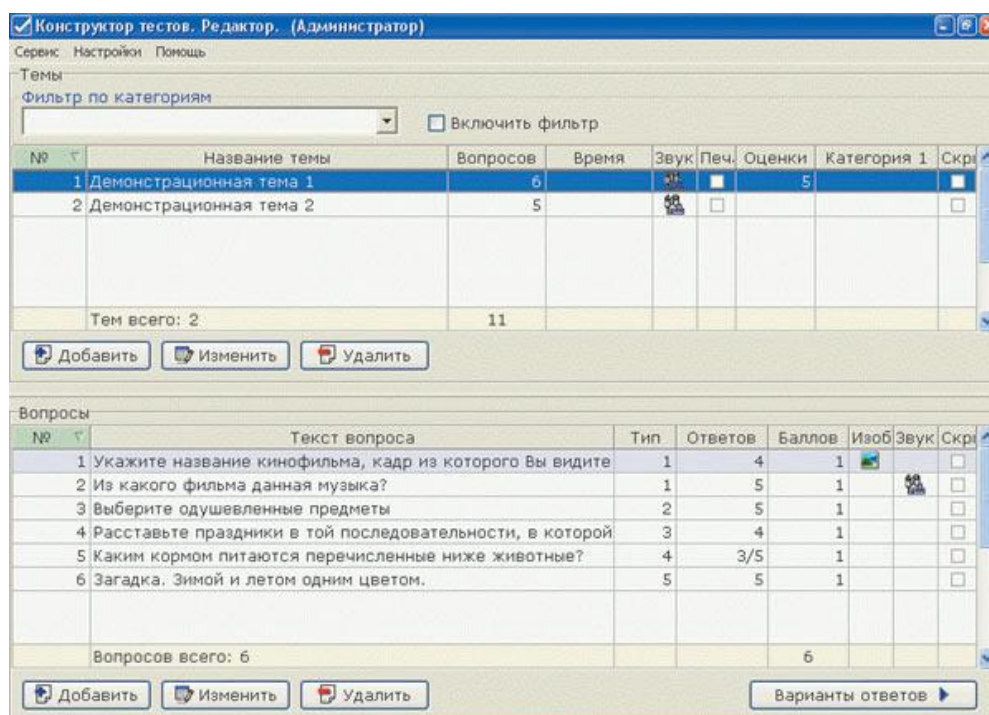


Рисунок 1.9 – Система тестирования **Конструктор тестов Keepsoft**

Возможности программы:

1. вопросы могут содержать музыку (файлы WAV, MID, RMI), изображения (файлы JPG, BMP, ICO, EMF, WMF), видеоролики (файлы AVI);
2. поддерживаются вопросы всех пяти вышеперечисленных типов;
3. печать на принтере и сохранение в файл тем, вопросов и ответов;

4. печать на принтере и сохранение в файл результатов тестирования;

5. экспорт тем, вопросов и ответов в файлы различных форматов (MS Excel, MS Word, MS Access, Paradox, DBase, текстовый файл, HTML, XML, RTF (RichText format), PDF (Adobe Acrobat), буфер обмена MS Windows, Lotus 1-2-3 и др.);

6. тестирование на одном компьютере нескольких пользователей.

Для каждого пользователя заводится персональная карточка пользователя, поля в которой являются настраиваемыми.

Дополнительные возможности:

1. установка различных прав доступа к редактированию базы данных в «Редакторе» для разных пользователей;

2. задание вопросов в произвольном порядке;

3. задание цены каждому вопросу в баллах;

4. ограничение времени на ответ;

5. возможность прервать тестирование и продолжить его в другое время;

6. выставление оценки по окончании тестирования. Система оценок настраивается в «Редакторе». Шкалу оценок можно настроить от 2- до 100-балльной системы;

7. синхронизация базы данных; при помощи этой функции можно легко обмениваться данными с другими пользователями и переносить данные с компьютера на компьютер;

Существует и множество других систем. Все системы отличаются функционалом, требованиями к ПК. Некоторые системы требуют наличие активного соединения с Интернет. Недостатком готовых систем является платная основа распространения (для большинства полнофункциональных систем), излишняя функциональность и сложность администрирования. Но основной проблемой является отсутствие возможности расширения функций и опций системы, а также, интеграция системы тестирования в общую

информационную среду. Эти проблемы позволяет решить собственная разработка.

1.5 Постановка задачи

Целью данного проекта и его практической составляющей является разработка автоматизированной информационной системы обучения и контроля знаний студентов.

Данная программа должна выполнять следующие функции:

- АИС должна работать в трех режимах:
- администратор – настройка системы, тестов;
- обучение (ознакомление с материалами в виде отдельных документов по тематикам);
- прохождение контрольных вопросов (тестирование).
- регистрация в информационной базе тестируемых;
- каждый пользователь может видеть только свои отчеты, сформированные по результатам прохождения тестов;
- в режиме обучения предусмотреть показ текста лекций и лабораторных практикумов в формате word (rtf и doc);
- предусмотреть несколько разнонаправленных тем по предмету при прохождении итоговых вопросов;
- время тестирования должно быть управляемым с помощью ручной корректировки;
- вести историю тестирования пользователей с отображением детализированного результата тестирования выбранного тестируемого;
- серверная часть должна хранить базу данных пользователей, имеющих доступ к системе и обеспечивать аутентификацию;
- удобный и гибкий поиск необходимой информации в базе данных;

- экспорт отчета и данных в программы Microsoft Word и Microsoft Excel;
- для доступа к данным из приложений ИС использовать язык запросов SQL

Основные характеристики программы:

1. Программа является сетевой и работает с БД Microsoft Access 2010 и выше.
2. Обеспечение поддержки клиент серверной технологии возлагается на программную часть: сервер обеспечивает доступ к БД и выборку данных, клиент – отображение информации на экран и фиксацию результатов тестирования
3. В программе предусмотрена работа с разделением прав, то есть у администратора все права, у обычного пользователя - ограниченные права - ввода в базу данных не имеет.
4. При создании нового пользователя система автоматически присваивает ему логин = фамилия, пароль = имя.
5. При добавлении в базу новых данных осуществляется контроль на дублируемость - есть ли уже данные в БД.
6. Программа удобная и гибкая в обращении - действия можно выбирать не только с помощью кнопок в окнах программы, но и с помощью правой кнопки мыши, а также осуществлять выбор и ввод с помощью двойного щелчка мыши левой кнопкой или клавиша « Enter» в поле ввода.
7. Программа предназначена для контроля знаний по выбранной тематике и учету полученных результатов.
8. В программе имеется большое количество отчетов по разным критериям: по пользователям и т.д.
9. Программа предполагает добавление материалов по темам: лекции и лабораторные практикумы в формате rtf, doc или pdf

10. Чтобы студенты смогли увидеть актуальное содержимое лекции на сегодняшний день - нужно установить в настройках с помощью удобного интерфейса тему сегодняшней лекции для занятия - делает преподаватель. Также у преподавателя имеется возможность установить активный тест, студент (тестируемый) не может изменить тест, который ему предлагается.

11. Удобный режим тестирования - после прохождения теста выводится протокол с результатом и количеством набранных баллов.

12. В программе должен быть предусмотрен удобный интерфейс поиска пользователей и т.д. по различным критериям.

13. Программа должна быть снабжена справочной информацией о разработчике, заказчике и программе, причем некоторые из них в удобном формате `html`.

В практической части необходимо реализовать универсальную подсистему для автоматизации проведения тестирования студентов.

Система не должна иметь непосредственную привязку к конкретному материалу, должна предусматривать возможность проектирования различных тестов с различными наборами вопросов и структурой.

Система должна иметь возможность работать в сети, с авторизованным доступом сотрудника, и отдельным приложением для администрирования системы.

Программный комплекс должен предусматривать реализацию двух составляющих:

Клиент и Администратор (Администратор включает также и функционал сервера БД)

Комплекс реализуется на основе клиент-серверной архитектуры. При условии работающего сервера клиенты могут проходить тестирование в локальной сети.

1) Сервер - связан с базой данных, и содержит основные программные средства для решения поставленных задач, точнее именно в этой программе

реализованы все классы, которые отвечают за обработку и передачу данных между другими составляющими системы.

2) Клиент - программа предназначена для организации диалога с пользователем. Работа с сервером и БД организована через интерфейс классов сервера, с помощью WCF передает необходимую информацию на сервер.

Клиент предусматривает:

1 Авторизация входа в систему
2 Просмотр документов и материалов по тематике (это могут быть учебные материалы или тесты для ознакомления)

3 Пройти тестирование

4 Посмотреть свои результаты

Администраторская часть предусматривает:

1. Добавление студентов и сотрудников (которые подлежат тестированию, и те, которые отвечают за процесс тестирования (условно админы) разделены на разные категории)

2. Добавление материалов для ознакомления

3. Создание тестов

Тесты могут включать вопросы трех типов:

ТВ – ТО; ВИ – ТО; ТВ - ОИ

ТВ - текстовый вопрос

ТО - текстовый ответ

ВИ - вопрос-изображения

ОИ - ответ изображения (выбор картинки)

Тесты группируются по направлениям (дисциплинам) и характеризуются конкретной тематикой

4. Просмотр результатов тестирования с возможностью фильтрации и поиска по различным критериям

Выводы по разделу

В этой главе была рассмотрена структура образовательного учреждения, так как идеей исследования и практической реализацией в работе является система тестирования знаний, причем ее можно использовать не только для тестирования студентов, но и для повышения квалификации сотрудников, различные курсы переподготовки и т.д.

Главная отличительная черта тестирования - объективность, гарантируемая измерением, функция которого заключается в том, чтобы давать количественную (бальную) оценку.

Постановка задачи и основные требования были сформулированы на основании проведенного анализа предметной области и при рассмотрении программ аналогов.

2. РАЗРАБОТКА РЕДАКТОРА ЭЛЕКТРОННОГО УЧЕБНОГО ПОСОБИЯ

2.1 Модель разрабатываемого программного комплекса

На сегодняшний день существует много технологий для реализации систем с архитектурой клиент-сервер, но среди этих технологий отличается Windows Communication Foundation (WCF). WCF – имя платформы следующего поколения, для сетевых систем с использованием расширения API .NET, WCF делает возможным построение безопасных, надежных и транзакционных систем через упрощенную унифицированную программную модель межплатформенного взаимодействия. При использовании этой системы в комплексе вместе с базами данных, возникает большое количество разнообразных возможностей для разработчика программного обеспечения.

Используя WCF и язык C#, разработана межсетевая система, основной задачей которой является организация прохождения тестирования студентов. Программа разделена на две части – клиент и сервер.

1) Сервер – связан с базой данных, и содержит основные программные средства для решения поставленных задач, точнее именно в этой программе реализованы все классы, которые отвечают за обработку и передачу данных между другими составляющими системы.

2) Клиент – программа предназначена для организации диалога с пользователем. Работа с сервером и БД организована через интерфейс классов сервера, который с помощью WCF передает необходимую информацию на сервер.

Благодаря технологиям WCF, OleDb, языка C # и базы данных Access можно разрабатывать многофункциональные системы автоматического управления, ориентированные на работу в сети, что значительно упрощает как процесс разработки программного обеспечения, так и внедрение программных комплексов в эксплуатацию.

Для быстрой и эффективной разработки программного продукта необходимо привлечь квалифицированную рабочую силу, выбрать правильные инструменты и определить правильное направление работы. Процесс разработки проекта необходимо хорошо продумать, чтобы быстро адаптировать его к возможным изменениям требований пользователей, потребностей бизнеса или технологии. Этого можно достичь, если в процессе разработки проекта использовать модели.

Бесспорно, разработка современных программных продуктов невозможна без предварительного моделирования. Накопленный опыт свидетельствует: чем больше и сложнее проект, тем важнее становится моделирование будущей системы. Не стоит надеяться на успешность проекта, если не уделено достаточного внимания предварительному моделированию системы.

Моделирование используют не только при создании больших систем. Ведь чем больше и сложнее система, тем большее значение приобретает моделирование при ее разработке. Дело в том, что моделировать сложную систему необходимо в любом случае, поскольку в противном случае мы не сможем ее представить как единое целое. Модели наглядно демонстрируют желаемую структуру и поведение системы, отражают ее архитектуру и помогают уточнить детали проекта с заказчиком для минимизации будущих рисков.

На начальных стадиях моделирования статических аспектов системы, можно использовать диаграмму прецедентов, которую применяют двумя способами:

- для моделирования контекста системы: условно отделяют систему и проявляют действующие лица, которые находятся за этим пределом и взаимодействуют с системой (диаграммы прецедентов необходимы на этом этапе для идентификации актеров и семантики ролей);
- для моделирования требований к системе: определяют то, что система должна делать с позиции внешнего пользователя, независимо от

того, как она должна это делать (диаграммы прецедентов необходимы на этом этапе для идентификации желаемого поведения системы, которая служит здесь «черным ящиком»).

Для разрабатываемой системы тестирования, диаграмма прецедентов приведена на рисунках 2.1-2.2.

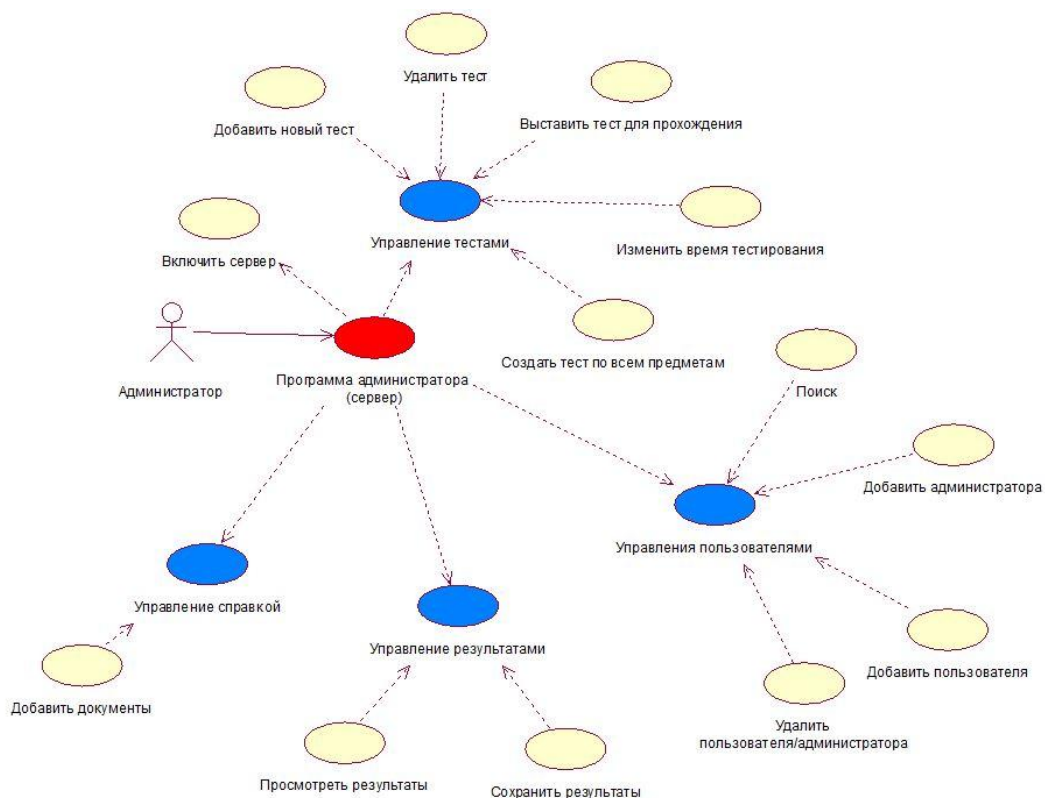


Рисунок 2.1 – Диаграмма прецедентов для Администратора системы

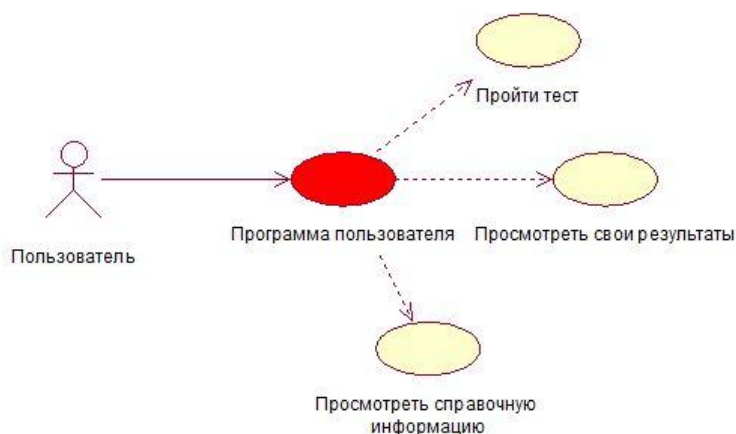


Рисунок 2.2 – Диаграмма прецедентов для пользователя

«Пользователь» и «Администратор» являются актерами модели (объекты, которые непосредственно взаимодействуют с системой).

Прецедентами же являются возможности моделируемой системы (часть её функциональности), благодаря которой актер может получить конкретный, измеримый и нужный ему результат.

Так как разработанная программа является клиент-серверным приложением, модель включает: клиент, сервер и базу данных.

Более полное представление о программном продукте с точки зрения взаимодействия его компонентов между собой и с внешней средой дает функциональная схема.

Функциональная схема - взаимодействия компонентов программного обеспечения с описанием информационных потоков, состав данных в потоках с указанием используемых файлов и устройств.

Функциональная схема программы тестирования показана на рисунке 2.3

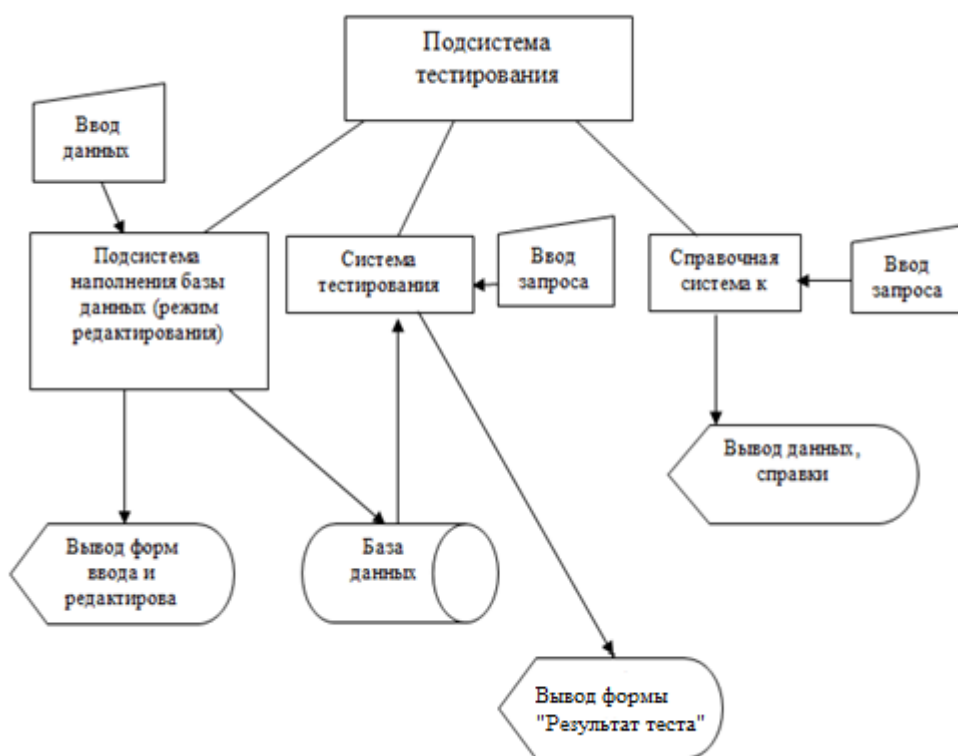


Рисунок 2.3 – Функциональная схема проекта

2.2 Физическая схема базы данных

Для разрабатываемой системы сущностями физической модели данных являются: «Вопросы», «Группы», «Дисциплина», «Пользователи», «ПростойОтчет», «Результат», «Справочники», «Тема», «Тесты», «Филиал».

Для моделирования данных можно применять различные методологии и нотации. В нашем случае использована модель сущность-связь и программное средство Visio для визуального построения модели.

Модель представлена на рисунке 2.4, РК – (primary key) первичный ключ, FK – (foreign key) внешний ключ.

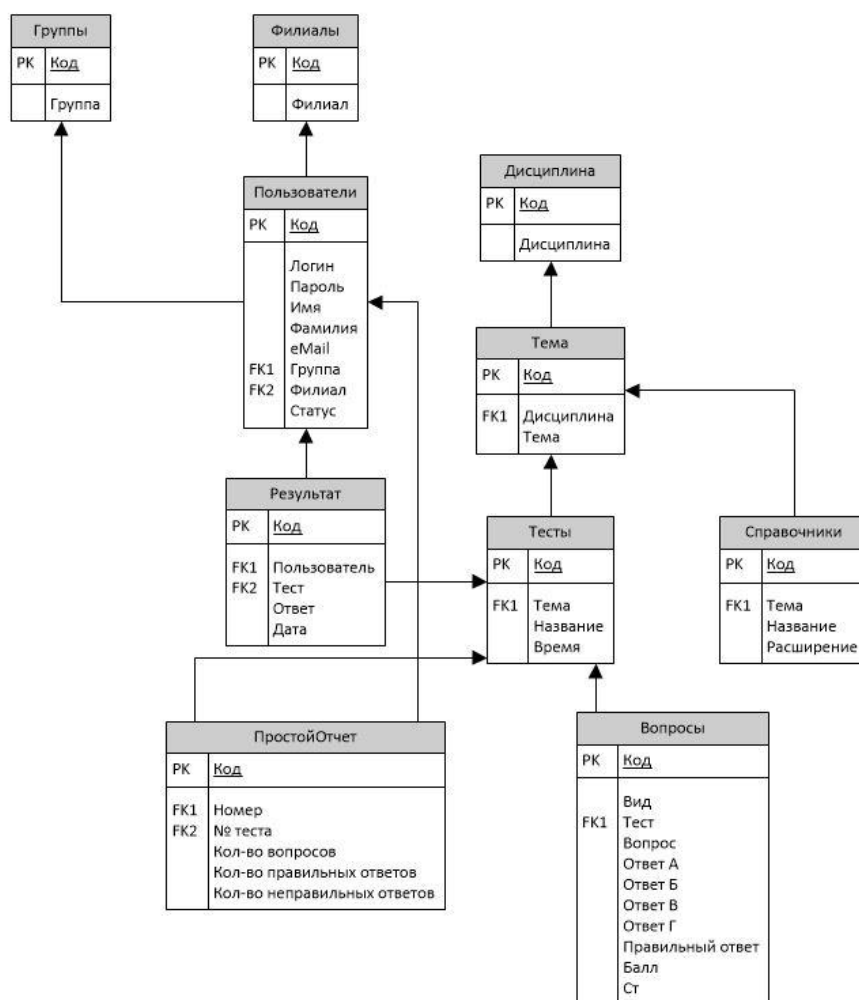


Рисунок 2.4 – Логическая схема базы данных

Опишем сущности (таблицы) представленные в модели:

Таблица 2.1– Сущность «Вопросы»

Сущность	Тип
Код	Счетчик
Вид	Текстовый
Тест	Числовой
Вопрос	Поле МЕМО
Ответ А	Поле МЕМО
Ответ Б	Поле МЕМО
Ответ В	Поле МЕМО
Ответ Г	Поле МЕМО
Правильный ответ	Текстовый
Балл	Числовой

Сущность «Вопросы» хранит основную информацию, оставляющую каждый отдельный вопрос. Как видно, все вопросы являются стандартизированными (имеют единый формат).

Таблица 2.2– Сущность «Группы»

Код	Счетчик
Группа	Текстовый

Сущность «Группы» хранит наименования групп присутствующих в заведении, где пользователи проходят тестирование.

Таблица 2.3 – Сущность «Дисциплина»

Код	Счетчик
Дисциплина	Текстовый

Сущность «Дисциплина» хранит наименование дисциплин, по которым может проходить тестирование.

Таблица 2.4– Сущность «Пользователи»

Код	Счетчик
Логин	Текстовый
Пароль	Текстовый
Имя	Текстовый
Фамилия	Текстовый
eMail	Текстовый
Группа	Числовой
Филиал	Числовой
Статус	Логический

Сущность «Пользователи» содержит основную информацию связанную с персонами проходившими тестирование. Без внесения этих данных, регистрация и работа пользователя с программой не будет являться возможной.

Таблица 2.5– Сущность «ПростойОтчет»

Код	Счетчик
Номер	Числовой
№ теста	Числовой
Кол-во вопросов	Текстовый
Кол-во правильных ответов	Текстовый
Кол-во неправильных ответов	Текстовый

Сущность «ПростойОтчет» содержит первичную информацию, которая формируется после завершения пользователем тестирования. Эта информация является результатом тестирования и выводится пользователю непосредственно после его прохождения.

Сущность «Результат» содержит информацию о каждом конкретном пройденном вопросе пользователя по тесту. Информация заносится после прохождения всего блока вопросов.

Сущность «Справочники» содержит информацию о справочной документации доступной пользователям.

Сущность «Тема» хранит информацию о возможных темах тестирования. Каждая новая тема должна быть связанная с уже внесенной в программу дисциплиной.

Сущность «Тесты» хранит список доступных тестов для прохождения.

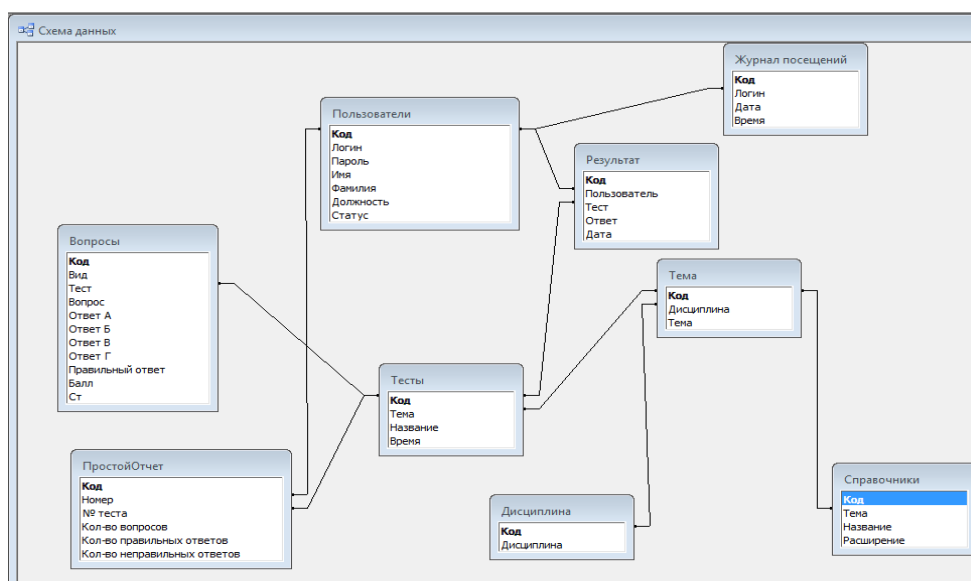


Рисунок 2.5 – Схема данных

2.3 Программное обеспечение задачи

2.3.1 Общие положения (дерево функций и сценарий диалога)

Взаимодействие с программой происходит через графический интерфейс, а именно диалоговые окна Windows Form. Технология Windows Form разработана компанией Microsoft и поддерживает все стандартны построения пользовательских визуальных интерфейсов, для платформы Windows.

Windows Forms позволяет разрабатывать интеллектуальные клиенты. Интеллектуальный клиент – это приложение с богатым графическим интерфейсом, простое в развертывании и обновлении, способное работать при наличии или отсутствии подключения к Интернету и использующее более безопасный доступ к ресурсам на локальном компьютере по сравнению с традиционными приложениями Windows

Для разработки структуры сценария диалога необходимо выбрать язык общения. При разработке системы было взято во внимание, что данный продукт будет использоваться на территории РФ, поэтому, система изначально локализована, это поможет легко освоить пользователю данный программный продукт.

В связи с тем, что при решении задачи используется технология обработки информации в режиме диалога, взаимодействие пользователя с программой можно представить в виде схемы диалога (Рисунок 2.6).

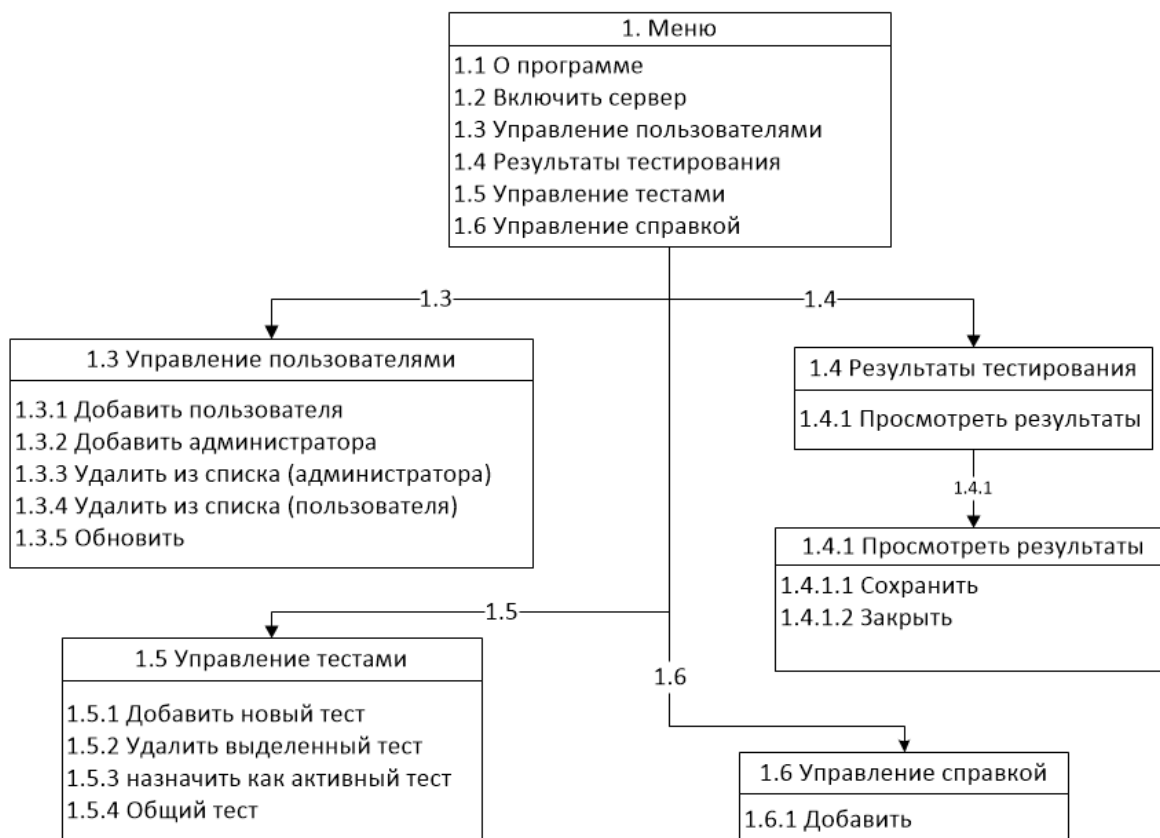


Рисунок 2.6 – Дерево диалогов

2.3.2 Описание программных модулей

Технология WCF позволяет разместить все основные модули программного обеспечения в серверной его части и предоставлять доступ к ним через интерфейсы. Поэтому в данной главе опишем основные модули реализующие полноту функционала разработанного программного комплекса.

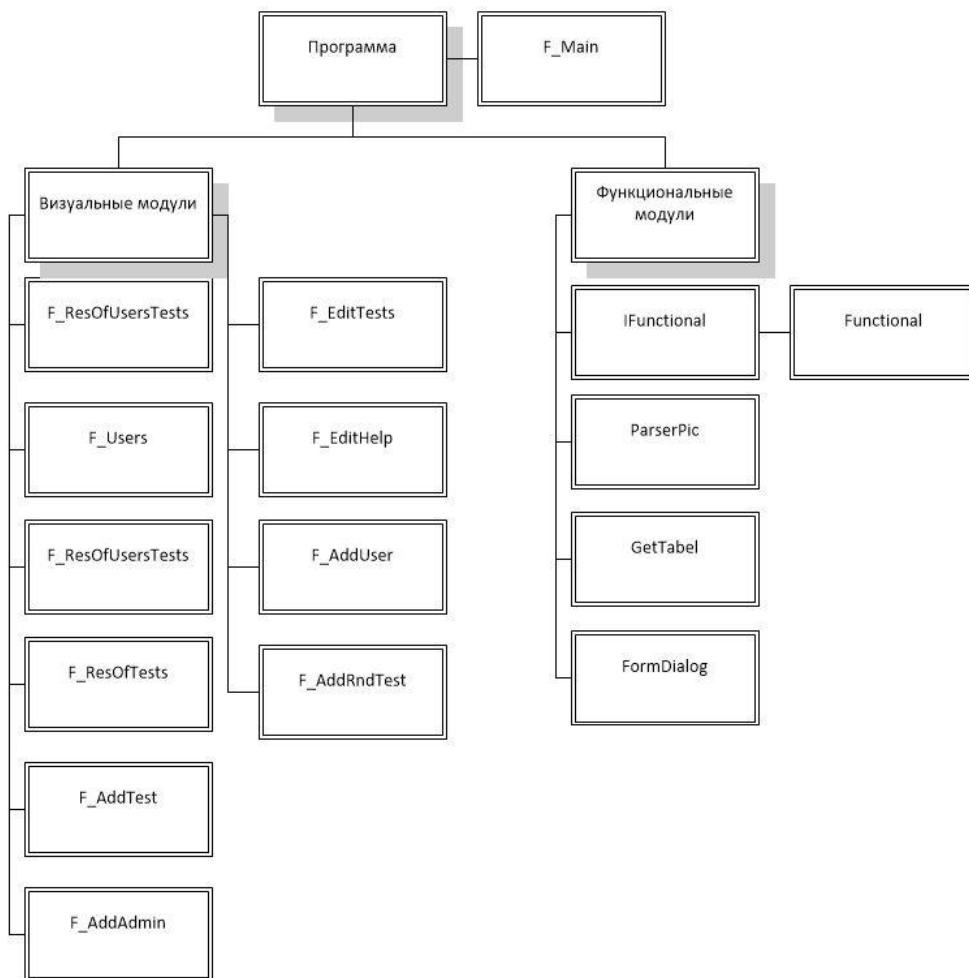


Рисунок 2.7 – Дерево модулей

Таблица 2.6 – Описание модулей

Название	Описание
F_Main	Главное окно программы
Функциональные модули	
IFunctional	Интерфейс методов для клиентской части программы
Functional	Набор методов по работе с программой
ParserPic	Класс, отвечающий за парсинг изображений в XML форму для тестов
GetTabel	Класс, имеющий методы для связи с БД
FormDialog	Хранит глобальные переменные
Визуальные модули	
F_ResOfUsersTests	Форма отображения детальной информации по тесту пользователя
F_Users	Форма вывода списка пользователей и предоставляющая методы по их редактированию
F_ResOfTests	Форма, отображающая пользователей и их тесты. Так же предоставляем методы по поиску и фильтрации данных
F_AddTest	Форма добавления новых тестов
F_AddAdmin	Форма добавления администраторов
F_EditTests	Форма редактирования тестов
F_EditHelp	Форма добавления справочной информации
F_AddRndTest	Форма настройки общего теста
F_AddUser	Форма добавления пользователей

Ниже приведем примеры реализации некоторых интерфейсов и механизмов клиентского и серверного приложений.

Листинг 2.1 – Пример кода взаимодействия системы с базой данных

```

static class GetTable
{
    static OleDbCommand OleDbComm = new OleDbCommand();
    static OleDbConnection OleDbConn = new OleDbConnection();
    static OleDbDataAdapter OleDbAdapter = new OleDbDataAdapter();
    // Устанавливает соединение с базой данных
    static public DataTable Get(string query)
    {
        OleDbConn.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=DBDate.accdb;Persist Security Info=True";
        OleDbComm.Connection = OleDbConn;

        DataTable MainDataTable = new DataTable();
        OleDbAdapter.SelectCommand = new OleDbCommand(query, OleDbConn);
        OleDbAdapter.Fill(dt);
        try
        {
            OleDbConn.Open();
        }
    }
}

```

```

        Catch {}
        OleDbConn.Close();
        return MainDataTable;
    }
}

```

Данный метод, с помощью библиотеки oledb осуществляет доступ приложения к базе данных.

Теперь можно привести некоторые основные методы и запросы, которые использует программа в ходе работы.

Листинг 2.2 – Метод регистрации пользователей

```

public Boolean Registration(String Pass, String Name, String Surname, String
eMail, Int32 Group, Int32 Branch)
{
    try
    {
        GetTabele.Get("INSERT INTO Пользователи (Логин, Пароль, Имя, Фамилия,
eMail, Группа, Филиал) VALUES ('" +
Surname + Name + "', '" + Pass + "', '" + Name + "', '" + Surname + "', '"
+ eMail + "', " + Group + ", " + Branch + ")");
        return true;
    }
    catch { return false; }
}

```

Как видно из метода, с помощью оператора INSERT, языка SQL, в таблицу «Пользователи» вносятся регистрационные данные пользователя: логин, пароль, имя, фамилия, eMail, группа и филиал.

Разработанная система состоит из клиент-серверного приложения и базы данных. База данных обеспечивает информационную часть системы, то есть, все данные которыми система оперирует, хранится непосредственно там.

Клиентское и серверное приложения, написанные на языке программирования C#, с использованием графического интерфейса Windows Form.

Серверная часть включает в себя основные механизмы работы системы. Содержит в себе основные функции по обработке данных и предоставления функциональных интерфейсов для клиентской части программы.

Взаимодействие с пользователем, а точнее непосредственно основную функциональную часть, заложенную в системе – прохождение тестов, осуществляется клиентским приложением, которое с помощью интерфейсов, построенных по технологии WCF, осуществляет связь с Сервером.

Листинг 2.3 – Метод авторизации

```
public DataTable GetUser(String Login, String Pass)
{
    DataTable _dt = new DataTable();
    _dt = (GetTabele.Get("SELECT Код, Логин FROM Пользователи WHERE Логин='"+
Login + "' AND Пароль='" + Pass + "'" ));
    DataTable dt = new DataTable("Пользователь");
    dt.Columns.AddRange(new DataColumn[2] { new DataColumn("Код"), new
DataColumn("Логин") });

    for (int i = 0; i < _dt.Rows.Count; i++)
        dt.Rows.Add(_dt.Rows[i][0], _dt.Rows[i][1]);
    return dt;
}
```

Данный метод с помощью операторов SELECT и WHERE ищет в базе данных пару логин/пароль. Если совпадение найдено, возвращаются данные о пользователе, после чего возможен вход в систему. На рисунке 2.8 изображен данный процесс в виде блок-схемы.

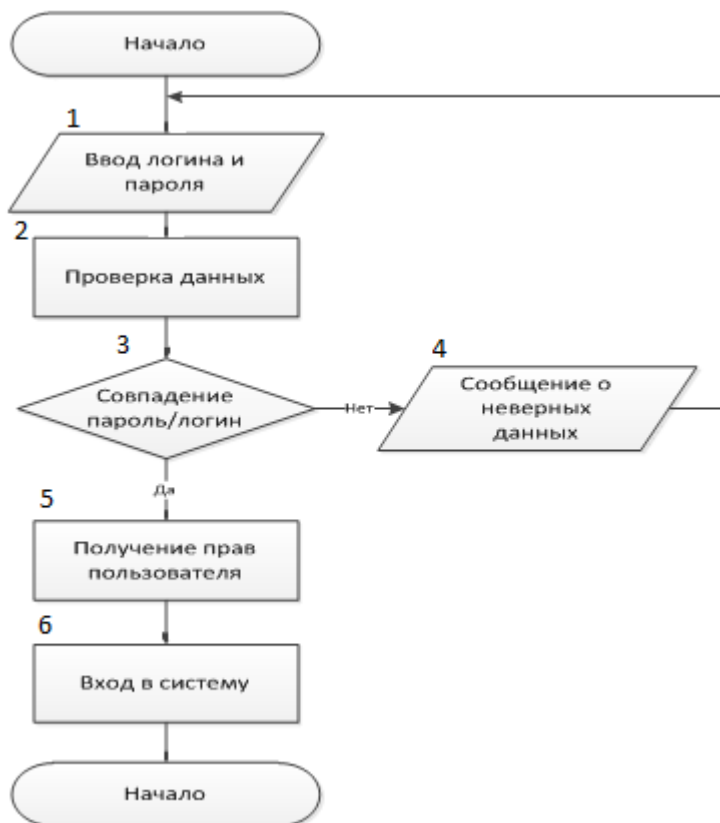


Рисунок 2.8 – Авторизация

Листинг 2.4 – Запрос выборки пользователей

```
"SELECT p.Код, p.Фамилия, p.Имя, p.Логин, p.eMail, g.Группа, b.Филиал FROM
  Пользователи p, Группы g, Филиалы b, p LEFT JOIN g ON p.Группа=g.Код, p LEFT JOIN b
ON p.Филиал=b.Код WHERE p.Статус=false
```

Представленный запрос осуществляет выборку данных из таблиц: Пользователи, Группы, Филиалы и с помощью оператора LEFT JOIN и включает их в один результирующий поток.

Листинг 2.5 – Метод запуска сервера

```
public void StartServer()
{
    var thread2 = new Thread(() =>
    {
        if (_bool == false)
        {
            serviceHostClient = new
                ServiceHost(typeof(Functional),
                new Uri("http://localhost:7792/Testing"));
            serviceHostClient.AddServiceEndpoint(typeof(IFunctional), new
                BasicHttpBinding(), "");

            serviceHostClient.Open();
            _bool = true;
        }
        else
        {
            serviceHostClient.Close();
            _bool = false;
        }
    });
    thread2.Start();
}
```

Данный метод с помощью служб WCF осуществляет запуск прослушивания соединения в поиске входящих запросов из клиентских приложений.

2.4 Характеристика интерфейса, эргономические решения, инструкция пользователя

Интерфейс разработанной системы реализовано стандартными методами, предоставляемых средой Microsoft Visual Studio на базе Windows Forms.

При разработке интерфейса применялись только стандартные компоненты, они описаны во многих источниках, в частности в литературе

[14,18,21,22]. Учитывая последнее, считаем детальное описание компонентов неуместным в работе.

Пример разработки одной из форм приведен на рисунке 2.9.

В общем - интерфейс поддерживает все современные правила формирования GUI на основе WinForms типа приложения.

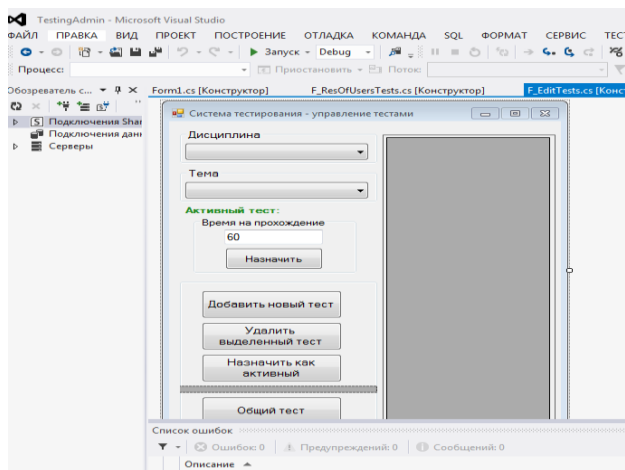


Рисунок 2.9 – Реализация интерфейса. Форма управления тестами

Инструкция пользователя и описание серверной части представлено ниже:

После запуска программы пользователю открывается доступ к его основной функциональной форме (Рисунок 2.10).

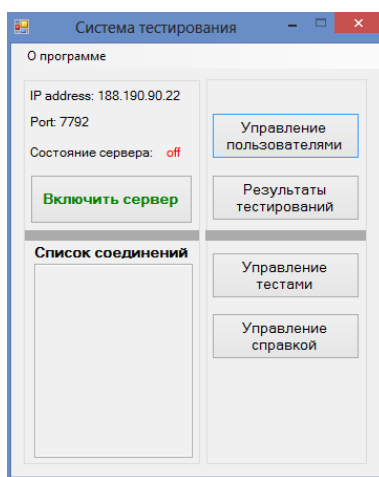


Рисунок 2.10 – Главная форма программы

Далее опишем основные функциональные части главной формы: «Включить сервер», «Управление пользователями», «Результаты тестирований», «Управление тестами», «Управление справкой».

При нажатии на пункт меню «Включить сервер», будут запущены службы устанавливающие возможность клиент-серверного соединения.

Пункт меню «Управление пользователями» осуществляет переход и открытие формы управления пользователями и администраторами системы (Рисунок 2.11).

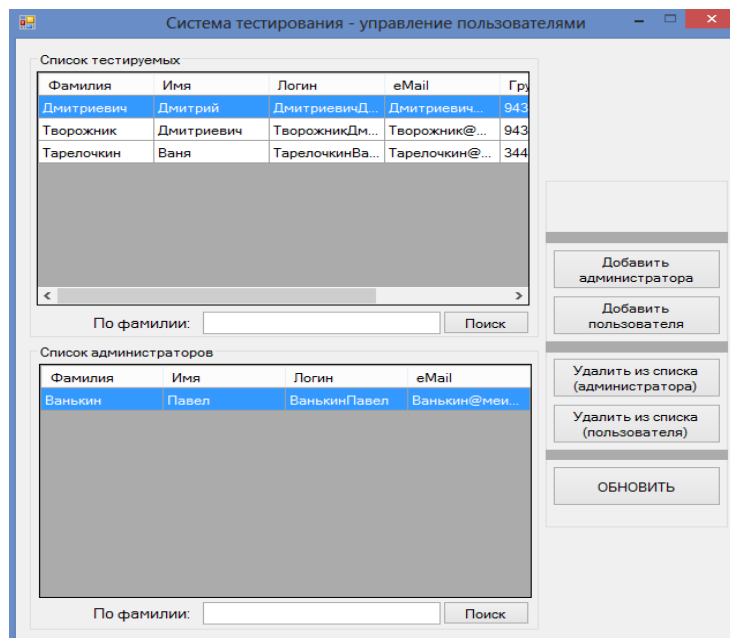


Рисунок 2.11 – Форма управления пользователями

При переходе по пункту меню «Результаты тестирования» откроется форма, представляющая собой список тестируемых и пройденных ими тестов. Так же в данной форме доступна опция фильтрации (сортировки) данных по различным категориям.

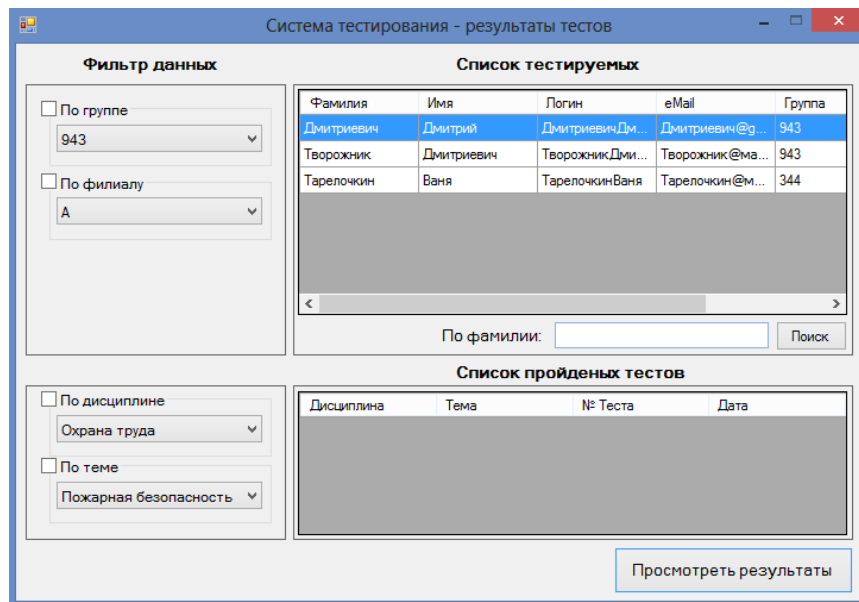


Рисунок 2.12 – Форма результатов тестирования

Пункт меню «Управление тестами» дает возможность пользователю управлять и вносить тесты в систему. Так же в данной форме можно устанавливать время прохождения тестов и назначать активный тест.

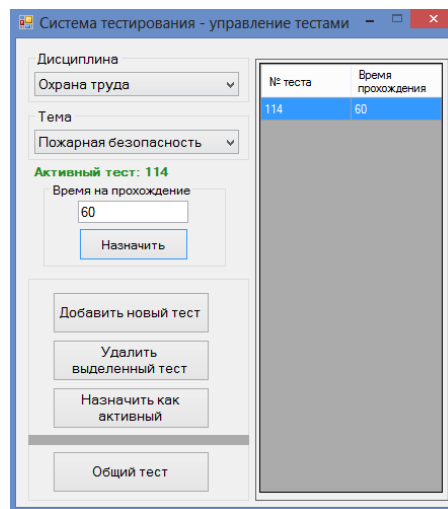


Рисунок 2.13 – Форма управления тестами

Из формы управления тестами можно выделить один самый важный объект функционирования – добавление нового теста (Рисунок 2.14).

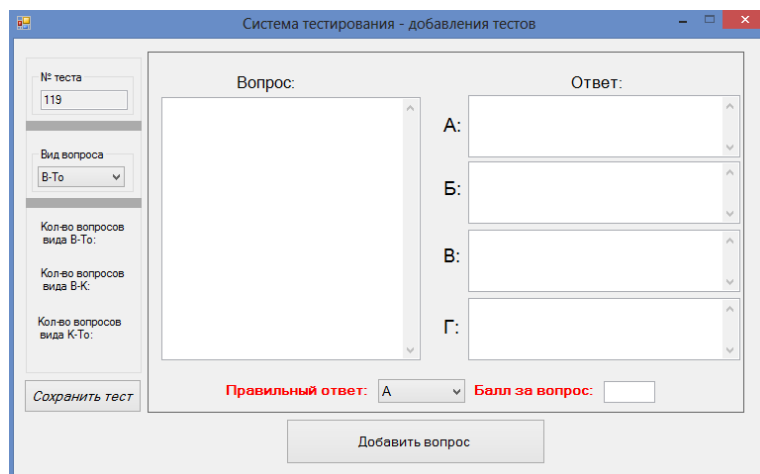


Рисунок 2.14 – Добавление нового теста

Как видно из рисунка, вопросы тестов могут состоять из трех категорий (видов):

- 1) В-То: Вопрос – Текстовый ответ
- 2) В-К: Вопрос – Картинка (Рисунок)
- 3) К-То: Картинка (Рисунок) – Текстовый ответ.

После добавления вопросов и верного ответа, необходимо так-же добавить оценочный балл за него, после чего он внесется в общий список вопросов для теста.

Последняя функциональная часть главной формы, это «Управление справкой». Данная часть позволяет вносить в систему различную справочную документацию, которую, в дальнейшем, можно будет просматривать в клиентских приложениях.

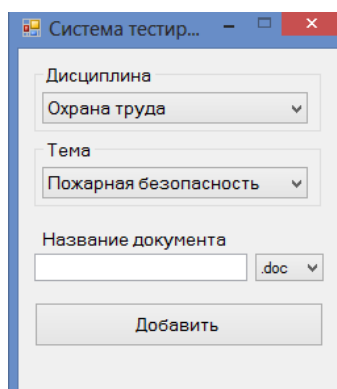


Рисунок 2.15 – Добавления справочной документации

После описания серверной части, можно перейти к описанию пользовательского интерфейса клиентской части разработанного программного обеспечения.

При открытии клиентского приложения появится форма авторизации (Рисунок 2.16) с возможностью регистрации нового аккаунта в системе (Рисунок 2.17).

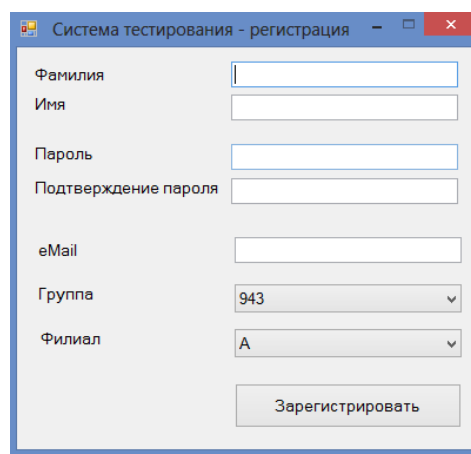
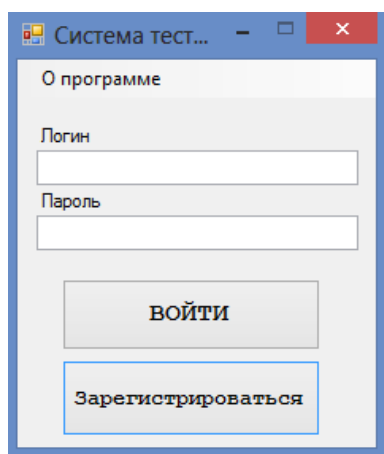


Рисунок 2.16 – Окно авторизации Рисунок 2.17 - Окно регистрации

После успешной авторизации или регистрации, пользователь сможет начать прохождения тестов. Логин пользователя складывается с его имени и фамилии.

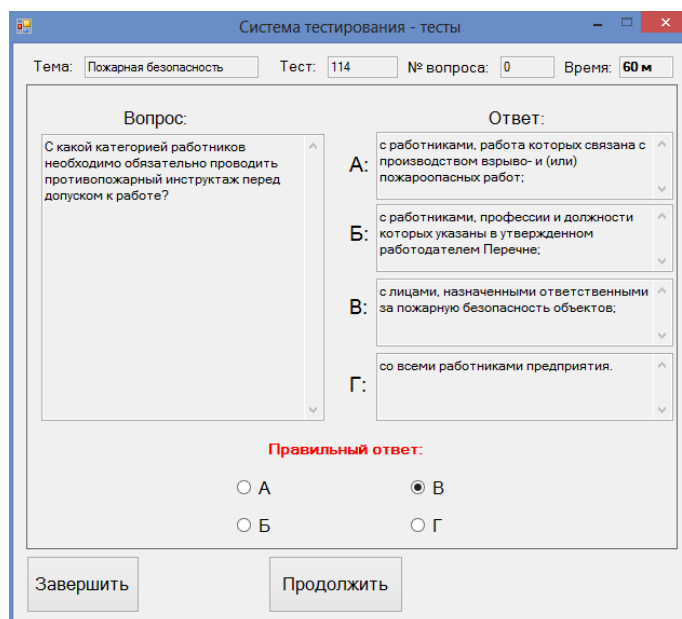


Рисунок 2.18 – Окно прохождения тестов

Выводы по разделу

В этой главе было проведено моделирование разрабатываемой программы, составлена схема базы данных. Используя WCF и язык C#, разработана межсетевая система, основной задачей которой является организация прохождения тестирования студентов. Программа разделена на две части – клиент и сервер. Взаимодействие с программой происходит через графический интерфейс, а именно через диалоговые окна Windows Form.

ВЫВОДЫ

Информационные технологии все больше проникают во все сферы человеческой деятельности, естественно и сектор образования требует внедрения современных технологий. При этом образование (особенно ВУЗы и профессиональное специализированное образование) требует первоочередного внимания. Логика достаточно проста – если специалисты в процессе обучения не ознакомлены с современными технологиями и не имеют навыков по работе с ИТ, то каким образом они могут интегрироваться в прикладные сферы, работа в которых, на текущий момент, в большей степени опирается на использование ИТ. Ускорение научно - технического прогресса, основанное на внедрении в производство гибких автоматизированных систем, микропроцессорных средств и устройств программного управления, роботов и обрабатывающих центров, поставило перед современной педагогической наукой важную задачу - воспитать и подготовить подрастающее поколение, способное активно включиться в качественно новый этап развития современного общества, связанный с информатизацией.

Отдельным вопросом является повышение качества образования за счет использования инновационных методик основанных на программных продуктах и информационных технологиях.

Большинство специальностей и направлений требует постоянного совершенствования, владение актуальными теоретическими знаниями и наличие навыков работы с использованием современных технологий в отрасли. Процесс проверки подобных знаний и компетенций частично может быть автоматизирован за счет использования информационной системы тестирования.

Эта работа исследует вопрос именно в направлении автоматизации контроля и оценки знаний за счет внедрения системы автоматизации

процесса тестирования, как составной части единого информационного пространства образовательного учреждения.

Практическая значимость работы заключается в возможности применения разработанной программы в реальных условиях для тестирования студентов или сотрудников (при необходимости). Практическая составляющая работы - программное приложение полностью функциональным и может быть адаптировано под различные нужды.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Закон РФ "Об образовании в Российской Федерации" // Статья 98. Информационные системы в системе образования.
2. Карпушина Т. Инновационные подходы в обучении// педагогический клуб «Открытый урок» / "Технология" 20.05.2017
3. Горбунова Л. И., Субботина Е. А. Использование информационных технологий в процессе обучения // Молодой ученый. — 2015. — №4. — С. 544-547.
4. Вигерс Карл, Разработка требований к программному обеспечению, Пер, с англ. - М.:Издательско-торговый дом "Русская Редакция", 2008. -576с.: ил
5. Вендров А.М., CASE-технологии. Современные методы и средства проектирования информационных систем - М.: Финансы и статистика, 2007 г, 456 стр.
6. Гвоздева Т. В., Б. А. Баллод, Проектирование информационных систем, М, Издательство: Феникс, 2009 г., 512 стр.
7. Голицына О. Л., И. И. Попов, Н. В. Максимов, Т. Л. Партыка, Информационные технологии, М, Издательство Инфра-М, 2009 г., 608 стр.
8. Котляров В. П., Т. В. Коликова, Основы тестирования программного обеспечения, Издательства: Интернет-университет информационных технологий, Бином. Лаборатория знаний, 2011 г., 288 стр.
9. Реляционные базы данных: практические приемы оптимальных решений. — СПб.: БХВ-Петербург, 2009 — 400с.:ил;
10. Емельянова Н. З., Партыка Т. Л., И. И. Попов, Проектирование информационных систем, М, Издательство: Форум, 2009 г., 432 стр.
11. Симионов Ю.Ф., Боромотов В.В. Информационный менеджмент. — Ростов н. Д: Феникс, 2012, 250с., ил.;
12. Илющечкин В. М. , Основы использования и проектирования баз данных, М, Издательство Юрайт, 2010 г., 224 стр.
13. Грабер, М. Введение в SQL: Самое простое введение в SQL с полезными примерами и пробными объяснениями и др. / Пер. с англ. В. Ястребов – Издательство «Лори», 2015. – 375 с. – ISBN – 5-85582-010-6.
14. Культин, Н. Б. Microsoft Visual C# в задачах и примерах./ Н.Б. Культин. – СПб.: БХВ-Петербург, 2009. – 320 с.: ил. – ISBN 978-5-9775-0410-2.
15. Михеева, Е.В. Информационные технологии в профессиональной деятельности: учеб. Пособие 2 издание. – М.: ТК Велби, Изд-во Проспект, 2012. – 448 с. –

ISBN 978-5-482-01569-8.

16. Нейгел, Кристиан, Ивѐн, Билл, Глинн, Джей, Скиннер, Морган, Уотсон, Карли. C# 2012 и платформа .NET 4.0 для профессионалов: Пер. с англ. / Кристиан Нейгел, Билл Ивѐн, Джей Глинн, Морган Скиннер, Карли Уотсон.– М.: Вильямс, 2013. – 1376 с. : ил. – ISBN 978-5-8459-1317-3 (рус.).

17. Балабанова Л.В., Сердак О.В. Отбор персонала через тестирование: учебное пособие: - М.: ИД «Профессионал», 2014. – 512 с.

18. Агуров Павел. C#. Разработка компонентов в MS Visual Studio 2005-2008. 2008, 467 стр..

19. Астелс, Дэвид; Миллер Гранвилл; Новак, Мирослав, Практическое руководство по экстремальному программированию, Пер. с англ. - М.: Издательский дом "Вильямс", 2008. - 320 с.: ил. - Парал. тит. Англ

20. ГОСТ ЕСПД 19.102-77 “Стадии разработки ”

21. Дейтел Х. C# в подлиннике. Наиболее полное руководство. 2006, 1057 стр..

22. ИНТУИТ - Разработка Windows-приложений на основе Visual C# 2012.

23. Черненко О.Н. Информационные технологии в учебном процессе: нормативное обеспечение, рекомендации из опыта работы. – Волгоград: Учитель, 2014 г.

24. Д.А. Дмитриев ИСПОЛЬЗОВАНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ. Вектор науки ТГУ. 4(7). 2015 89

25. Siddeeq, Y. AES Cryptosystem Development Using Neural Networks / Y. Siddeeq, A.H. Mahdi // International Journal of Computer and Electrical Engineering. – 2011. – v. 3. – № 2. – p. 315- 318

26. Gaviano, M. Algorithm 829: Software for generation of classes of test functions with known local and global minima for global optimization / M.Gaviano, D.E. Kvasov, D Lera, Y. D. Sergeyev // ACM Trans. Math. Software.— 2003. – Vol. 29, № 4. – p. 469-480

27. Gergel, V.P. A software system for multiextremal optimization / V.P. Gergel //11 European J. Oper. Res. – 1993. – Vol. 65, № 3. – p. 305-313.

28. Marshalko, G.B. On the security of a neural network-based biometric authentication scheme / G.B. Marshalko // Матем. вопр. криптогр. – 2014. – с. 87–98.

29. Latha, P. Face Recognition using Neural Networks / P. Latha, L. Ganesan, S. Annadurai // Signal Processing: An International Journal (SPIJ). – v.3. – №5. – p.153- 160.

30. Trevathan, J. Neural network-based handwritten signature verification / J. Trevathan, W. Read, A. McCabe // J Comput. – v. 3. – №8. – 2008. – p. 9–22.

ПРИЛОЖЕНИЕ 1. КОД ПРОГРАММЫ

```
namespace Testing
{
    partial class F_Registration
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.cmb_Groups = new System.Windows.Forms.ComboBox();
            this.btn_Reg = new System.Windows.Forms.Button();
            this.tb_Name = new System.Windows.Forms.TextBox();
            this.tb_Surname = new System.Windows.Forms.TextBox();
            this.label4 = new System.Windows.Forms.Label();
            this.label3 = new System.Windows.Forms.Label();
            this.tb_Pass = new System.Windows.Forms.TextBox();
            this.label6 = new System.Windows.Forms.Label();
            this.label7 = new System.Windows.Forms.Label();
            this.tb_ConPass = new System.Windows.Forms.TextBox();
            this.label2 = new System.Windows.Forms.Label();
            this.SuspendLayout();
            //
            // cmb_Groups
            //
            this.cmb_Groups.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList;
```



```

        this.cmb_Groups.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.cmb_Groups.FormattingEnabled = true;
        this.cmb_Groups.Location = new System.Drawing.Point(182, 153);
        this.cmb_Groups.Name = "cmb_Groups";
        this.cmb_Groups.Size = new System.Drawing.Size(193, 24);
        this.cmb_Groups.TabIndex = 20;
        //
        // btn_Reg
        //
        this.btn_Reg.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.btn_Reg.Location = new System.Drawing.Point(182, 195);
        this.btn_Reg.Name = "btn_Reg";
        this.btn_Reg.RightToLeft = System.Windows.Forms.RightToLeft.Yes;
        this.btn_Reg.Size = new System.Drawing.Size(193, 53);
        this.btn_Reg.TabIndex = 18;
        this.btn_Reg.Text = "Регистрация";
        this.btn_Reg.UseVisualStyleBackColor = true;
        this.btn_Reg.Click += new System.EventHandler(this.btn_Reg_Click);
        //
        // tb_Name
        //
        this.tb_Name.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.tb_Name.Location = new System.Drawing.Point(182, 40);
        this.tb_Name.Name = "tb_Name";
        this.tb_Name.Size = new System.Drawing.Size(193, 22);
        this.tb_Name.TabIndex = 17;
        //
        // tb_Surname
        //
        this.tb_Surname.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.tb_Surname.Location = new System.Drawing.Point(182, 12);
        this.tb_Surname.Name = "tb_Surname";
        this.tb_Surname.Size = new System.Drawing.Size(193, 22);
        this.tb_Surname.TabIndex = 16;
        //
        // label4
        //
        this.label4.AutoSize = true;
        this.label4.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.label4.Location = new System.Drawing.Point(12, 15);
        this.label4.Name = "label4";
        this.label4.Size = new System.Drawing.Size(67, 16);
        this.label4.TabIndex = 15;
        this.label4.Text = "Фамилия";
        //
        // label3
        //

```

```

        this.label3.AutoSize = true;
        this.label3.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.label3.Location = new System.Drawing.Point(12, 40);
        this.label3.Name = "label3";
        this.label3.Size = new System.Drawing.Size(34, 16);
        this.label3.TabIndex = 14;
        this.label3.Text = "Имя";
        //
        // tb_Pass
        //
        this.tb_Pass.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.tb_Pass.Location = new System.Drawing.Point(182, 82);
        this.tb_Pass.Name = "tb_Pass";
        this.tb_Pass.PasswordChar = '*';
        this.tb_Pass.Size = new System.Drawing.Size(193, 22);
        this.tb_Pass.TabIndex = 17;
        //
        // label6
        //
        this.label6.AutoSize = true;
        this.label6.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.label6.Location = new System.Drawing.Point(12, 82);
        this.label6.Name = "label6";
        this.label6.Size = new System.Drawing.Size(57, 16);
        this.label6.TabIndex = 14;
        this.label6.Text = "Пароль";
        //
        // label7
        //
        this.label7.AutoSize = true;
        this.label7.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.label7.Location = new System.Drawing.Point(12, 111);
        this.label7.Name = "label7";
        this.label7.Size = new System.Drawing.Size(164, 16);
        this.label7.TabIndex = 14;
        this.label7.Text = "Подтверждение пароля";
        //
        // tb_ConPass
        //
        this.tb_ConPass.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.tb_ConPass.Location = new System.Drawing.Point(182, 111);
        this.tb_ConPass.Name = "tb_ConPass";
        this.tb_ConPass.PasswordChar = '*';
        this.tb_ConPass.Size = new System.Drawing.Size(193, 22);
        this.tb_ConPass.TabIndex = 17;
        //
        // label2

```

```

//
this.label2.AutoSize = true;
this.label2.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.label2.Location = new System.Drawing.Point(12, 153);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(57, 16);
this.label2.TabIndex = 23;
this.label2.Text = "Должность";
//
// F_Registration
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.AutoSizeMode = System.Windows.Forms.AutoSizeMode.GrowAndShrink;
this.ClientSize = new System.Drawing.Size(390, 260);
this.Controls.Add(this.label2);
this.Controls.Add(this.cmb_Groups);
this.Controls.Add(this.btn_Reg);
this.Controls.Add(this.tb_ConPass);
this.Controls.Add(this.tb_Pass);
this.Controls.Add(this.tb_Name);
this.Controls.Add(this.tb_Surname);
this.Controls.Add(this.label4);
this.Controls.Add(this.label7);
this.Controls.Add(this.label6);
this.Controls.Add(this.label3);
this.MaximizeBox = false;
this.Name = "F_Registration";
this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
this.Text = "Система тестирования - регистрация";
this.FormClosed += new
System.Windows.Forms.FormClosedEventHandler(this.F_Registration_FormClosed);
this.Load += new System.EventHandler(this.Registration_Load);
this.ResumeLayout(false);
this.PerformLayout();

```

```

}

```

```

#endregion

```

```

private System.Windows.Forms.ComboBox cmb_Groups;
private System.Windows.Forms.Button btn_Reg;
private System.Windows.Forms.TextBox tb_Name;
private System.Windows.Forms.TextBox tb_Surname;
private System.Windows.Forms.Label label4;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.TextBox tb_Pass;
private System.Windows.Forms.Label label6;
private System.Windows.Forms.Label label7;
private System.Windows.Forms.TextBox tb_ConPass;
private System.Windows.Forms.Label label2;

```

```

    }
}
private void F_Main_FormClosed(object sender, FormClosedEventArgs e)
{
    FormDialog.F_Logo.Close();
} }

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

```

```

namespace Testing
{
    public partial class F_Registration : Form
    {
        DataTable dtBrunch = new DataTable();
        DataTable dtGroup = new DataTable();
        public F_Registration()
        {
            InitializeComponent();
            ButtonsImages(btn_Reg, "Reg.png");
        }

        private void Registration_Load(object sender, EventArgs e)
        {
            dtGroup = ListenServer.wcfClient.GetGroup();

            cmb_Groups.DataSource = dtGroup;
            cmb_Groups.DisplayMember = "Должность";
        }

        void ButtonsImages(Button button1, String image)
        {
            button1.Image = Image.FromFile(@"images\" + image + "", false);
            //button1.BackgroundImageLayout = ImageLayout.Zoom;
            // Align the image and text on the button.
            button1.ImageAlign = ContentAlignment.MiddleLeft;
            button1.TextAlign = ContentAlignment.MiddleRight;
            // Give the button a flat appearance.
            button1.FlatStyle = FlatStyle.Flat;
        }

        private void btn_Reg_Click(object sender, EventArgs e)
        {
            if ( cmb_Groups.SelectedIndex >= 0 && tb_Surname.Text.Length > 0
                && tb_Name.Text.Length > 0)

```

```

    {
        Boolean isReg = Reg();
        if (isReg == false)
        {
            DialogResult result = MessageBox.Show("Ошибка регистрации", "Ошибка",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        if (isReg == true)
        {
            DialogResult result = MessageBox.Show("Регистрация прошла успешно.
            Продолжить работу?", "Сообщение", MessageBoxButtons.YesNo,
            MessageBoxIcon.Information);
            if (result == DialogResult.Yes)
            {
                //Если регистрация успешно
                FormDialog.F_Main.Visible = false;
                Close();
            }
            if (result == DialogResult.No)
            {
                Close();
            }
        }
        } else MessageBox.Show("Не все поля заполнены", "Ошибка",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    Boolean Reg()
    {
        Boolean isReg;
        if (tb_Pass.Text == tb_ConPass.Text)
        {
            isReg = ListenServer.wcfClient.Registration(tb_Pass.Text, tb_Name.Text,
            tb_Surname.Text,
            Convert.ToInt32(dtGroup.Rows[cmb_Groups.SelectedIndex][0]));
            return isReg;
        } else
        {
            MessageBox.Show("Пароли не совпадают", "Ошибка", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
            return false;
        }
    }
}

private void F_Registration_FormClosed(object sender, FormClosedEventArgs e)
{
    FormDialog.F_Main.Visible = true;
}

```

```

    }
}
namespace Testing
{
    partial class F_Menu
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.btn_NewTest = new System.Windows.Forms.Button();
            this.btn_Close = new System.Windows.Forms.Button();
            this.btn_Help = new System.Windows.Forms.Button();
            this.SuspendLayout();
            //
            // btn_NewTest
            //
            this.btn_NewTest.Font = new System.Drawing.Font("Courier New", 12F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
            this.btn_NewTest.Location = new System.Drawing.Point(25, 51);
            this.btn_NewTest.Name = "btn_NewTest";
            this.btn_NewTest.RightToLeft = System.Windows.Forms.RightToLeft.Yes;
            this.btn_NewTest.Size = new System.Drawing.Size(177, 60);
            this.btn_NewTest.TabIndex = 0;
            this.btn_NewTest.Text = "Новый тест";
            this.btn_NewTest.UseVisualStyleBackColor = true;
            this.btn_NewTest.Click += new System.EventHandler(this.btn_NewTest_Click);
            //
            // btn_Close

```

```

//
this.btn_Close.Font = new System.Drawing.Font("Courier New", 12F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.btn_Close.Location = new System.Drawing.Point(25, 183);
this.btn_Close.Name = "btn_Close";
this.btn_Close.RightToLeft = System.Windows.Forms.RightToLeft.Yes;
this.btn_Close.Size = new System.Drawing.Size(177, 60);
this.btn_Close.TabIndex = 0;
this.btn_Close.Text = "ЗАКРЫТЬ";
this.btn_Close.UseVisualStyleBackColor = true;
this.btn_Close.Click += new System.EventHandler(this.btn_Close_Click);
//
// btn_Help
//
this.btn_Help.Font = new System.Drawing.Font("Courier New", 11.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.btn_Help.Location = new System.Drawing.Point(25, 117);
this.btn_Help.Name = "btn_Help";
this.btn_Help.RightToLeft = System.Windows.Forms.RightToLeft.Yes;
this.btn_Help.Size = new System.Drawing.Size(177, 60);
this.btn_Help.TabIndex = 0;
this.btn_Help.Text = "Учебные\г\ндокументы";
this.btn_Help.UseVisualStyleBackColor = true;
this.btn_Help.Click += new System.EventHandler(this.btn_Help_Click);
//
// F_Menu
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.AutoSizeMode = System.Windows.Forms.AutoSizeMode.GrowAndShrink;
this.ClientSize = new System.Drawing.Size(226, 311);
this.Controls.Add(this.btn_Help);
this.Controls.Add(this.btn_Close);
this.Controls.Add(this.btn_NewTest);
this.MaximizeBox = false;
this.Name = "F_Menu";
this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
this.FormClosed += new
System.Windows.Forms.FormClosedEventHandler(this.F_Menu_FormClosed);
this.Load += new System.EventHandler(this.F_Menu_Load);
this.ResumeLayout(false);

}

#endregion

private System.Windows.Forms.Button btn_NewTest;
private System.Windows.Forms.Button btn_Close;
private System.Windows.Forms.Button btn_Help;
}
}
using System;

```

```

using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Testing
{
    public partial class F_Menu : Form
    {
        F_AddTest F_AddTest = new F_AddTest();
        F_Help F_Help = new F_Help();
        public F_Menu()
        {
            InitializeComponent();

            ButtonsImages(btn_NewTest, "NewTest.png");
            ButtonsImages(btn_Help, "Help.png");
            ButtonsImages(btn_Close, "Close.png");
        }

        private void btn_NewTest_Click(object sender, EventArgs e)
        {
            FormDialog.dtTests = ListenServer.wcfClient.GetTests();
            FormDialog.Topic =
ListenServer.wcfClient.GetTopic(Convert.ToInt32(FormDialog.dtTests.Rows[0][2]));
            FormDialog.NameOfTest =
ListenServer.wcfClient.GetNameOfTest(Convert.ToInt32(FormDialog.dtTests.Rows[0][2]));

            if (F_AddTest.Created == true) F_AddTest.Close();
            F_AddTest = new F_AddTest();
            F_AddTest.Show();

            FormDialog.F_Menu.Visible=false;

        }

        private void F_Menu_FormClosed(object sender, FormClosedEventArgs e)
        {
            ListenServer.wcfClient.DeleteConnectUsers(FormDialog.Login);
            FormDialog.F_Main.Show();
        }

        private void F_Menu_Load(object sender, EventArgs e)

```



```

    {
        ListenServer.wcfClient.getConnectUsers(FormDialog.Login);
        FormDialog.Time = ListenServer.wcfClient.GetTime();
    }
void ButtonsImages(Button button1, String image)
{
    button1.Image = Image.FromFile(@"images\" + image + "", false);
    //button1.BackgroundImageLayout = ImageLayout.Zoom;
    // Align the image and text on the button.
    button1.ImageAlign = ContentAlignment.MiddleLeft;
    button1.TextAlign = ContentAlignment.MiddleRight;
    // Give the button a flat appearance.
    button1.FlatStyle = FlatStyle.Flat;
}
private void btn_Help_Click(object sender, EventArgs e)
{
    if (F_Help.Created == true) F_Help.Close();
    F_Help = new F_Help();
    F_Help.Show();
}

private void btn_Close_Click(object sender, EventArgs e)
{
    Close();
}
}
}

namespace Testing
{
    partial class F_Main
    {
        /// <summary>
        /// Требуется переменная конструктора.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Освободить все используемые ресурсы.
        /// </summary>
        /// <param name="disposing">истинно, если управляемый ресурс должен быть удален;
        иначе ложно.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Код, автоматически созданный конструктором форм Windows

```

```

/// <summary>
/// Обязательный метод для поддержки конструктора - не изменяйте
/// содержимое данного метода при помощи редактора кода.
/// </summary>
private void InitializeComponent()
{
    System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(F_Main));
    this.btn_Reg = new System.Windows.Forms.Button();
    this.btn_Login = new System.Windows.Forms.Button();
    this.tb_Login = new System.Windows.Forms.TextBox();
    this.tb_Pass = new System.Windows.Forms.TextBox();
    this.label1 = new System.Windows.Forms.Label();
    this.label2 = new System.Windows.Forms.Label();
    this.menuStrip1 = new System.Windows.Forms.MenuStrip();
    this.обновитьПодключениеToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
    this.тестToolStripMenuItem = new System.Windows.Forms.ToolStripItem();
    this.button1 = new System.Windows.Forms.Button();
    this.menuStrip1.SuspendLayout();
    this.SuspendLayout();
    //
    // btn_Reg
    //
    this.btn_Reg.BackColor = System.Drawing.Color.Transparent;
    this.btn_Reg.Font = new System.Drawing.Font("Courier New", 9.75F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
    this.btn_Reg.ForeColor = System.Drawing.SystemColors.ControlLightLight;
    this.btn_Reg.Location = new System.Drawing.Point(28, 189);
    this.btn_Reg.Name = "btn_Reg";
    this.btn_Reg.RightToLeft = System.Windows.Forms.RightToLeft.Yes;
    this.btn_Reg.Size = new System.Drawing.Size(163, 48);
    this.btn_Reg.TabIndex = 0;
    this.btn_Reg.Text = "Регистрация";
    this.btn_Reg.UseVisualStyleBackColor = false;
    this.btn_Reg.Click += new System.EventHandler(this.btn_Reg_Click);
    this.btn_Reg.MouseEnter += new System.EventHandler(this.btn_Reg_MouseEnter);
    this.btn_Reg.MouseLeave += new System.EventHandler(this.btn_Reg_MouseLeave);
    //
    // btn_Login
    //
    this.btn_Login.BackColor = System.Drawing.Color.Transparent;
    this.btn_Login.Font = new System.Drawing.Font("Courier New", 12F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
    this.btn_Login.ForeColor = System.Drawing.SystemColors.ControlLightLight;
    this.btn_Login.Location = new System.Drawing.Point(28, 138);
    this.btn_Login.Name = "btn_Login";
    this.btn_Login.RightToLeft = System.Windows.Forms.RightToLeft.Yes;
    this.btn_Login.Size = new System.Drawing.Size(163, 45);
    this.btn_Login.TabIndex = 1;
    this.btn_Login.Text = "ВОЙТИ";
}

```

```

this.btn_Login.UseVisualStyleBackColor = false;
this.btn_Login.Click += new System.EventHandler(this.btn_Login_Click);
this.btn_Login.MouseEnter += new System.EventHandler(this.btn_Login_MouseEnter);
this.btn_Login.MouseLeave += new
System.EventHandler(this.btn_Login_MouseLeave);
//
// tb_Login
//
this.tb_Login.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.tb_Login.Location = new System.Drawing.Point(12, 57);
this.tb_Login.Name = "tb_Login";
this.tb_Login.Size = new System.Drawing.Size(203, 22);
this.tb_Login.TabIndex = 2;
//
// tb_Pass
//
this.tb_Pass.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.tb_Pass.Location = new System.Drawing.Point(12, 98);
this.tb_Pass.Name = "tb_Pass";
this.tb_Pass.PasswordChar = '*';
this.tb_Pass.Size = new System.Drawing.Size(203, 22);
this.tb_Pass.TabIndex = 3;
//
// label1
//
this.label1.AutoSize = true;
this.label1.BackColor = System.Drawing.Color.Transparent;
this.label1.ForeColor = System.Drawing.SystemColors.ControlLightLight;
this.label1.Location = new System.Drawing.Point(12, 41);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(124, 13);
this.label1.TabIndex = 4;
this.label1.Text = "Логин (Фамилия+Имя)";
//
// label2
//
this.label2.AutoSize = true;
this.label2.BackColor = System.Drawing.Color.Transparent;
this.label2.ForeColor = System.Drawing.SystemColors.ControlLightLight;
this.label2.Location = new System.Drawing.Point(12, 82);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(45, 13);
this.label2.TabIndex = 5;
this.label2.Text = "Пароль";
//
// menuStrip1
//
this.menuStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
this.обновитьПодключениеToolStripMenuItem,
this.тестToolStripMenuItem});

```

```

this.menuStrip1.Location = new System.Drawing.Point(0, 0);
this.menuStrip1.Name = "menuStrip1";
this.menuStrip1.Size = new System.Drawing.Size(237, 24);
this.menuStrip1.TabIndex = 6;
this.menuStrip1.Text = "menuStrip1";
//
// обновитьПодключениеToolStripMenuItem
//
this.обновитьПодключениеToolStripMenuItem.Name =
"обновитьПодключениеToolStripMenuItem";
this.обновитьПодключениеToolStripMenuItem.Size = new System.Drawing.Size(94,
20);
this.обновитьПодключениеToolStripMenuItem.Text = "О программе";
this.обновитьПодключениеToolStripMenuItem.Click += new
System.EventHandler(this.обновитьПодключениеToolStripMenuItem_Click);
//
// тестToolStripMenuItem
//
this.тестToolStripMenuItem.Name = "тестToolStripMenuItem";
this.тестToolStripMenuItem.Size = new System.Drawing.Size(94, 20);
this.тестToolStripMenuItem.Text = "Справочники";
this.тестToolStripMenuItem.Visible = false;
this.тестToolStripMenuItem.Click += new
System.EventHandler(this.тестToolStripMenuItem_Click);
//
// button1
//
this.button1.BackColor = System.Drawing.Color.Transparent;
this.button1.Font = new System.Drawing.Font("Courier New", 9.75F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.button1.ForeColor = System.Drawing.SystemColors.ControlLightLight;
this.button1.Location = new System.Drawing.Point(28, 243);
this.button1.Name = "button1";
this.button1.RightToLeft = System.Windows.Forms.RightToLeft.Yes;
this.button1.Size = new System.Drawing.Size(163, 48);
this.button1.TabIndex = 0;
this.button1.Text = "Справочники";
this.button1.UseVisualStyleBackColor = false;
this.button1.Click += new System.EventHandler(this.button1_Click);
this.button1.MouseEnter += new System.EventHandler(this.button1_MouseEnter);
this.button1.MouseLeave += new System.EventHandler(this.button1_MouseLeave);
//
// F_Main
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.AutoSizeMode = System.Windows.Forms.AutoSizeMode.GrowAndShrink;
this.BackColor = System.Drawing.SystemColors.Control;
this.BackgroundImage =
((System.Drawing.Image)resources.GetObject("$this.BackgroundImage"));
this.ClientSize = new System.Drawing.Size(237, 311);
this.Controls.Add(this.label2);

```

```

        this.Controls.Add(this.label1);
        this.Controls.Add(this.tb_Pass);
        this.Controls.Add(this.tb_Login);
        this.Controls.Add(this.btn_Login);
        this.Controls.Add(this.button1);
        this.Controls.Add(this.btn_Reg);
        this.Controls.Add(this.menuStrip1);
        this.MainMenuStrip = this.menuStrip1;
        this.MaximizeBox = false;
        this.Name = "F_Main";
        this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
        this.Text = "ИС ТЕСТ";
        this.FormClosed += new
System.Windows.Forms.FormClosedEventHandler(this.F_Main_FormClosed);
        this.Load += new System.EventHandler(this.F_Main_Load);
        this.menuStrip1.ResumeLayout(false);
        this.menuStrip1.PerformLayout();
        this.ResumeLayout(false);
        this.PerformLayout();

    }

#endregion

private System.Windows.Forms.Button btn_Reg;
private System.Windows.Forms.Button btn_Login;
private System.Windows.Forms.TextBox tb_Login;
private System.Windows.Forms.TextBox tb_Pass;
private System.Windows.Forms.Label label1;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.MenuStrip menuStrip1;
private System.Windows.Forms.ToolStripMenuItem
обновитьПодключениеToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem тестToolStripMenuItem;
private System.Windows.Forms.Button button1;
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Testing
{
    public partial class F_Main : Form
    {

```

```

F_Registration F_Registration = new F_Registration();
F_Tree F_Tree = new F_Tree();
public F_Main()
{
    InitializeComponent();
    ButtonsImages(button1, "Help.png");
    ButtonsImages(btn_Login, "Login.png");
    ButtonsImages(btn_Reg, "Reg.png");
}

private void F_Main_Load(object sender, EventArgs e)
{
    ListenServer.ConnctctToServer();
    FormDialog.F_Main = this;
    FormDialog.F_Menu = new F_Menu();
}

void ButtonsImages(Button button1,String image)
{
    button1.Image = Image.FromFile(@"images\" +image +"" , false);
    //button1.BackgroundImageLayout = ImageLayout.Zoom;
    // Align the image and text on the button.
    button1.ImageAlign = ContentAlignment.MiddleLeft;
    button1.TextAlign = ContentAlignment.MiddleRight;
    // Give the button a flat appearance.
    button1.FlatStyle = FlatStyle.Flat;
}

private void btn_Reg_Click(object sender, EventArgs e)
{
    try
    {
        if (F_Registration.Created == true) F_Registration.Close();
        F_Registration = new F_Registration();
        F_Registration.Show();
        FormDialog.F_Main.Visible = false;
    }
    catch { MessageBox.Show("Ошибка подключения к серверу. Связь не устеновлена",
"Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error); }
}

private void btn_Login_Click(object sender, EventArgs e)
{
    try
    {
        FormDialog.dtUser = ListenServer.wcfClient.GetUser(tb_Login.Text, tb_Pass.Text);
        //Если успешно
        if (FormDialog.dtUser.Rows.Count > 0)
        {
            FormDialog.Login = tb_Login.Text;

```

```

        if (FormDialog.F_Menu.Created == true) FormDialog.F_Menu.Close();
        FormDialog.F_Menu = new F_Menu();
        FormDialog.F_Menu.Show();
        FormDialog.F_Main.Visible = false;

        tb_Login.Clear();
        tb_Pass.Clear();
    }
    else MessageBox.Show("Ошибка авторизации. Неверный 'Логин' или 'Пароль'.",
"Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    catch { MessageBox.Show("Ошибка подключения к серверу. Связь не установлена",
"Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error); }
    }

private void обновитьПодключениеToolStripMenuItem_Click(object sender, EventArgs
e)
{
    try
    {
        System.Windows.Forms.Help.ShowHelp(this, @"help.chm");
    }
    catch { MessageBox.Show("Файл справки не найден", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Error); }
}

private void F_Main_FormClosed(object sender, FormClosedEventArgs e)
{
    try
    {
        ListenServer.wcfClient.DeleteConnectUsers(FormDialog.Login);
    }
    catch { }
    FormDialog.F_Logo.Close();
}

private void тестToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (F_Tree.Created == true) F_Tree.Close();
    F_Tree = new F_Tree();
    F_Tree.Show();
    //FormDialog.F_Main.Visible = false;
}

private void button1_Click(object sender, EventArgs e)
{
    if (F_Tree.Created == true) F_Tree.Close();
    F_Tree = new F_Tree();
    F_Tree.Show();
}

private void btn_Login_MouseEnter(object sender, EventArgs e)

```

```

    {
        btn_Login.ForeColor = Color.Black;
    }

private void btn_Login_MouseLeave(object sender, EventArgs e)
{
    btn_Login.ForeColor = Color.White;
}

private void btn_Reg_MouseEnter(object sender, EventArgs e)
{
    btn_Reg.ForeColor = Color.Black;
}

private void btn_Reg_MouseLeave(object sender, EventArgs e)
{
    btn_Reg.ForeColor = Color.White;
}

private void button1_MouseEnter(object sender, EventArgs e)
{
    button1.ForeColor = Color.Black;
}

private void button1_MouseLeave(object sender, EventArgs e)
{
    button1.ForeColor = Color.White;
}
}
}

namespace Testing
{
    partial class F_Logo
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);

```



```

}

#region Windows Form Designer generated code

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    this.components = new System.ComponentModel.Container();
    this.pictureBox1 = new System.Windows.Forms.PictureBox();
    this.timer1 = new System.Windows.Forms.Timer(this.components);
    ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();
    this.SuspendLayout();
    //
    // pictureBox1
    //
    this.pictureBox1.Location = new System.Drawing.Point(-1, -1);
    this.pictureBox1.Name = "pictureBox1";
    this.pictureBox1.Size = new System.Drawing.Size(567, 408);
    this.pictureBox1.TabIndex = 1;
    this.pictureBox1.TabStop = false;
    this.pictureBox1.Click += new System.EventHandler(this.pictureBox1_Click);
    //
    // timer1
    //
    this.timer1.Tick += new System.EventHandler(this.timer1_Tick);
    //
    // F_Logo
    //
    this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
    this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
    this.AutoSizeMode = System.Windows.Forms.AutoSizeMode.GrowAndShrink;
    this.ClientSize = new System.Drawing.Size(558, 402);
    this.Controls.Add(this.pictureBox1);
    this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None;
    this.MaximizeBox = false;
    this.MinimizeBox = false;
    this.Name = "F_Logo";
    this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
    this.Text = "Система тестирования";
    this.Load += new System.EventHandler(this.F_Logo_Load);
    ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();
    this.ResumeLayout(false);

}

#endregion

private System.Windows.Forms.PictureBox pictureBox1;
private System.Windows.Forms.Timer timer1;

```

```

    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Testing
{
    public partial class F_Logo : Form
    {
        public F_Logo()
        {
            InitializeComponent();
            FormDialog.F_Logo = this;
        }

        private void F_Logo_Load(object sender, EventArgs e)
        {
            pictureBox1.ImageLocation = (@"заставка.jpg");
            pictureBox1.BackgroundImageLayout = ImageLayout.Stretch;
            pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;
            // F_Main f_main = new F_Main();
            //f_main.Show();
            // new System.Threading.Timer(state => { f_main.Show(); Close(); }, null, 1000, 0);
            timer1.Interval = 2000;
            timer1.Start();
        }

        private void pictureBox1_Click(object sender, EventArgs e)
        {
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            F_Main f_main = new F_Main();
            f_main.Show();
            this.Visible = false;
            timer1.Stop();
        }
    }
}

namespace Testing
{

```

```

partial class F_Help
{
    /// <summary>
    /// Required designer variable.
    /// </summary>
    private System.ComponentModel.IContainer components = null;

    /// <summary>
    /// Clean up any resources being used.
    /// </summary>
    /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
    protected override void Dispose(bool disposing)
    {
        if (disposing && (components != null))
        {
            components.Dispose();
        }
        base.Dispose(disposing);
    }

    #region Windows Form Designer generated code

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
        this.groupBox2 = new System.Windows.Forms.GroupBox();
        this.cmb_Topic = new System.Windows.Forms.ComboBox();
        this.groupBox1 = new System.Windows.Forms.GroupBox();
        this.cmb_Subject = new System.Windows.Forms.ComboBox();
        this.groupBox3 = new System.Windows.Forms.GroupBox();
        this.cmb_Doc = new System.Windows.Forms.ComboBox();
        this.rtb_Main = new System.Windows.Forms.RichTextBox();
        this.button1 = new System.Windows.Forms.Button();
        this.groupBox2.SuspendLayout();
        this.groupBox1.SuspendLayout();
        this.groupBox3.SuspendLayout();
        this.SuspendLayout();
        //
        // groupBox2
        //
        this.groupBox2.Controls.Add(this.cmb_Topic);
        this.groupBox2.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.groupBox2.Location = new System.Drawing.Point(236, 12);
        this.groupBox2.Name = "groupBox2";
        this.groupBox2.Size = new System.Drawing.Size(218, 51);
        this.groupBox2.TabIndex = 3;
        this.groupBox2.TabStop = false;
    }
}

```

```

this.groupBox2.Text = "Тема";
//
// cmb_Topic
//
this.cmb_Topic.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList;
this.cmb_Topic.FormattingEnabled = true;
this.cmb_Topic.Location = new System.Drawing.Point(6, 21);
this.cmb_Topic.Name = "cmb_Topic";
this.cmb_Topic.Size = new System.Drawing.Size(202, 24);
this.cmb_Topic.TabIndex = 1;
this.cmb_Topic.SelectedIndexChanged += new
System.EventHandler(this.cmb_Topic_SelectedIndexChanged);
//
// groupBox1
//
this.groupBox1.Controls.Add(this.cmb_Subject);
this.groupBox1.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.groupBox1.Location = new System.Drawing.Point(12, 12);
this.groupBox1.Name = "groupBox1";
this.groupBox1.Size = new System.Drawing.Size(218, 51);
this.groupBox1.TabIndex = 4;
this.groupBox1.TabStop = false;
this.groupBox1.Text = "ДИСЦИПЛИНА";
//
// cmb_Subject
//
this.cmb_Subject.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList;
this.cmb_Subject.FormattingEnabled = true;
this.cmb_Subject.Location = new System.Drawing.Point(6, 21);
this.cmb_Subject.Name = "cmb_Subject";
this.cmb_Subject.Size = new System.Drawing.Size(202, 24);
this.cmb_Subject.TabIndex = 1;
this.cmb_Subject.SelectedIndexChanged += new
System.EventHandler(this.cmb_Subject_SelectedIndexChanged);
//
// groupBox3
//
this.groupBox3.Controls.Add(this.cmb_Doc);
this.groupBox3.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.groupBox3.Location = new System.Drawing.Point(460, 12);
this.groupBox3.Name = "groupBox3";
this.groupBox3.Size = new System.Drawing.Size(218, 51);
this.groupBox3.TabIndex = 3;
this.groupBox3.TabStop = false;
this.groupBox3.Text = "Документ";
//
// cmb_Doc
//

```

```

        this.cmb_Doc.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList;
        this.cmb_Doc.FormattingEnabled = true;
        this.cmb_Doc.Location = new System.Drawing.Point(6, 21);
        this.cmb_Doc.Name = "cmb_Doc";
        this.cmb_Doc.Size = new System.Drawing.Size(202, 24);
        this.cmb_Doc.TabIndex = 1;
        //
        // rtb_Main
        //
        this.rtb_Main.Location = new System.Drawing.Point(12, 69);
        this.rtb_Main.Name = "rtb_Main";
        this.rtb_Main.ReadOnly = true;
        this.rtb_Main.ScrollBars = System.Windows.Forms.RichTextBoxScrollBars.ForcedBoth;
        this.rtb_Main.Size = new System.Drawing.Size(786, 556);
        this.rtb_Main.TabIndex = 5;
        this.rtb_Main.Text = "";
        this.rtb_Main.SelectionChanged += new
System.EventHandler(this.rtb_Main_SelectionChanged);
        this.rtb_Main.MouseClick += new
System.Windows.Forms.MouseEventHandler(this.rtb_Main_MouseClick);
        this.rtb_Main.KeyPress += new
System.Windows.Forms.KeyPressEventHandler(this.rtb_Main_KeyPress);
        //
        // button1
        //
        this.button1.Location = new System.Drawing.Point(684, 19);
        this.button1.Name = "button1";
        this.button1.Size = new System.Drawing.Size(114, 44);
        this.button1.TabIndex = 6;
        this.button1.Text = "Открыть";
        this.button1.UseVisualStyleBackColor = true;
        this.button1.Click += new System.EventHandler(this.button1_Click);
        //
        // F_Help
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.AutoSizeMode = System.Windows.Forms.AutoSizeMode.GrowAndShrink;
        this.ClientSize = new System.Drawing.Size(809, 630);
        this.Controls.Add(this.button1);
        this.Controls.Add(this.rtb_Main);
        this.Controls.Add(this.groupBox3);
        this.Controls.Add(this.groupBox2);
        this.Controls.Add(this.groupBox1);
        this.MaximizeBox = false;
        this.MinimizeBox = false;
        this.Name = "F_Help";
        this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
        this.Load += new System.EventHandler(this.F_Help_Load);
        this.groupBox2.ResumeLayout(false);
        this.groupBox1.ResumeLayout(false);

```

```

        this.groupBox3.ResumeLayout(false);
        this.ResumeLayout(false);

    }

#endregion

private System.Windows.Forms.GroupBox groupBox2;
private System.Windows.Forms.ComboBox cmb_Topic;
private System.Windows.Forms.GroupBox groupBox1;
private System.Windows.Forms.ComboBox cmb_Subject;
private System.Windows.Forms.GroupBox groupBox3;
private System.Windows.Forms.ComboBox cmb_Doc;
private System.Windows.Forms.RichTextBox rtb_Main;
private System.Windows.Forms.Button button1;
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Testing
{
    public partial class F_Help : Form
    {
        DataTable dtSubject = new DataTable();
        DataTable dtTopic = new DataTable();
        DataTable dtDoc = new DataTable();

        Dictionary<String, int> dicTopic = new Dictionary<string, int>();
        Dictionary<String, int> dicDoc = new Dictionary<string, int>();

        public F_Help()
        {
            InitializeComponent();
        }

        private void F_Help_Load(object sender, EventArgs e)
        {
            dtSubject = ListenServer.wcfClient.GetSimpleSubject();
            dtTopic = ListenServer.wcfClient.GetSimpleTopic();
            dtDoc = ListenServer.wcfClient.GetSimpleHelp();

            cmb_Subject.DataSource = dtSubject;

```

```

        cmb_Subject.DisplayMember = "Дисциплина";
    }
    void ChangeTopic()
    {
        cmb_Topic.Items.Clear();
        dicTopic.Clear();
        for (int i = 0; i < dtTopic.Rows.Count; i++)
        {
            if (dtTopic.Rows[i][1].ToString() ==
dtSubject.Rows[cmb_Subject.SelectedIndex][0].ToString())
            {
                string a = dtTopic.Rows[i][0].ToString();
                string b = dtTopic.Rows[i][2].ToString();
                dicTopic.Add(dtTopic.Rows[i][2].ToString(),
Convert.ToInt32(dtTopic.Rows[i][0]));
                cmb_Topic.Items.Add(dtTopic.Rows[i][2].ToString());
            }
        }
        if (dicTopic.Count > 0) cmb_Topic.SelectedIndex = 0;
        // dgv_Main.Columns[0].Visible = false;
    }

    void ChangeDoc()
    {
        cmb_Doc.Items.Clear();
        dicDoc.Clear();
        dicTopic.Clear();
        for (int i = 0; i < dtDoc.Rows.Count; i++)
        {
            if (dtDoc.Rows[i][1].ToString() ==
dtTopic.Rows[cmb_Topic.SelectedIndex][0].ToString())
            {
                dicDoc.Add(dtDoc.Rows[i][2].ToString(), Convert.ToInt32(dtDoc.Rows[i][0]));
                cmb_Doc.Items.Add(dtDoc.Rows[i][2].ToString());
            }
        }
        if (dicDoc.Count > 0) cmb_Doc.SelectedIndex = 0;
        // dgv_Main.Columns[0].Visible = false;
    }

    private void cmb_Subject_SelectedIndexChanged(object sender, EventArgs e)
    {
        // try
        {
            ChangeTopic();
        }
        // catch { }
    }

    private void cmb_Topic_SelectedIndexChanged(object sender, EventArgs e)
    {

```

```

        ChangeDoc();
    }
    String Path = "";
    private void button1_Click(object sender, EventArgs e)
    {
        // rtb_Main.Text = OpenDocxFile(@"novyj_test_po_pb");
        // rtb_Main.LoadFile(@"novyj_test_po_pb.docx", RichTextBoxStreamType.PlainText);
        // OpenDocxFile();

        int a = 0;
        Path = System.Reflection.Assembly.GetExecutingAssembly().Location;
        for (int i = 0; i < dtDoc.Rows.Count; i++)
            if (Convert.ToInt32(dtDoc.Rows[i][0]) == dicDoc[cmb_Doc.Text]) a = i;
        string str = "Справочники\\" + cmb_Subject.Text + "\\" + cmb_Topic.Text + "\\" +
cmb_Doc.Text + dtDoc.Rows[a][3];
        //Path = Path.Replace("Testing.exe", "novyj_test_po_pb.docx");
        Path = Path.Replace("Testing.exe", str);
        try
        {
            OpenDocxFile();
        }
        catch { MessageBox.Show("Файл не
найден", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error); }
    }
    void OpenDocxFile()
    {
        Microsoft.Office.Interop.Word.Application app = new
Microsoft.Office.Interop.Word.Application();//процесс ворда
        Object docxFileName = Path;//имя файла
        Object missing = Type.Missing;
        //открыли документ
        app.Documents.Open(ref docxFileName, ref missing,
            ref missing, ref missing, ref missing,
            ref missing, ref missing, ref missing, ref missing,
            ref missing, ref missing);
        //путь к папке с временными файлами
        string temp = System.IO.Path.GetTempPath();
        //для передачи параметров при пересохранении
        Object lookComments = false;
        Object password = String.Empty;
        Object AddToRecentFiles = true;
        Object WritePassword = String.Empty;
        Object ReadOnlyRecommended = false;
        Object EmbedTrueTypeFonts = false;
        Object SaveFormsData = false;
        Object SaveAsAOCELetter = false;
        //имя файла без расширения
        Object rtfFileName = "novyj_test_po_pb.docx";
        //создали рандом

```



```

Random random = new Random();
//проверяем есть ли файл с таким именем
while (System.IO.File.Exists(rtfFileName + ".rtf"))
    //генерируем случайное имя файла
    rtfFileName += random.Next(0, 9).ToString();
//формат RTF
Object wdFormatRTF = Microsoft.Office.Interop.Word.WdSaveFormat.wdFormatRTF;
//приписали расширение
rtfFileName += ".rtf";
//приписали путь к временным файлам
rtfFileName = temp + rtfFileName;
//пересохранили
app.ActiveDocument.SaveAs(ref rtfFileName,
    ref wdFormatRTF, ref lookComments, ref password, ref AddToRecentFiles, ref
WritePassword, ref ReadOnlyRecommended,
    ref EmbedTrueTypeFonts, ref missing, ref SaveFormsData, ref SaveAsAOCELetter,
ref missing,
    ref missing, ref missing, ref missing, ref missing);
//переменная
Object @false = false;
//закрыли текущий документ
app.ActiveDocument.Close(ref @false, ref missing, ref missing);
//вышли из ворда
app.Quit(ref @false, ref missing, ref missing);
//прочли файл
rtb_Main.LoadFile((String)rtfFileName);
}

private void rtb_Main_KeyPress(object sender, KeyPressEventArgs e)
{
}

private void rtb_Main_MouseClick(object sender, MouseEventArgs e)
{
}

private void rtb_Main_SelectionChanged(object sender, EventArgs e)
{
    rtb_Main.SelectionLength = 0;
}
}
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;

```

```

using System.Linq;
using System.Runtime.Serialization;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Xml;

namespace Testing
{
    public partial class F_AddTest : Form
    {
        int testIndex = 0;

        public F_AddTest()
        {
            InitializeComponent();
            tb_Topic.Text = FormDialog.Topic;
            tb_TestName.Text = FormDialog.NameOfTest;

            ButtonsImages(btn_Close, "Close.png");
            ButtonsImages(btn_Next, "Login.png");
        }

        private void btn_Next_Click(object sender, EventArgs e)
        {
            AddAnswer();
            TestPanelReload();
        }

        private void F_AddTest_Load(object sender, EventArgs e)
        {
            tb_Topic.Text = FormDialog.Topic;
            tb_TestName.Text = FormDialog.NameOfTest;
            tb_Time.Text = FormDialog.Time.ToString() + " м";
            timer1.Interval = 1000 * 4;
            timer1.Start();

            TestPanelReload();
        }

        void ButtonsImages(Button button1, String image)
        {
            button1.Image = Image.FromFile(@"images\" + image + ".png", false);
            //button1.BackgroundImageLayout = ImageLayout.Zoom;
            // Align the image and text on the button.
            button1.ImageAlign = ContentAlignment.MiddleLeft;
            button1.TextAlign = ContentAlignment.MiddleRight;
            // Give the button a flat appearance.
            button1.FlatStyle = FlatStyle.Flat;
        }
    }
    /// <summary>
    /// Обновляет расположение панелей

```

```

/// </summary>
void TestPanelReload()
{
    if (testIndex < FormDialog.dtTests.Rows.Count)
    {
        rbtn_D.Visible = true;
        tb_QuestIndex.Text = testIndex.ToString();
        if (FormDialog.dtTests.Rows[testIndex][1].ToString() == "B-To")
        {
            p_Q1.Location = new Point(12, 38);

            p_Q2.Location = new Point(800, 800);
            p_Q3.Location = new Point(800, 800);

            p_Q1.Controls.Add(lb_CorrectAnswer);
            p_Q1.Controls.Add(p_Answer);

            lb_CorrectAnswer.Location = new Point(220, 329);
            lb_CorrectAnswer.BringToFront();

            p_Answer.Location = new Point(187, 348);
            p_Answer.BringToFront();
        }
        if (FormDialog.dtTests.Rows[testIndex][1].ToString() == "B-K")
        {
            p_Q2.Location = new Point(12, 38);

            p_Q1.Location = new Point(800, 800);
            p_Q3.Location = new Point(800, 800);

            p_Q2.Controls.Add(lb_CorrectAnswer);
            p_Q2.Controls.Add(p_Answer);

            lb_CorrectAnswer.Location = new Point(220, 329);
            lb_CorrectAnswer.BringToFront();

            p_Answer.Location = new Point(187, 348);
            p_Answer.BringToFront();
        }
        if (FormDialog.dtTests.Rows[testIndex][1].ToString() == "K-To")
        {
            p_Q3.Location = new Point(12, 38);

            p_Q1.Location = new Point(800, 800);
            p_Q2.Location = new Point(800, 800);

            p_Q3.Controls.Add(lb_CorrectAnswer);
            p_Q3.Controls.Add(p_Answer);

            lb_CorrectAnswer.Location = new Point(220, 329);

```

```

        lb_CorrectAnswer.BringToFront();

        p_Answer.Location = new Point(187, 348);
        p_Answer.BringToFront();

        rbtn_D.Visible = false;
    }
    TestQuestionsReload();
    testIndex++;
}
}

/// <summary>
/// Обновляет вопросы
/// </summary>
void TestQuestionsReload()
{
    if (testIndex < FormDialog.dtTests.Rows.Count)
    {
        if (FormDialog.dtTests.Rows[testIndex][1].ToString() == "B-To")
        {
            tb_Question1.Text = FormDialog.dtTests.Rows[testIndex][3].ToString();
            tb_A1.Text = FormDialog.dtTests.Rows[testIndex][4].ToString();
            tb_B1.Text = FormDialog.dtTests.Rows[testIndex][5].ToString();
            tb_C1.Text = FormDialog.dtTests.Rows[testIndex][6].ToString();
            tb_D1.Text = FormDialog.dtTests.Rows[testIndex][7].ToString();
        }
        if (FormDialog.dtTests.Rows[testIndex][1].ToString() == "B-K")
        {
            tb_Question2.Text = FormDialog.dtTests.Rows[testIndex][3].ToString();
            Deserialize(ref pb_A2, FormDialog.dtTests.Rows[testIndex][4].ToString());
            Deserialize(ref pb_B2, FormDialog.dtTests.Rows[testIndex][5].ToString());
            Deserialize(ref pb_C2, FormDialog.dtTests.Rows[testIndex][6].ToString());
        }
        if (FormDialog.dtTests.Rows[testIndex][1].ToString() == "K-To")
        {
            Deserialize(ref pb_Question2, FormDialog.dtTests.Rows[testIndex][3].ToString());
            tb_A3.Text = FormDialog.dtTests.Rows[testIndex][4].ToString();
            tb_B3.Text = FormDialog.dtTests.Rows[testIndex][5].ToString();
            tb_C3.Text = FormDialog.dtTests.Rows[testIndex][6].ToString();
            tb_D3.Text = FormDialog.dtTests.Rows[testIndex][7].ToString();
        }
    }
}

/// <summary>
/// десериализует картинки
/// </summary>
/// <param name="pb"></param>
/// <param name="strXML"></param>
void Deserialize(ref PictureBox pb, String strXML)

```

```

{
    File.WriteAllText("file.xml", strXML, Encoding.Default);
    // Десериализуем
    DataContractSerializer serializer = new DataContractSerializer(typeof(SerializerXML));
    SerializerXML myClass;
    using (XmlReader writer = XmlReader.Create("file.xml"))
    {
        myClass = (SerializerXML)serializer.ReadObject(writer);
    }

    // Считываем данные из полученного экземпляра
    pb.Image = myClass.MyBitmap;
    pb.BackgroundImageLayout = ImageLayout.Zoom;
    pb.SizeMode = PictureBoxSizeMode.Zoom;
}

void AddAnswer()
{
    if (testIndex < FormDialog.dtTests.Rows.Count)
    {
        if (rbtn_A.Checked == true) FormDialog.testAnswer += "А-";
        if (rbtn_B.Checked == true) FormDialog.testAnswer += "Б-";
        if (rbtn_C.Checked == true) FormDialog.testAnswer += "В-";
        if (rbtn_D.Checked == true) FormDialog.testAnswer += "Г-";

        rbtn_A.Checked = false;
        rbtn_B.Checked = false;
        rbtn_C.Checked = false;
        rbtn_D.Checked = false;
    } else
    {
        if (rbtn_A.Checked == true) FormDialog.testAnswer += "А-";
        if (rbtn_B.Checked == true) FormDialog.testAnswer += "Б-";
        if (rbtn_C.Checked == true) FormDialog.testAnswer += "В-";
        if (rbtn_D.Checked == true) FormDialog.testAnswer += "Г-";

        rbtn_A.Checked = false;
        rbtn_B.Checked = false;
        rbtn_C.Checked = false;
        rbtn_D.Checked = false;
        //End test
    }
    try
    {
        FormDialog.testResult =
ListenServer.wcfClient.SaveAnswer(Convert.ToInt32(FormDialog.dtUser.Rows[0][0]),
Convert.ToInt32(FormDialog.dtTests.Rows[0][2]), FormDialog.testAnswer);
        MessageBox.Show("Тест успешно сохранен", "Информация",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        if (FormDialog.F_TestResult.Created == true) FormDialog.F_TestResult.Close();
        FormDialog.F_TestResult = new F_TestResult();
        FormDialog.F_TestResult.Show();
        Close();
    }
}

```

```

    }
    catch
    {
    }
}

private void timer1_Tick(object sender, EventArgs e)
{
    if (FormDialog.Time > 0)
    {
        // Display the new time left
        // by updating the Time Left label.
        FormDialog.Time = FormDialog.Time - 1;
        tb_Time.Text = FormDialog.Time.ToString() + " м";

    }
    else
    {
        // If the user ran out of time, stop the timer, show
        // a MessageBox, and fill in the answers.

        MessageBox.Show("Вы не закончили тест вовремя! Время вышло!");
        // Сохраняем результаты
        for (int i = testIndex; i < FormDialog.dtTests.Rows.Count; i++)
        {
            FormDialog.testAnswer += "0-";
        }
        try
        {
            FormDialog.testAnswer += "0-";
            FormDialog.testResult =
ListenServer.wcfClient.SaveAnswer(Convert.ToInt32(FormDialog.dtUser.Rows[0][0]),
Convert.ToInt32(FormDialog.dtTests.Rows[0][2]), FormDialog.testAnswer);
            // MessageBox.Show("Тест успешно сохранен", "Информация",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            if (FormDialog.F_TestResult.Created == true) FormDialog.F_TestResult.Close();
            FormDialog.F_TestResult = new F_TestResult();
            FormDialog.F_TestResult.Show();
            timer1.Stop();
            Close();
        }
        catch
        {
        }

        this.Close();
    }
}

```

```

private void F_AddTest_FormClosed(object sender, FormClosedEventArgs e)
{
    if (testIndex < FormDialog.dtTests.Rows.Count)
    {
        timer1.Stop();
        FormDialog.NameOfTest = "";
        FormDialog.testAnswer = "";
        FormDialog.testResult = "";
        FormDialog.Topic = "";
        FormDialog.dtTests = new DataTable();
        FormDialog.dtUser = new DataTable();
        ListenServer.wcfClient.DeleteConnectUsers(FormDialog.Login);
        FormDialog.F_Main.Visible = true;
    }
}

private void btn_Close_Click(object sender, EventArgs e)
{
    Close();
}

}
}

using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Runtime.Serialization;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Xml;

namespace Testing
{
    [Serializable]
    [DataContract]
    class SerializerXML
    {
        [DataMember]
        public Bitmap MyBitmap { get; set; }
        [DataMember]
        public string Name { get; set; }
    }
}

```

```

using System;

```

```

using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Testing
{
    static class Program
    {
        /// <summary>
        /// Главная точка входа для приложения.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new F_Logo());
        }
    }
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.ServiceModel;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

```

```

namespace Testing
{
    static class ListenServer
    {

        public static IFunctional wcfClient;
        static BasicHttpBinding bhb = new BasicHttpBinding();
        //MaxReceivedMessageSize
        public static void ConntctToServer()
        {
            try
            {
                bhb.MaxReceivedMessageSize = Int32.MaxValue;
                ChannelFactory<IFunctional> myChannelFactory;
                myChannelFactory = new ChannelFactory<IFunctional>(
                    bhb,
                    new EndpointAddress("http://localhost:7792/Testing"));
                wcfClient = myChannelFactory.CreateChannel();
            }
            catch
            {

```



```

        MessageBox.Show("Связь с сервером не была установлена", "Ошибка
подключения", MessageBoxButtons.RetryCancel, MessageBoxIcon.Error);
    }
}
}
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.ServiceModel;
using System.Data;

```

```

namespace Testing
{
    [ServiceContract]
    public interface IFunctional
    {
        [OperationContract]
        Boolean Registration(String Pass, String Name, String Surname, Int32 rank);
        [OperationContract]
        DataTable GetBranch();
        [OperationContract]
        DataTable GetGroup();
        [OperationContract]
        DataTable GetUser(String Name, String Surname);
        [OperationContract]
        DataTable GetTests();
        [OperationContract]
        String SaveAnswer(Int32 User, Int32 Test, String Answer);
        [OperationContract]
        String GetTopic(int Inex);
        [OperationContract]
        String GetNameOfTest(int Index);
        [OperationContract]
        void getConnectUsers(String Login);
        [OperationContract]
        int GetTime();
        [OperationContract]
        void DeleteConnectUsers(String Login);
        [OperationContract]
        DataTable GetSimpleSubject();
        [OperationContract]
        DataTable GetSimpleTopic();
        [OperationContract]
        DataTable GetSimpleHelp();
    }
}

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Testing
{
    public partial class F_EditTests : Form
    {
        DataTable dtSubject = new DataTable();
        DataTable dtTopic = new DataTable();
        DataTable dtTests = new DataTable();

        Dictionary<String, int> dicTopic = new Dictionary<string, int>();

        F_AddTest F_AddTest = new F_AddTest();

        public F_EditTests()
        {
            InitializeComponent();
        }

        private void btn_AddTest_Click(object sender, EventArgs e)
        {
            FormDialog.IdOfTopic = dicTopic[cmb_Topic.Text]; //ID темы

            if (F_AddTest.Created == true) F_AddTest.Close();
            F_AddTest = new F_AddTest();
            F_AddTest.Show();
        }

        private void F_EditTests_Load(object sender, EventArgs e)
        {
            dtSubject = GetTabele.Get("SELECT * FROM Дисциплина");
            dtTopic = GetTabele.Get("SELECT * FROM Тема");
            dtTests = GetTabele.Get("SELECT * FROM Тесты");

            cmb_Subject.DataSource = dtSubject;
            cmb_Subject.DisplayMember = "Дисциплина";
            GetActiveTest();
            dgv_Main.Columns[0].Visible = false;

            FormDialog.lb_Activetest = lb_ActiveTest;
        }
    }
}

```

```

String activeTest;
//Изменяет значение комбобокса с темами в зависимости от значения комбобокса
Дисциплин
void ChangeTopic()
{
    cmb_Topic.Items.Clear();
    dicTopic.Clear();
    for (int i = 0; i < dtTopic.Rows.Count; i++)
    {
        if (dtTopic.Rows[i][1].ToString() ==
dtSubject.Rows[cmb_Subject.SelectedIndex][0].ToString())
        {
            dicTopic.Add(dtTopic.Rows[i][2].ToString(), (int)dtTopic.Rows[i][0]);
            cmb_Topic.Items.Add(dtTopic.Rows[i][2].ToString());
        }
    }
    if (dicTopic.Count > 0) cmb_Topic.SelectedIndex = 0;
    dgv_Main.Columns[0].Visible = false;
}
DataTable ChangeDataGrid()
{
    DataRow dr;
    DataTable dt = new DataTable();
    dt.Columns.Add("Код");
    dt.Columns.Add("№ теста");
    dt.Columns.Add("Время прохождения");

    ;
    for (int i = 0; i < dtTests.Rows.Count; i++)
    {
        try
        {
            if (dicTopic[cmb_Topic.Text] == Convert.ToInt32(dtTests.Rows[i][1]))
            {
                dr = dt.NewRow();
                dr["Код"] = dtTests.Rows[i][0];
                dr["№ теста"] = dtTests.Rows[i][2];
                dr["Время прохождения"] = dtTests.Rows[i][3];
                dt.Rows.Add(dr);
            }
        }
        catch { }
    }
    return dt;
}

private void cmb_Subject_SelectedIndexChanged(object sender, EventArgs e)
{
    ChangeTopic();
}

private void cmb_Topic_SelectedIndexChanged(object sender, EventArgs e)

```

```

    {
        dgv_Main.DataSource = ChangeDataGrid();
    }

private void btn_AddActiveTest_Click(object sender, EventArgs e)
{
    SetActiveTest();
    GetActiveTest();
}
void SetActiveTest()
{
    for (int i = 0; i < dtTests.Rows.Count; i++)
    {
        if (dtTests.Rows[i][2].ToString() == dgv_Main[1,
dgv_Main.CurrentRow.Index].Value.ToString())
            File.WriteAllText("Type.kay", dtTests.Rows[i][0].ToString(), Encoding.Default);
    }
}
void GetActiveTest()
{
    string strfilename = @"Type.kay";
    FileStream stream = new FileStream(strfilename, FileMode.Open);
    StreamReader reader = new StreamReader(stream);
    String FILE = reader.ReadToEnd();
    reader.Close();

    for (int i = 0; i < dtTests.Rows.Count; i++)
    {
        if (dtTests.Rows[i][0].ToString() == FILE)
            lb_ActiveTest.Text = "Активный тест: " + dtTests.Rows[i][2].ToString();
    }
}
F_AddRndTest F_AddRndTest = new F_AddRndTest();
private void btn_CreatureRNDTest_Click(object sender, EventArgs e)
{
    if (F_AddRndTest.Created == true) F_AddRndTest.Close();
    F_AddRndTest = new F_AddRndTest();
    F_AddRndTest.Show();
}

private void btn_DeleteTest_Click(object sender, EventArgs e)
{
    DialogResult dialogRes = MessageBox.Show("Выполнить операцию удаления?", "",
MessageBoxButtons.OKCancel, MessageBoxIcon.Question);
    if (dialogRes == System.Windows.Forms.DialogResult.OK)
    {
        GetTabele.Get("DELETE * FROM Тесты WHERE Код=" + dgv_Main[0,
dgv_Main.CurrentRow.Index].Value.ToString());
        dtTests = GetTabele.Get("SELECT * FROM Тесты");
        ChangeDataGrid();
    }
}

```

```

    }

    private void tb_EditTime_Click(object sender, EventArgs e)
    {
        if (tb_Time.Text.Length > 0)
        {
            GetTabele.Get("UPDATE Тесты SET Время=" + tb_Time.Text);
            dtTests = GetTabele.Get("SELECT * FROM Тесты");
            dgv_Main.DataSource = ChangeDataGrid();
        }
        else MessageBox.Show("Значение поля времени не может быть пустым", "",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
    }

    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Testing
{
    public partial class F_AddUser : Form
    {
        DataTable dtBrunch = new DataTable();
        DataTable dtGroup = new DataTable();

        Functional Functional = new Functional();
        public F_AddUser()
        {
            InitializeComponent();
        }

        private void btn_Reg_Click(object sender, EventArgs e)
        {
            if (cmb_Groups.SelectedIndex >= 0 && tb_Surname.Text.Length > 0
                && tb_Name.Text.Length > 0 && tb_Pass.Text.Length > 0)
            {
                if (tb_Pass.Text == tb_ConPass.Text)
                {
                    Functional.Registration(tb_Pass.Text, tb_Name.Text, tb_Surname.Text,
                        Convert.ToInt32(dtGroup.Rows[cmb_Groups.SelectedIndex][0]));
                    MessageBox.Show("Регистарция прошла успешно");
                    tb_Surname.Clear();
                }
            }
        }
    }
}

```

```

        tb_Name.Clear();
        tb_Pass.Clear();
        tb_ConPass.Clear();

    }
    else MessageBox.Show("Пароли не совпадают", "Ошибка",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else MessageBox.Show("Не все поля заполнены", "Ошибка",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }

private void F_AddUser_Load(object sender, EventArgs e)
{
    dtGroup = Functional.GetGroup();

    cmb_Groups.DataSource = dtGroup;
    cmb_Groups.DisplayMember = "Должность";

}

}
}

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

```

```

namespace Testing
{
    public partial class F_AddRndTest : Form
    {
        DataTable dtAllTests = new DataTable();
        List<int> indexOfRNDQuestions = new List<int>();

        Random rnd = new Random();

        public F_AddRndTest()
        {
            InitializeComponent();
        }

        private void btn_AddRndTest_Click(object sender, EventArgs e)
        {

```

```

//try
{
    bool _bool = false;
    int rndNumb = 0;
    for (int i = indexOfRNDQuestions.Count; i <
Convert.ToInt32(tb_QuestionsCount.Text); )
    {
        int a = Convert.ToInt32(dtAllTests.Rows[0][0].ToString());
        int b = Convert.ToInt32(dtAllTests.Rows[dtAllTests.Rows.Count -
1][0].ToString());
        rndNumb = rnd.Next(Convert.ToInt32(dtAllTests.Rows[0][0]),
Convert.ToInt32(dtAllTests.Rows[dtAllTests.Rows.Count - 1][0]) + 1);
        for (int j = 0; j < indexOfRNDQuestions.Count; j++)
            if (indexOfRNDQuestions[j] == rndNumb)
                _bool = true;
        // else
        // {
        if (_bool == false)
            for (int k = 0; k < dtAllTests.Rows.Count; k++)
                if (Convert.ToInt32(dtAllTests.Rows[k][0]) != rndNumb)
                    _bool = true;
                else
                {
                    _bool = false;
                    break;
                }
            // }
        if (_bool == false)
            indexOfRNDQuestions.Add(rndNumb);
        _bool = false;
        i = indexOfRNDQuestions.Count;
    }
    DataTable dtRNDTests = new DataTable();
    for (int i = 0; i < indexOfRNDQuestions.Count; i++)
    {
        DataTable _dt = new DataTable();
        _dt = GetTabele.Get("SELECT * FROM Вопросы WHERE Код=" +
indexOfRNDQuestions[i]);
        dtRNDTests.Merge(_dt);
    }
    //dataGridView1.DataSource = dtRNDTests;
    //Инициализация таблицы вручную
    DataTable dt = new DataTable("Вопросы");
    dt.Columns.AddRange(new DataColumn[9] {new DataColumn("Вид"),
new DataColumn("Тест"), new DataColumn("Вопрос"), new DataColumn("Ответ А"),
new DataColumn("Ответ Б"), new DataColumn("Ответ В"), new DataColumn("Ответ
Г"),
new DataColumn("Правильный ответ"), new DataColumn("Балл")});
    for (int i = 0; i < dtRNDTests.Rows.Count; i++)
        dt.Rows.Add(dtRNDTests.Rows[i][1], dtRNDTests.Rows[i][2],
dtRNDTests.Rows[i][3],

```

```

        dtRNDTests.Rows[i][4], dtRNDTests.Rows[i][5], dtRNDTests.Rows[i][6],
dtRNDTests.Rows[i][7],
        dtRNDTests.Rows[i][8], dtRNDTests.Rows[i][9]);
    String Query = "";
    GetTabele.Get("INSERT INTO Тесты (Тема, Название, Время) VALUES (" +
GetTabele.Get("SELECT Код FROM Тема WHERE Тема='-").Rows[0][0] + ", 'RND', " +
tb_Time.Text + ")");
    for (int i = 0; i < dt.Rows.Count; i++)
    {
        GetTabele.Get("INSERT INTO Вопросы (Вид, Тест, Вопрос, [Ответ А], [Ответ
Б], [Ответ В], [Ответ Г], [Правильный ответ], Балл, Ст) VALUES (" + "" + dt.Rows[i][0] +
", " + GetTabele.Get("(SELECT Код FROM Тесты WHERE Название='RND') ").Rows[0][0] +
", " + dt.Rows[i][2] + ", " + dt.Rows[i][3] + ", " +
        + dt.Rows[i][4] + ", " + dt.Rows[i][5] + ", " + dt.Rows[i][6] + ", " +
dt.Rows[i][7] + ", " + dt.Rows[i][8] + ", true)");
    }
    // Query = Query.TrimEnd(',');
    // GetTabele.Get("INSERT INTO Вопросы (Вид, Тест, Вопрос, [Ответ А], [Ответ
Б], [Ответ В], [Ответ Г], [Правильный ответ], Балл) VALUES " + Query + "");
    SetActiveTest();
    GetActiveTest();
    MessageBox.Show("Тест успешно сгенерирован");
    Close();
}
// catch { }
}

private void F_AddRndTest_Load(object sender, EventArgs e)
{
    GetTabele.Get("UPDATE Тесты SET Название='RND " +
DateTime.Now.Date.ToString().Replace("0:00:00", "") + "/" + DateTime.Now.Minute + "
WHERE Название='RND'");
    //GetTabele.Get("DELETE * FROM Тема WHERE Тема='-");
    //GetTabele.Get("DELETE * FROM Дисциплина WHERE Дисциплина='-");
    // GetTabele.Get("INSERT INTO Дисциплина (Дисциплина) VALUES ('-");
    //// int IndexOfRNDTopic
    // GetTabele.Get("INSERT INTO Тема (Дисциплина, Тема) VALUES (" +
GetTabele.Get("(SELECT Код FROM Дисциплина WHERE Дисциплина='-")").Rows[0][0] +
",'-");
}
// GetTabele.Get("");
dtAllTests = GetTabele.Get("SELECT * FROM Вопросы WHERE Ст=false");

int a = Convert.ToInt32(dtAllTests.Rows[0][0].ToString());
int b = Convert.ToInt32(dtAllTests.Rows[dtAllTests.Rows.Count-1][0].ToString());
}
void SetActiveTest()
{
    File.WriteAllText("Type.kay", GetTabele.Get("(SELECT Код FROM Тесты WHERE
Название='RND') ").Rows[0][0].ToString(), Encoding.Default);
}
void GetActiveTest()

```



```

    {
        DataTable dtTests = new DataTable();
        dtTests = GetTabele.Get("SELECT * FROM Тесты");
        string strfilename = @"Type.kay";
        FileStream stream = new FileStream(strfilename, FileMode.Open);
        StreamReader reader = new StreamReader(stream);
        String FILE = reader.ReadToEnd();
        reader.Close();

        for (int i = 0; i < dtTests.Rows.Count; i++)
        {
            if (dtTests.Rows[i][0].ToString() == FILE)
                FormDialog.lb_Activetest.Text = "Активный тест: " +
dtTests.Rows[i][2].ToString();
        }

    }

private void tb_Time_KeyPress(object sender, KeyPressEventArgs e)
{
    char l = e.KeyChar;
    if ((l < '0' || l > '9') && l != '\b')
    {
        e.Handled = true;
    }
}

private void btn_Cansel_Click(object sender, EventArgs e)
{
    Close();
}
}
}

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

```

```

namespace Testing
{
    public partial class F_AddAdmin : Form
    {
        public F_AddAdmin()
        {
            InitializeComponent();
        }
    }
}

```

```

private void btn_Reg_Click_1(object sender, EventArgs e)
{
    if (tb_Surname.Text.Length > 0 && tb_Name.Text.Length > 0 && tb_Pass.Text.Length
>0)
    {
        if (tb_Pass.Text == tb_ConPass.Text)
        {
            try
            {
                GetTabele.Get("INSERT INTO Пользователи (Логин, Пароль, Имя, Фамилия,
Старус) VALUES (" +
                tb_Surname.Text + tb_Name.Text + ", " + tb_Pass.Text + ", " + tb_Name.Text + ", " +
                tb_Surname.Text + ", true)");
                MessageBox.Show("Регистарция прошла успешно");
                tb_Surname.Clear();
                tb_Name.Clear();
                tb_Pass.Clear();

                tb_ConPass.Clear();
            }
            catch { MessageBox.Show("Ошибка регистрации", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Error); }
        }
        else MessageBox.Show("Пароли не совпадают", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else MessageBox.Show("Не все поля заполнены", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Error);
}

private void tb_Surname_KeyPress(object sender, KeyPressEventArgs e)
{
    char l = e.KeyChar;
    if ((l < 'a' || l > 'я') && l != '\b' && (l < 'A' || l > 'Я'))
    {
        e.Handled = true;
    }
}
}
}

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

```

```

namespace Testing
{
    public partial class F_Logo : Form
    {
        public F_Logo()
        {
            InitializeComponent();
            FormDialog.F_Logo = this;
        }

        private void pictureBox1_Click(object sender, EventArgs e)
        {

        }

        private void F_Logo_Load(object sender, EventArgs e)
        {
            pictureBox1.ImageLocation = (@"заставка.jpg");
            pictureBox1.BackgroundImageLayout = ImageLayout.Stretch;
            pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;
            // F_Main f_main = new F_Main();
            //f_main.Show();
            // new System.Threading.Timer(state => { f_main.Show(); Close(); }, null, 1000, 0);
            timer1.Interval = 2000;
            timer1.Start();
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            F_Main f_main = new F_Main();
            f_main.Show();
            this.Visible = false;
            timer1.Stop();
        }
    }
}

```

```

DataTable ChangeDataGrid()
{
    DataRow dr;
    DataTable dt = new DataTable();
    dt.Columns.Add("Код");
    dt.Columns.Add("№ теста");
    dt.Columns.Add("Время прохождения");

    ;
    for (int i = 0; i < dtTests.Rows.Count; i++)
    {
        try
        {
            if (dicTopic[cmb_Topic.Text] == Convert.ToInt32(dtTests.Rows[i][1]))

```

```

        {
            dr = dt.NewRow();
            dr["Код"] = dtTests.Rows[i][0];
            dr["№ теста"] = dtTests.Rows[i][2];
            dr["Время прохождения"] = dtTests.Rows[i][3];
            dt.Rows.Add(dr);
        }
    }
    catch { }
}
return dt;
}

private void cmb_Subject_SelectedIndexChanged(object sender, EventArgs e)
{
    ChangeTopic();
}

private void cmb_Topic_SelectedIndexChanged(object sender, EventArgs e)
{
    dgv_Main.DataSource = ChangeDataGrid();
}

private void btn_AddActiveTest_Click(object sender, EventArgs e)
{
    SetActiveTest();
    GetActiveTest();
}

void SetActiveTest()
{
    for (int i = 0; i < dtTests.Rows.Count; i++)
    {
        if (dtTests.Rows[i][2].ToString() == dgv_Main[1,
dgv_Main.CurrentRow.Index].Value.ToString())
            File.WriteAllText("Type.kay", dtTests.Rows[i][0].ToString(), Encoding.Default);
    }
}

void GetActiveTest()
{
    string strfilename = @"Type.kay";
    FileStream stream = new FileStream(strfilename, FileMode.Open);
    StreamReader reader = new StreamReader(stream);
    String FILE = reader.ReadToEnd();
    reader.Close();

    for (int i = 0; i < dtTests.Rows.Count; i++)
    {
        if (dtTests.Rows[i][0].ToString() == FILE)
            lb_ActiveTest.Text = "Активный тест: " + dtTests.Rows[i][2].ToString();
    }
}
}

```

```

F_AddRndTest F_AddRndTest = new F_AddRndTest();
private void btn_CreatureRNDTest_Click(object sender, EventArgs e)
{
    if (F_AddRndTest.Created == true) F_AddRndTest.Close();
    F_AddRndTest = new F_AddRndTest();
    F_AddRndTest.Show();
}

private void btn_DeleteTest_Click(object sender, EventArgs e)
{
    DialogResult dialogRes = MessageBox.Show("Выполнить операцию удаления?", "",
    MessageBoxButtons.OKCancel, MessageBoxIcon.Question);
    if (dialogRes == System.Windows.Forms.DialogResult.OK)
    {
        GetTabele.Get("DELETE * FROM Тесты WHERE Код=" + dgv_Main[0,
    dgv_Main.CurrentRow.Index].Value.ToString());
        dtTests = GetTabele.Get("SELECT * FROM Тесты");
        ChangeDataGrid();
    }
}

private void tb_EditTime_Click(object sender, EventArgs e)
{
    if (tb_Time.Text.Length > 0)
    {
        GetTabele.Get("UPDATE Тесты SET Время=" + tb_Time.Text);
        dtTests = GetTabele.Get("SELECT * FROM Тесты");
        dgv_Main.DataSource = ChangeDataGrid();
    }
    else MessageBox.Show("Значение поля времени не может быть пустым", "",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
}

}

}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Testing
{
    public partial class F_AddUser : Form
    {
        DataTable dtBrunch = new DataTable();
        DataTable dtGroup = new DataTable();
    }
}

```

```

Functional Functional = new Functional();
public F_AddUser()
{
    InitializeComponent();
}

private void btn_Reg_Click(object sender, EventArgs e)
{
    if (cmb_Groups.SelectedIndex >= 0 && tb_Surname.Text.Length > 0
    && tb_Name.Text.Length > 0 && tb_Pass.Text.Length > 0)
    {
        if (tb_Pass.Text == tb_ConPass.Text)
        {
            Functional.Registration(tb_Pass.Text, tb_Name.Text, tb_Surname.Text,
            Convert.ToInt32(dtGroup.Rows[cmb_Groups.SelectedIndex][0]));
            MessageBox.Show("Регистарция прошла успешно");
            tb_Surname.Clear();
            tb_Name.Clear();
            tb_Pass.Clear();
            tb_ConPass.Clear();

        }
        else MessageBox.Show("Пароли не совпадают", "Ошибка",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else MessageBox.Show("Не все поля заполнены", "Ошибка",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
}

private void F_AddUser_Load(object sender, EventArgs e)
{
    dtGroup = Functional.GetGroup();

    cmb_Groups.DataSource = dtGroup;
    cmb_Groups.DisplayMember = "Должность";

}

}

}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;

```

```

using System.Threading.Tasks;
using System.Windows.Forms;

namespace Testing
{
    public partial class F_AddRndTest : Form
    {
        DataTable dtAllTests = new DataTable();
        List<int> indexOfRNDQuestions = new List<int>();

        Random rnd = new Random();

        public F_AddRndTest()
        {
            InitializeComponent();
        }

        private void btn_AddRndTest_Click(object sender, EventArgs e)
        {
            //try
            {
                bool _bool = false;
                int rndNumb = 0;
                for (int i = indexOfRNDQuestions.Count; i <
Convert.ToInt32(tb_QuestionsCount.Text); )
                {
                    int a = Convert.ToInt32(dtAllTests.Rows[0][0].ToString());
                    int b = Convert.ToInt32(dtAllTests.Rows[dtAllTests.Rows.Count -
1][0].ToString());
                    rndNumb = rnd.Next(Convert.ToInt32(dtAllTests.Rows[0][0]),
Convert.ToInt32(dtAllTests.Rows[dtAllTests.Rows.Count - 1][0]) + 1);
                    for (int j = 0; j < indexOfRNDQuestions.Count; j++)
                        if (indexOfRNDQuestions[j] == rndNumb)
                            _bool = true;
                    // else
                    // {
                    if (_bool == false)
                        for (int k = 0; k < dtAllTests.Rows.Count; k++)
                            if (Convert.ToInt32(dtAllTests.Rows[k][0]) != rndNumb)
                                _bool = true;
                            else
                            {
                                _bool = false;
                                break;
                            }
                    // }
                    if (_bool == false)
                        indexOfRNDQuestions.Add(rndNumb);
                    _bool = false;
                    i = indexOfRNDQuestions.Count;
                }
            }
        }
    }
}

```

```

        DataTable dtRNDTests = new DataTable();
        for (int i = 0; i < indexOfRNDQuestions.Count; i++)
        {
            DataTable _dt = new DataTable();
            _dt = GetTabele.Get("SELECT * FROM Вопросы WHERE Код=" +
indexOfRNDQuestions[i]);
            dtRNDTests.Merge(_dt);
        }
        //dataGridView1.DataSource = dtRNDTests;
        //Инициализация таблицы вручную
        DataTable dt = new DataTable("Вопросы");
        dt.Columns.AddRange(new DataColumn[9] {new DataColumn("Вид"),
new DataColumn("Тест"), new DataColumn("Вопрос"), new DataColumn("Ответ А"),
new DataColumn("Ответ Б"), new DataColumn("Ответ В"), new DataColumn("Ответ
Г"),
new DataColumn("Правильный ответ"), new DataColumn("Балл")});
        for (int i = 0; i < dtRNDTests.Rows.Count; i++)
            dt.Rows.Add(dtRNDTests.Rows[i][1], dtRNDTests.Rows[i][2],
dtRNDTests.Rows[i][3],
                dtRNDTests.Rows[i][4], dtRNDTests.Rows[i][5], dtRNDTests.Rows[i][6],
dtRNDTests.Rows[i][7],
                dtRNDTests.Rows[i][8], dtRNDTests.Rows[i][9]);
        String Query = "";
        GetTabele.Get("INSERT INTO Тесты (Тема, Название, Время) VALUES (" +
GetTabele.Get("SELECT Код FROM Тема WHERE Тема='-").Rows[0][0] + ", 'RND', " +
tb_Time.Text + ")");
        for (int i = 0; i < dt.Rows.Count; i++)
        {
            GetTabele.Get("INSERT INTO Вопросы (Вид, Тест, Вопрос, [Ответ А], [Ответ
Б], [Ответ В], [Ответ Г], [Правильный ответ], Балл, Ст) VALUES (" + "" + dt.Rows[i][0] +
", " + GetTabele.Get("(SELECT Код FROM Тесты WHERE Название='RND') ").Rows[0][0] +
", " + dt.Rows[i][2] + ", " + dt.Rows[i][3] + ", " +
                + dt.Rows[i][4] + ", " + dt.Rows[i][5] + ", " + dt.Rows[i][6] + ", " +
dt.Rows[i][7] + ", " + dt.Rows[i][8] + ", true)");
        }
        // Query = Query.TrimEnd(',');
        // GetTabele.Get("INSERT INTO Вопросы (Вид, Тест, Вопрос, [Ответ А], [Ответ
Б], [Ответ В], [Ответ Г], [Правильный ответ], Балл) VALUES " + Query + ";");
        SetActiveTest();
        GetActiveTest();
        MessageBox.Show("Тест успешно сгенерирован");
        Close();
    }
    // catch { }
}

```