Министерство образования и науки Российской Федерации

ФГАОУ ВПО «Уральский федеральный университет имени первого президента России Б.Н.Ельцина»

ИРИТ - РТФ

Кафедра информационных технологий

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. Кафедрой Л.Г. Доросинский\_\_\_\_

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

«\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2013 г.

**Создание системы компьютерного тестирования для оценки профессиональных компетенций специалистов медицинских учреждений**

Пояснительная записка

\_\_\_\_\_\_23010000000010\_\_\_\_\_\_

Руководитель Доцент, канд.пед.наук Папуловская Н.В.

Нормоконтролер Ст.преподаватель Спиричева Н.Р.

Студент гр. Р-490201 Дмитриев И.Л.

Екатеринбург 2013 г.

# Реферат

В состав ВКР входят:

* пояснительная записка 66 с., 38 рисунков, 4 таблицы, 6 источников, 1 приложение;
* графические (демонстрационные) материалы 20 листов.

**ТЕСТИРОВАНИЕ, КОМПЕТЕНЦИИ, СЕТЕВОЕ ВЗАИМОДЕЙСТВИЕ, ПЕРЕДАЧА ДАННЫХ, БАЗА ДАННЫХ ВОПРОСОВ, РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ**

Целью работы является разработка системы тестирования для автоматизации процесса контроля результатов обучения в медицинских учреждения и создания удобного инструмента для выявления профессиональных компетенций специалистов.

Проведено исследование систем аналогов, в результате которого принято решение разрабатывать новую систему тестирования.

Для разработки системы использованы MS Visual Studio 2012 и язык программирования C#.

В результате разработки системы были созданы следующие программные продукты:

1. «Medtest DB Editor» - приложение для автоматизации работы руководителя тестирования с базой данных: регистрирования участников, создания разделов тестирования, вопросов, создания паролей для просмотра результатов каждого участника по отделениям, регистрации и учета количества попыток каждого участника в тестировании, для просмотра результатов и составления диаграмм по ним
2. «Server for Medtest» - программа, предназначенная для передачи информации клиентам по сети
3. «Medtest» - приложение, предназначенное предназначена для проведения компьютерных тестов в лечебно-профилактических учреждениях. Количество вопросов теста - 20. Данная программа позволяет выявлять результат обучения медсестер и уровень их знаний, определять компетенции сотрудников

Содержание

[Реферат 2](#_Toc359279108)

[Нормативные ссылки 4](#_Toc359279109)

[Определения, обозначения и сокращения 5](#_Toc359279110)

[Введение 6](#_Toc359279111)

[Основная часть 8](#_Toc359279112)

[1 Постановка задачи 8](#_Toc359279113)

[2 Анализ существующих аналогов 10](#_Toc359279114)

[3 Требования к разрабатываемой программе 14](#_Toc359279115)

[4 Структура базы данных 16](#_Toc359279116)

[5 Реализация серверной части системы 21](#_Toc359279117)

[6 Реализация клиентской части системы 25](#_Toc359279118)

[7 Руководство пользователя 27](#_Toc359279119)

[7.1 Описание программы «Server for Medtest» 27](#_Toc359279120)

[7.2 Описание программы «Medtest DB Editor» 32](#_Toc359279121)

[7.2.1 Дополнительные функции программы 33](#_Toc359279122)

[7.2.2 Участники 35](#_Toc359279123)

[7.2.3 Разделы 36](#_Toc359279124)

[7.2.4 Вопросы 37](#_Toc359279125)

[7.2.5 Пароли 39](#_Toc359279126)

[7.2.6 Попытки 40](#_Toc359279127)

[7.2.7 Результаты 41](#_Toc359279128)

[7.3 Описание программы «Medtest» 42](#_Toc359279129)

[8 Результаты работы системы 49](#_Toc359279130)

[Заключение 52](#_Toc359279131)

[Список использованных источников 53](#_Toc359279132)

[Приложение A 54](#_Toc359279133)

# Нормативные ссылки

В пояснительной записке использованы ссылки на следующие стандарты:

|  |  |
| --- | --- |
| **ГОСТ 7.1-2003** | Система стандартов по информации, библиотечному и издательскому делу. |
| **ГОСТ 7.32-2001** | Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления. |
| **ФГОС 060500** | Федеральный государственный образовательный стандарт по направлению подготовки «Сестринское дело». |
| **ФГОС 060101** | Федеральный государственный образовательный стандарт по направлению подготовки «Лечебное дело». |

# Определения, обозначения и сокращения

В пояснительной записке используются следующие определения, обозначения и сокращения:

|  |  |
| --- | --- |
| БД | База данных |
| ЛПУ | Лечебно-профилактическое учреждение |
| ОС | Операционная система |
| ПО | Программное обеспечение |
| СУБД | Система управления базами данных |
| IDEF0 | Методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов |
| IP | Internet Protocol - межсетевой протокол. Относится к маршрутизируемым протоколам сетевого уровня семейства TCP/IP |
| MS | Microsoft |

# Введение

В условиях модернизации в здравоохранении проблема диагностики результатов учебной деятельности весьма актуальна с точки зрения совершенствования учебно-воспитательного процесса. Решение этой проблемы требует внедрения новых эффективных методов управления педагогическим процессом. Контроль знаний необходим при любой системе обучения и любой организации учебного процесса.

Компьютерное тестирование - один из эффективных инструментов контроля уровня знаний, полученных в процессе обучения или работы. Многие предприятия специально проводят компьютерное тестирование, для того чтобы выявить наиболее слабые стороны у сотрудников, выяснить, какое направление обучения необходимо данному специалисту для достижения предприятием каких-либо целей. Также компьютерное тестирование позволяет избежать рутинной проверки работ, существенно облегчает работу и сокращает время.

В последнее время все более актуальны, становятся вопросы по изучению сущности новых информационных технологий в контексте возможности их применения как контроля качества знаний обучающихся.

Современное положение в сфере тестового контроля осложняется тем, что у большинства нет специальной подготовки по методике разработки и применения компьютерных тестов именно медицинского направления и выявление определенных компетенций специалистов. Также не получило достаточного распространения использование компьютерных технологий (тестирование) для контроля качества знаний обучающихся в медицине непосредственно на рабочем месте в лечебно-профилактическом учреждении.

Практическая значимость и недостаточная разработанность современной теории контроля знаний на основе применения компьютерных тестовых технологий определили **актуальную проблему**: «Сложность выявления недостатков в обучении и повышении качества знаний сотрудников с помощью традиционных методов контроля знаний». Компьютерные технологии как инновационный инструмент оценивания профессиональной компетенции сотрудника приобретают в связи с этим особое значение для повышения качества знаний обучаемых.

Эти тенденции и повлекли возникновение темы:

**«Создание системы компьютерного тестирования для оценки профессиональных компетенций специалистов медицинских учреждений»**

Цель: Автоматизировать процесс контроля результатов обучения в медицинских учреждения и создать удобный инструмент для выявления профессиональных компетенций специалистов.

# Основная часть

# Постановка задачи

Разработать информационную систему, состоящую из двух частей: серверной и клиентской. Серверная часть должна представлять собой программы для автоматизации работы руководителя тестирования с базой данных и передачи данных по сети. Клиентская часть должна представлять собой программу тестирования пользователей (участников тестирования), выдачи результатов теста и списка компетенций для дальнейшей передачи всей информации серверу и записи в БД. Система должна обеспечивать: регистрацию участников, создание разделов тестирования, заполнения списка компетенций, вопросов, создание паролей для просмотра результатов каждого участника по отделениям, регистрацию и учет количества попыток каждого участника в тестировании, тестирование пользователей, просмотр результатов и составление диаграмм по ним.

В информационной системе должен быть реализован следующий функционал:

* работа с базой тестирования: наполнение, хранение, редактирование, удаление;
* построение диаграмм по результатам тестирования;
* сохранение таблиц в виде Excel, txt;
* сохранение всей базы в виде xml;
* создание запросов к базе тестирования;
* передача сообщений серверу и клиенту по сети: авторизация пользователя для тестирования, выбор раздела, тестирование по выбранному разделу, передача результатов тестирования и списка полученных компетенций;

Требования к процессу разработки:

* составить описание бизнес-процесса в соответствии со стандартом IDEF0;
* составить описание структуры базы данных;
* взаимодействие с пользователем должно быть организовано с помощью графического пользовательского интерфейса (оконное приложение windows);

# Анализ существующих аналогов

Разрабатываемый программный продукт является системой, состоящей из базы данных и нескольких оконных приложений. В качестве СУБД используется Microsoft SQL Server. Оконное приложение для администратора предназначено для автоматизации работы руководителя тестирования с базой данных: регистрирования участников, создания разделов тестирования, вопросов, создания паролей для просмотра результатов каждого участника по отделениям, регистрации и учета количества попыток каждого участника в тестировании, для просмотра результатов и составления диаграмм по ним. Оконное приложения для тестирования предназначено для участника и предоставляет возможность выбирать тест для прохождения и отвечать на него, по окончании тестирования пользователю выводится результат тестирования.

Существует достаточно большое количество аналогов разрабатываемой системы. Например, одним из них является программный продукт «Тест-Экзаменатор 2 в 1 v 3.1». На рисунке 1 отображен интерфейс данной программы.

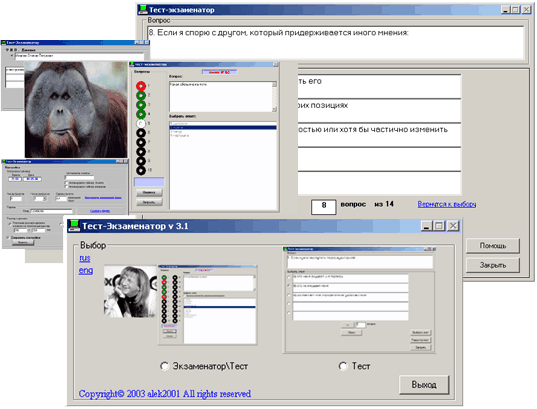


Рисунок 1 - Интерфейс программы «Тест-Экзаменатор 2 в 1 v 3.1»

Бесплатно распространяемая программа. Автор – Алексей ([alek2001@mail.ru](mailto:alek2001@mail.ru))

Основные недостатки: неудобный интерфейс, сложная система добавления вопросов, отсутствие работы по сети, отсутствие определения компетенций.

Другим аналогом можно рассмотреть программу MyTestX – это система программ (программа тестирования учащихся, редактор тестов и журнал результатов) для создания и проведения компьютерного тестирования, сбора и анализа результатов, выставления оценки по указанной в тесте шкале. На рисунке 2 отображен интерфейс данной программы

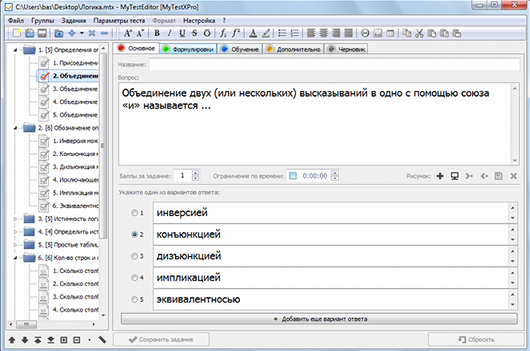


Рисунок 2 - Интерфейс программы «MyTestX»

Программа удобна в использовании. MyTestX работает со многими типами заданий: одиночный выбор, множественный выбор, установление соответствия, ручной ввод текста. В тесте можно использовать любое количество любых типов, можно только один, можно и все сразу. В заданиях с выбором ответа (одиночный, множественный выбор, указание порядка, указание истинности) можно использовать до 10 (включительно) вариантов ответа.

Основные недостатки: платная, сложный интерфейс, отсутствие определения компетенций.

Еще одним аналогом можно рассмотреть тестовую систему Айрен - это бесплатная программа, позволяющая создавать тесты для проверки знаний и проводить тестирование в локальной сети, или на одиночных компьютерах. На рисунке 3 отображен интерфейс данной программы

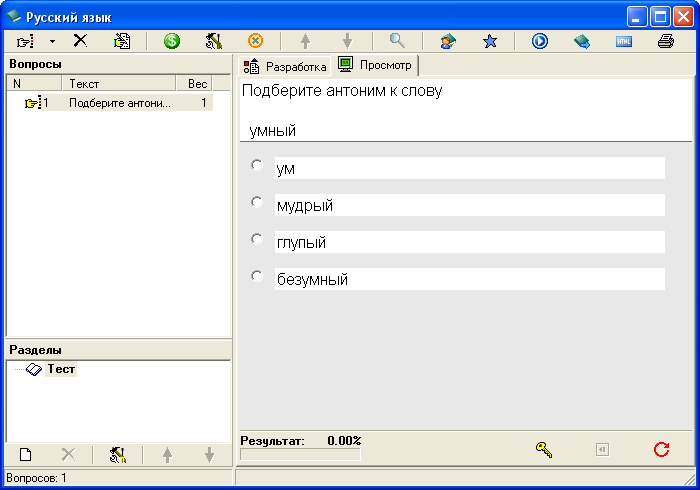


Рисунок 3 - Интерфейс программы «Айрен»

Основные недостатки: сложный и непонятный интерфейс, отсутствие определения компетенций, устаревшая версия (не поддерживает новые ОС).

Более полный анализ аналогов можно увидеть в таблице 1.

Таблица 1 – Анализ существующих аналогов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Критерии оценивания** | **Тест-Экзаменатор** | **MyTestX** | **Айрен** | **Разрабатываемая система** |
| Удобный интерфейс | - | + | - | + |
| Работа по сети | - | + | + | + |
| Определение компетенций | - | - | - | + |
| Формат базы данных | Текстовый | XML | Текстовый | SQL |
| Формат сохранения результатов | .txt | .txt, .xml | Неизвестен | .xlsx, .docx, .xml, .txt |
| Стоимость | Бесплатная | Платная | Бесплатная | Бесплатная |
| Поддержка ОС | Windows XP, 7 | Windows all | Windows XP | Windows XP, 7, 8 |

# Требования к разрабатываемой программе

Исходя из анализа существующих аналогов, для разрабатываемой программы были выдвинуты следующие требования:

* удобный интерфейс;
* простота освоения;
* работа по сети;
* функциональные требования:
  + регистрация и учет пользователей;
  + обработка результатов тестирования;
  + регистрация разделов теста и вопросов к нему;
  + формирование списка используемых попыток решения теста.
* Сохранение результатов тестирования в различных форматах

Для документирования требований к функциональности системы и ее базе данных использовались соглашения стандарта описания бизнес процессов – IDEF0. Данный стандарт предусматривает для любого блока задание не только входных и выходных ресурсов, но и условий выполнения блока, задаваемых на стрелках, входящих сверху, а также требуемых для исполнения функции ресурсов, задаваемых на стрелках, входящих в блок снизу.

На рисунке 4 представлена схема процесса прохождения тестирования. Получение заявки инициирует первую функцию процесса – «Авторизация пользователя». Результатом выполнения функции являются данные о пользователе. Следующая функция «Прохождение тестирования» начинается под управлением номера проходящего теста. Данная функция начинается, когда приходят данные участника. После выполнения прохождения начинается «Обработка результатов тестирования» результатом которой является формирование итогового результата и результата по каждой попытке.

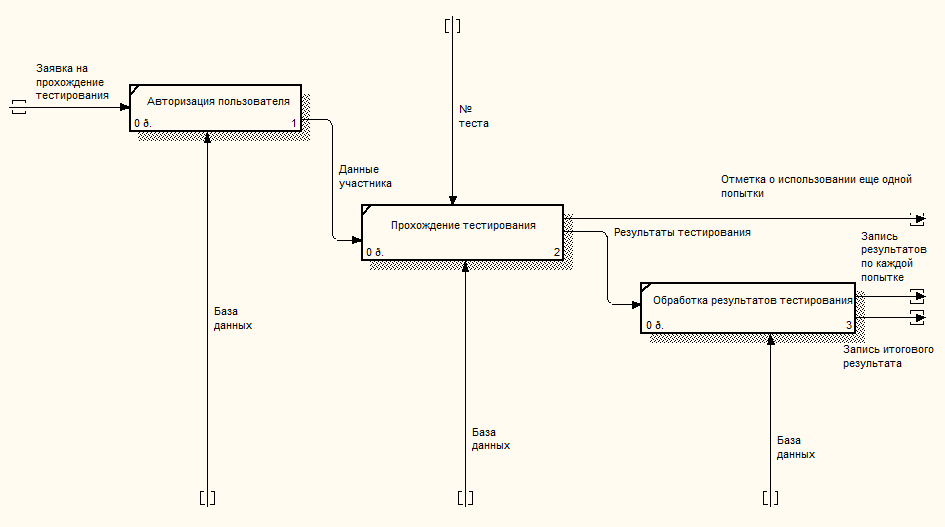


Рисунок 4 - Схема процесса прохождения тестирования

Разрабатываемая программа должна удовлетворять следующим требованиям:

1. Программа должна решать поставленную задачу.
2. Программа должна обеспечить понятный пользователю интерфейс.
3. Программа должна быть разработана на языке программирования C#.
4. В качестве СУБД должен быть использован MS SQL Server 2012.

# Структура базы данных

В результате проведенного анализа были выявлены сущности и их свойства (атрибуты), которые необходимо хранить в базе данных:

1. «Participants» с полями:
   * ID
   * Фамилия
   * Имя
   * Отчество
   * Дата рождения
   * Категория
   * Отделение
   * Пароль
2. «Sections»
   * ID
   * Название
   * Время
   * Количество попыток
   * Запустить тестирование
   * Компетенция 1
   * Вопросов 1
   * Компетенция 2
   * Вопросов 2
   * Компетенция 3
   * Вопросов 3
   * Компетенция 4
   * Вопросов 4
   * Компетенция 5
   * Вопросов 5
   * Компетенция 6
   * Вопросов 6
   * Компетенция 7
   * Вопросов 7
   * Компетенция 8
   * Вопросов 8
   * Компетенция 9
   * Вопросов 9
   * Компетенция 10
   * Вопросов 10
3. «DictCateg»
   * ID
   * Категория
4. «Passwords»
   * ID
   * Название отделения
   * Пароль
5. «Competence»
   * ID
   * Компетенция
6. «Questions»
   * ID
   * Компетенция
   * Вопрос
   * Приложение к вопросу
   * Убран из теста
   * Ответ 1
   * Цель 1
   * Ответ 2
   * Цель 2
   * Ответ 3
   * Цель 3
   * Ответ 4
   * Цель 4
   * Ответ 5
   * Цель 5
   * Цель текст
7. «Attempts»
   * ID
   * Фамилия участника
   * Название раздела
   * Количество попыток
8. «Answers»
   * ID
   * Фамилия участника
   * Название раздела
   * Вопрос
   * Результат 1
   * Результат 2
   * Результат 3
   * Результат 4
   * Результат 5
   * Результат текст
   * Ответ текст
   * Итог
9. «Results»
   * ID
   * Фамилия участника
   * Название раздела
   * Номер попытки
   * Итог
   * Процент 1
   * Процент 2
   * Процент 3
   * Процент 4
   * Процент 5
   * Процент 6
   * Процент 7
   * Процент 8
   * Процент 9
   * Процент 10

Логическая структура разрабатываемой базы данных представлена на рисунке 5. Каждая сущность была задана схемой отдельного отношения (нормализованной таблицы). При этом атрибуты сущностей образуют столбцы таблицы, а экземпляры сущности представлены кортежами отношения. Первичный ключ сущности образует первичный ключ таблицы. В результате этапа проектирования была построена БД, структура которой показана на рисунке 6 (скрипт создания БД приведен в Приложении A).

Настройки подключения к БД:

* Имя БД: Medtest\_db
* Логин: sa
* Пароль: 12345

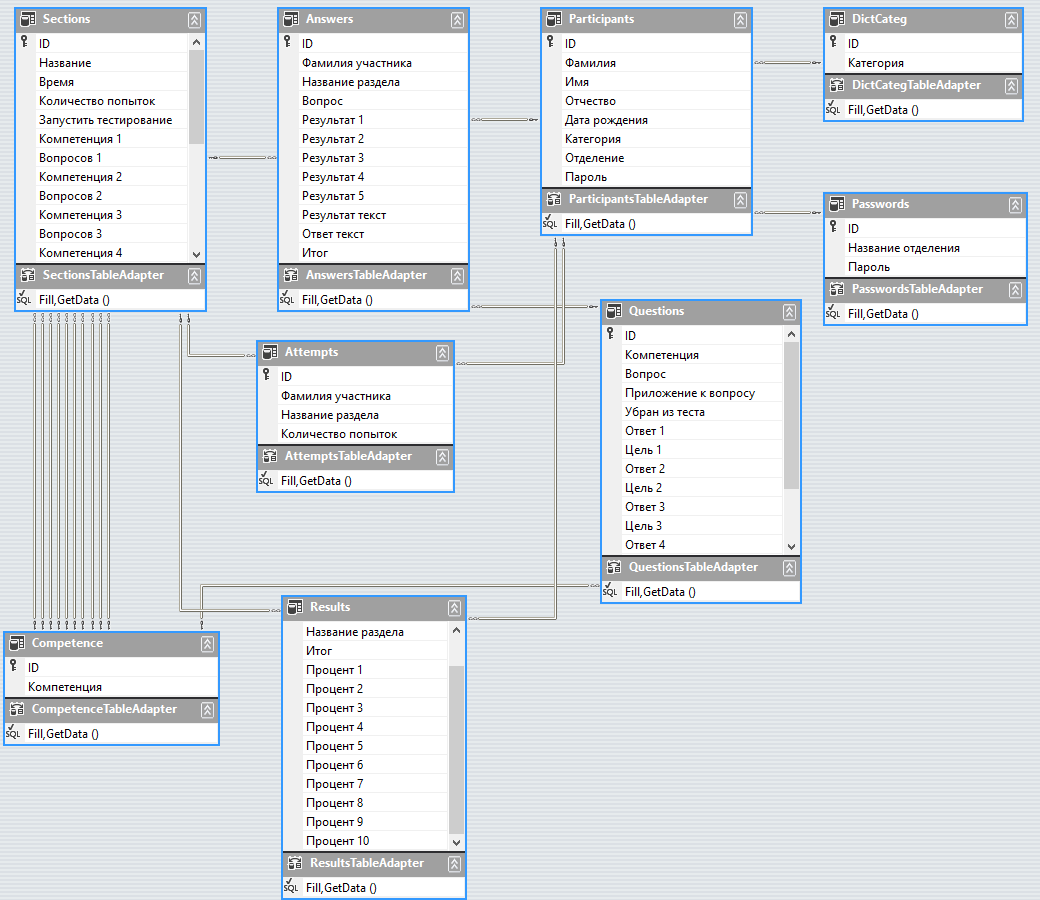


Рисунок 5 - Структура БД сделанная в Visual Studio



Рисунок 6 - Структура БД

# Реализация серверной части системы

При разработке программы основная сложность была в продумывании и грамотной реализации сетевого взаимодействия клиента и сервера. В итоге разработки сетевого взаимодействия было определенно 6 типов сообщений которыми могут общаться сервер и клиент. Данные сообщения представлены в таблице 2.

Таблица 2 – Сетевые сообщения

|  |  |
| --- | --- |
| **Тип сообщения** | **Значение** |
| 0 | Проверка работоспособности сервера |
| 1 | Авторизация пользователя |
| 2 | Запрос списка тестов |
| 3 | Запрос информации о выбранном тесте |
| 4 | Запрос вопросов по выбранному тесту и проверка на количество использованных попыток |
| 5 | Передача результатов |

Рассмотрим основные фрагменты программы «Server for Medtest»:

1. Передача сообщения

struct SendArgument

{

/// <summary>

/// IP адрес сервера

/// </summary>

public string Hostname;

/// <summary>

/// Сообщение для передачи

/// </summary>

public string Message;

}

private void sendBackgroundWorker\_DoWork(object sender, DoWorkEventArgs e)

{

SendArgument argumen = (SendArgument)e.Argument;

TcpClient client = new TcpClient(argumen.Hostname, 10000);

using (NetworkStream outputStream = client.GetStream())

using (BinaryWriter writer = new BinaryWriter(outputStream))

{

writer.Write(argumen.Message);

}

client.Close();

}

Для передачи сообщения используется три класса:

* TcpClient – предоставляет клиентские подключения для сетевых служб протокола TCP;
* NetworkStream – обеспечивает базовый поток данных для доступа к сети;
* BinaryWriter – записывает простые типы данных в поток как двоичные значения и поддерживает запись строк в определенной кодировке.

1. Прием сообщения

/// <summary>

/// Тип сообщения который передает клиент:

/// 0 - Проверка что сервер работет,

/// 1 - Авторизация пользователя,

/// 2 - Запрос списка тестов,

/// 3 - Запрос выбранного теста,

/// 4 - Запрос вопросов по выбранному тесту,

/// 5 - Передача результатов.

/// </summary>

private int message\_type;

private void receiveBackgroundWorker\_DoWork(object sender, DoWorkEventArgs e)

{

background\_number = 1;

TcpListener listener = new TcpListener(IPAddress.Any, 20000);

listener.Start(30);

while (!listener.Pending())

{

TcpClient client = listener.AcceptTcpClient();

using (NetworkStream inputStream = client.GetStream())

using (BinaryReader reader = new BinaryReader(inputStream))

{

string[] enter = reader.ReadString().Split('\n');

//IP-адрес

string hostname = enter[0];

message\_type = Convert.ToInt32(enter[1]);

if (message\_type == 0)

{

sendBackgroundWorker.RunWorkerAsync(

new SendArgument()

{

Hostname = hostname,

Message = "Server"

});

}

}

client.Close();

}

listener.Stop();

}

Для приема сообщения используется четыре класса:

* TcpListener – прослушивает подключения от TCP-клиентов сети;
* TcpClient – предоставляет клиентские подключения для сетевых служб протокола TCP;
* NetworkStream – обеспечивает базовый поток данных для доступа к сети;
* BinaryReader – считывает простые типы данных как двоичные значения в заданной кодировке

1. Отказоустойчивость

Для меньшей загрузки сервера можно запустить несколько серверов в различных местах. Также для избегания ошибок при неполной передаче сообщения или при прочих ошибках в сети в программе существует счетчик для переключения, который отображает какой именно участок программы вызвал ошибку и перезапускает неисправный участок.

/// <summary>

/// Показывает какой receiveBackgroundWorker сейчас работает

/// </summary>

private int background\_number = 0;

Серверная часть состоит из 2 программ:

1. «Server for Medtest» - программа сервер, необходима для получения и сохранения результатов теста и для отправки сообщений различных типов по запросу клиента. Данная программа сохраняет результаты каждого участника в общий список с результатами (файл Excel). И записывает каждый ответ участника в файл MS Word, который всегда можно распечатать.
2. «Medtest DB Editor» - предназначена для автоматизации работы пользователя с БД и дает возможность: регистрировать участников, создавать разделы тестирования, заполнять список компетенций, вопросов, вести учет количества попыток каждого участника в тестировании, просмотр результатов тестирования и составление диаграмм по ним.

# Реализация клиентской части системы

При разработке клиентской части программы необходимо было составить или выбрать алгоритм для подсчета результатов тестирования и выявления компетенций. В ходе разработки было принято решение выводить количество верных заданий по компетенциям в процентном соотношении и на основании этого выявлять уровень владения компетенцией в целом. Для решения этой задачи было выбрано две различные системы оценки знаний:

1. Система оценки знаний США

Данная система была использована для подсчета и определения итоговой суммарной оценки тестирования без учета различных компетенций. Она представлена в таблице 3.

Таблица 3 – Система оценки знаний США

|  |  |
| --- | --- |
| **Оценка** | **Процент** |
| Отлично | 90% - 100% |
| Хорошо | 80% - 89% |
| Средний уровень | 70% - 79% |
| Удовлетворительно | 65% - 69% |
| Неудовлетворительно | 64% и ниже |

1. Комбинированная система оценки

Данная система была составлена на основании систем оценивания знаний различных стран: США, Австрия, Албания, Болгария. Она позволила определить уровень владения компетенциями на основании количества верных заданий по ним. Система представлена в таблице 4.

Таблица 4 – Комбинированная система оценки

|  |  |
| --- | --- |
| **Уровень владения компетенцией** | **Процент** |
| Высокий | 80% - 100% |
| Средний | 50% - 79% |
| Низкий | 26% - 49% |
| Не владеет | 25% и ниже |

# Руководство пользователя

## Описание программы «Server for Medtest»

Рассмотрим программу «Server for Medtest» более подробно. После запуска программы открывается окно соединения с базой данных, которое показано на рисунке 7. IP адрес сервера отображается в заголовке формы. Пользователь может ввести в данном окне различные параметры, которые стандартно используются для соединения с базой данных: имя сервера, имя пользователя, пароль, имя БД. Для подключения к базе данных с выбранными параметрами пользователь должен нажать кнопку «Соединить». Если пользователь не готов соединиться в данный момент с базой он может нажать кнопку «Отмена» и выйти из программы.

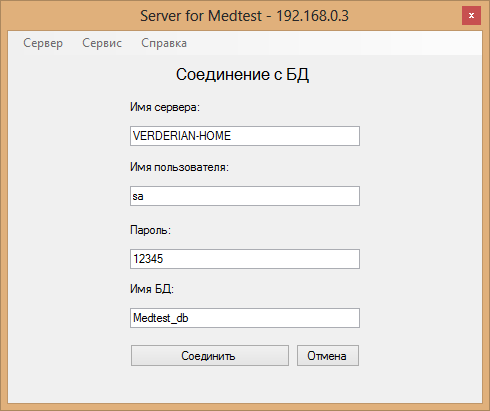


Рисунок 7 – Окно программы «Server for Medtest»: Соединение с БД

После соединения с базой открывется основное окно программы, которое показано на рисунке 8. В данном окне есть строка состояния процесса принятия сообщения от клиента, строка состояния работоспособности сервера, лог сообщений сервера (внизу), меню состоящее из трех пунктов: «Сервер», «Сервис», «Справка».

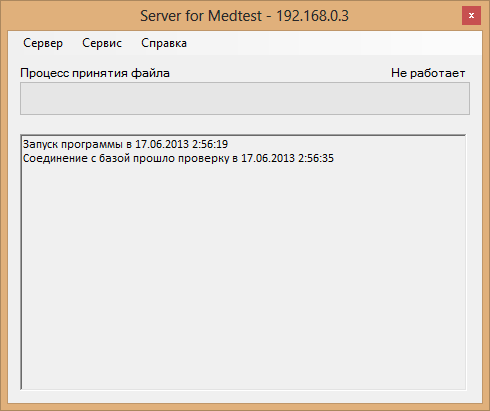


Рисунок 8 – Окно программы «Server for Medtest»: Основное окно

В основном окне программы пользователь может запустить сервер для этого надо зайти в пункт меню «Сервер» и выбрать подпункт «Запустить». Для выхода из сервера нужно нажать кнопку «Выход». Процесс запуска сервера можно увидеть на рисунке 9.

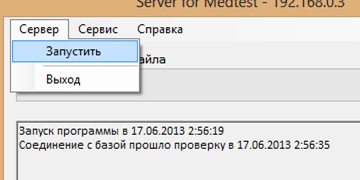


Рисунок 9 – Окно программы «Server for Medtest»: Процесс запуска сервера

После запуска сервера можно заметить как в основном окне изменился статус работоспособности сервера и лог сервера (внизу) рисунок 10. Стоит заметить что при любой ошибке сервера или его остановке соотвествующий статус отображается в строке статуса работоспособности сервера.

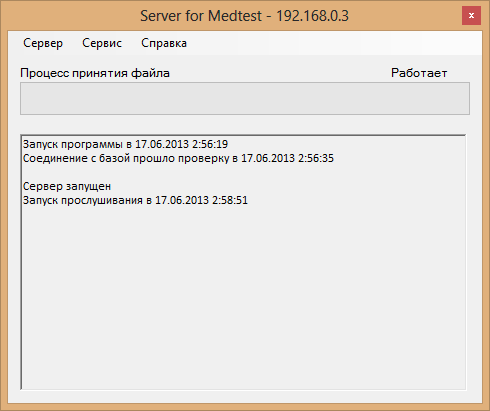


Рисунок 10 – Окно программы «Server for Medtest»: Основное окно после запуска сервера

Типичный обмен сообщениями клиента и сервера можно увидеть на рисунке 11.

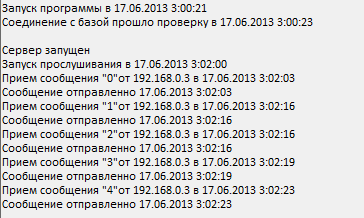


Рисунок 11 – Окно программы «Server for Medtest»: Типичный лог сервера

В программе «Server for Medtest» предусмотрена гибкая настройка параметров сервера предназначенная для того чтобы при изменении параметров подключения к базе или места сохранения результатов тестирования не приходилось постоянно при каждом отдельном запуске программы менять настройки. Для отображения настроек необходимо зайти в пункт меню «Сервис» и выбрать подпункт «Настройки». Данный процесс отображен на рисунке 12.

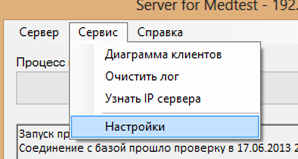


Рисунок 12 – Окно программы «Server for Medtest»: Пункт меню «Сервис»

Также в пункте меню «Сервис» пользователь может очистить лог или узнать IP сервера. Окно «Настройки» можно увидеть на рисунке 13.

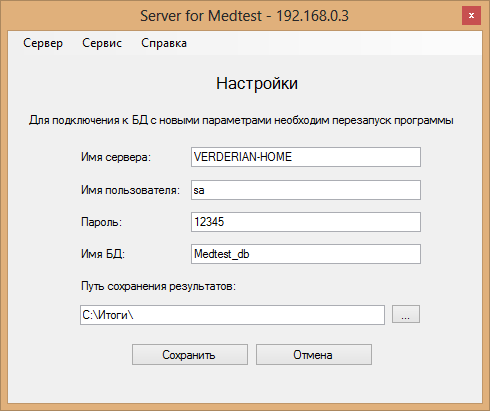


Рисунок 13 – Окно программы «Server for Medtest»: Окно настроек сервера

В окне настроек сервера можно изменить путь сохранения результатов и параметры соединения с БД: имя сервера, имя пользователя, пароль, имя БД. Для сохранения настроек нужно нажать кнопку «Сохранить», для возврата в основное окно без изменений нужно нажать кнопку «Отмена».

Пользователь может запросить информацию о программе при выборе соответствующего подпункта в пункте меню «Справка». Информация о программе «Server for Medtest» отображена на рисунке 14.

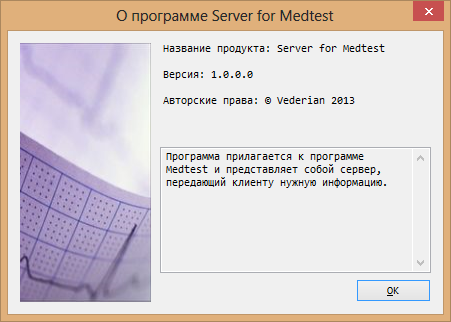


Рисунок 14 – Окно программы «Server for Medtest»: О программе

## Описание программы «Medtest DB Editor»

В результате выполнения работ была разработана программа «Medtest DB Editor». Данное приложение является оконным, оно имеет восемь вкладок, на каждой из которой содержится весь необходимый функционал для работы с определенной сущностью БД.

Перечень всех вкладок:

1. Участники
2. Разделы
3. Вопросы
   1. Просмотреть вопросы
   2. Добавить вопрос
4. Пароли
5. Попытки
6. Результаты
   1. Итоговые результаты
   2. Диаграмма

Также в программе есть дополнительные окна – окно для любого запроса по всей базе данных и окно «О программе». Далее будет представлено подробное описание каждого раздела приложения – каждой вкладки.

### *Дополнительные функции программы*

|  |
| --- |
|  |
| Рисунок 15 - Окно программы: панель «Соединение с сервером» |

При запуске программы появляется панель «Соединение с сервером» (рисунок 15), которое служит для подключения администратора к базе. В данной панели присутствует 4 текстовых поля:

* Имя сервера
* Имя пользователя
* Пароль
* Имя БД

При нажатии кнопки «Соединить» происходит соединение с базой данных по заданным параметрам идентификации, если параметры ввода не верны или заполнены не все поля то пользователь получает текстовое уведомление (рисунок 16).

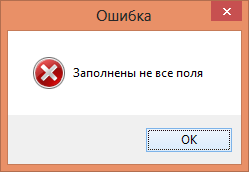


Рисунок 16 - Окно программы: ошибка

После авторизации пользователь может зайти в меню и сохранить всю базу в формате .xml (рисунок 17).

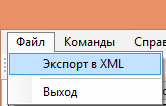
**

Рисунок 17 - Окно программы: Меню «Файл»

### *Участники*

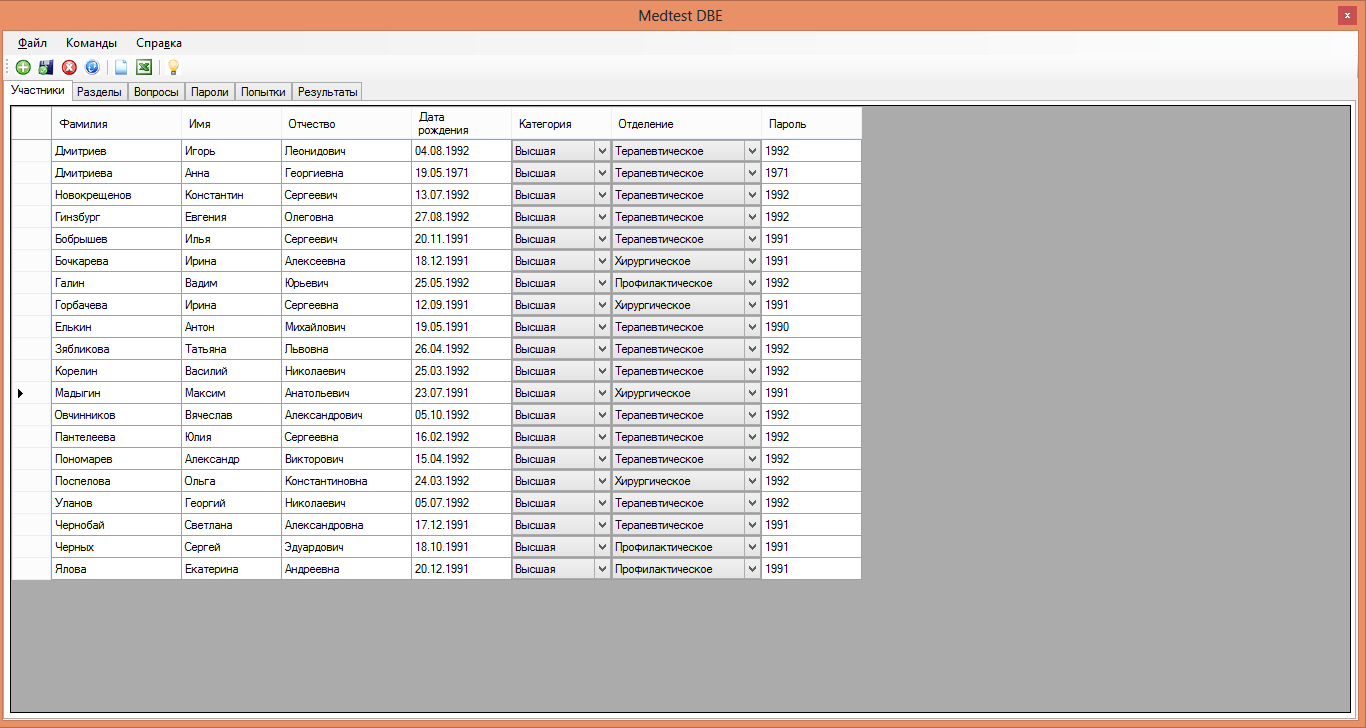


Рисунок 18 - Окно программы: раздел «Участники»

Раздел «Участники» (рисунок 18) содержит

* Список всех участников
* Кнопку добавления участника
* Кнопку сохранения участника
* Кнопку удаления участника
* Кнопку обновления таблицы участников
* Кнопку для сохранения таблицы участников в текстовый документ формата .txt или в Excel

Чтобы добавить нового участника, достаточно нажать кнопку «Добавить».

Чтобы отредактировать информацию о существующем участнике, нужно дважды щелкнуть по ячейке в таблице, а после редактирования нажать кнопку «Сохранить».

Чтобы удалить нужных участников, нужно выделить соответствующие строки в таблице нажать кнопку «Удалить».

Чтобы обновить таблицу участников, нужно нажать кнопку «Обновить».

Чтобы сохранить таблицу участников в формате .txt, нужно нажать кнопку «Сохранить в .txt».

Чтобы сохранить таблицу участников в Excel, нужно нажать кнопку «Сохранить в Excel».

### *Разделы*

|  |
| --- |
| Рисунок 19 - Окно программы: раздел «Разделы»  Раздел «Разделы» (рисунок 19) содержит   * Список названий всех разделов * Кнопку добавления раздела * Кнопку сохранения раздела * Кнопку удаления раздела * Кнопку обновления таблицы разделов * Кнопку для сохранения таблицы разделов в текстовый документ формата .txt или в Excel   Чтобы добавить новый раздел, достаточно нажать кнопку «Добавить».  Чтобы отредактировать информацию о существующем разделе, нужно дважды щелкнуть по ячейке в таблице, а после редактирования нажать кнопку «Сохранить».  Чтобы удалить нужные разделы, нужно выделить соответствующие строки в таблице нажать кнопку «Удалить».  Чтобы обновить таблицу разделов, нужно нажать кнопку «Обновить».  Чтобы сохранить таблицу разделов в формате .txt, нужно нажать кнопку «Сохранить в .txt».  Чтобы сохранить таблицу разделов в Excel, нужно нажать кнопку «Сохранить в Excel». |

### *Вопросы*

|  |
| --- |
|  |
| Рисунок 20 - Окно программы: раздел «Вопросы» вкладка «Посмотреть вопросы»  Вкладка «Посмотреть вопросы» раздела «Вопросы» (рисунок 20) содержит   * Список всех вопросов * Список-меню Разделы * Кнопку сохранения вопроса * Кнопку удаления вопроса * Кнопку обновления таблицы вопросов * Кнопку для сохранения таблицы вопросов в текстовый документ формата .txt или в Excel   Чтобы просмотреть список вопросов по разделу, нужно в списке-меню выбрать соответствующий раздел.  Чтобы отредактировать информацию о существующем вопросе в разделе, нужно дважды щелкнуть по ячейке в таблице, а после редактирования нажать кнопку «Сохранить».  Чтобы удалить нужные вопросы, нужно выделить соответствующие строки в таблице нажать кнопку «Удалить».  Чтобы обновить таблицу вопросов, нужно нажать кнопку «Обновить».  Чтобы сохранить таблицу вопросов в формате .txt, нужно нажать кнопку «Сохранить в .txt».  Чтобы сохранить таблицу вопросов в Excel, нужно нажать кнопку «Сохранить в Excel».  Чтобы добавить новый вопрос, нужно нажать на вкладку «Добавить вопрос».    Рисунок 21 - Окно программы: раздел «Вопросы» вкладка «Добавить вопрос»  Вкладка «Добавить вопрос» раздела «Вопросы» (рисунок 21) содержит   * Список-меню Разделы * Кнопку добавления вопроса * Кнопку очищения заполненных полей   Чтобы добавить вопрос, нужно в списке-меню выбрать соответствующий раздел и заполнить все необходимые поля, а после нажать клавишу «Добавить». |

### *Пароли*

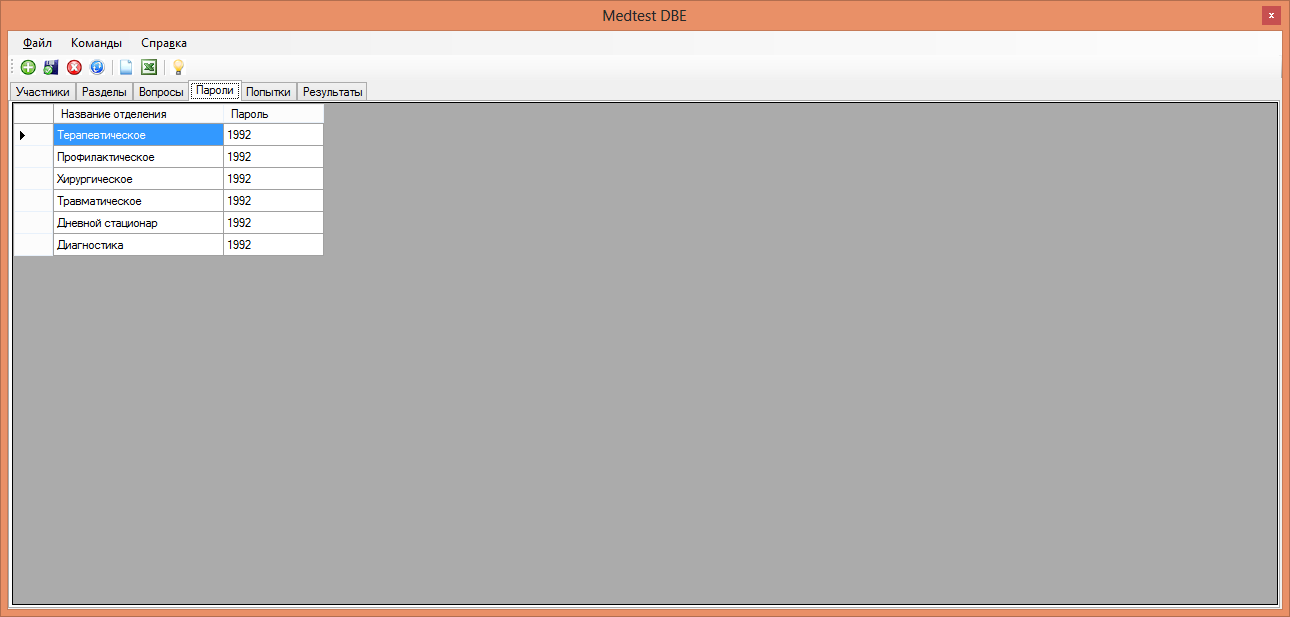


Рисунок 22 - Окно программы: раздел «Пароли»

Раздел «Пароли» (рисунок 22) содержит

* Список названий всех отделов
* Кнопку добавления отдела
* Кнопку сохранения отдела
* Кнопку удаления отдела
* Кнопку обновления таблицы отделов
* Кнопку для сохранения таблицы отделов в текстовый документ формата .txt или в Excel

Чтобы добавить новый отдел, достаточно нажать кнопку «Добавить».

Чтобы отредактировать информацию о существующем отделе, нужно дважды щелкнуть по ячейке в таблице, а после редактирования нажать кнопку «Сохранить».

Чтобы удалить нужные отделы, нужно выделить соответствующие строки в таблице нажать кнопку «Удалить».

Чтобы обновить таблицу отделов, нужно нажать кнопку «Обновить».

Чтобы сохранить таблицу отделов в формате .txt, нужно нажать кнопку «Сохранить в .txt».

Чтобы сохранить таблицу отделов в Excel, нужно нажать кнопку «Сохранить в Excel».

### *Попытки*

|  |
| --- |
|  |
| Рисунок 23 - Окно программы: раздел «Попытки»  Раздел «Попытки» (рисунок 23) содержит   * Список всех попыток * Кнопку добавления попытки * Кнопку сохранения попытки * Кнопку удаления попытки * Кнопку обновления таблицы попыток * Кнопку для сохранения таблицы попыток в текстовый документ формата .txt или в Excel   Чтобы добавить новую попытку, достаточно нажать кнопку «Добавить».  Чтобы отредактировать информацию о существующей попытке, нужно дважды щелкнуть по ячейке в таблице, а после редактирования нажать кнопку «Сохранить».  Чтобы удалить нужную попытку, нужно выделить соответствующие строки в таблице нажать кнопку «Удалить».  Чтобы обновить таблицу попыток, нужно нажать кнопку «Обновить».  Чтобы сохранить таблицу попыток в формате .txt, нужно нажать кнопку «Сохранить в .txt».  Чтобы сохранить таблицу попыток в Excel, нужно нажать кнопку «Сохранить в Excel». |

### *Результаты*

|  |
| --- |
|  |
| Рисунок 24 - Окно программы: раздел «Результаты» |

Раздел «Результаты» (рисунок 24) содержит

* Список всех результатов
* Кнопку удаления результата
* Кнопку обновления таблицы результатов
* Кнопку для сохранения таблицы результатов в текстовый документ формата .txt или в Excel

## Описание программы «Medtest»

Рассмотрим клиентскую часть данного комплекса программ (для сотрудников):

Сотрудник имеет на своем рабочем компьютере программу “Medtest”, которая представляет собой оболочку для тестирования. Пользователь вне зависимости от того есть ли рядом руководитель, может проходить тестирование. Рассмотрим поэтапно, как пользователь работает с данной программой: при запуске программы «Medtest» идет проверка на то что сервер с IP-адресом находящимся в первоначальных настройках программы доступен. Если проверка не пройдена то открывается сообщение выбора, которое отображено на рисунке 25. Если пользователь в данном окне нажмет кнопку «Да», то он откроет окно с настройками IP адреса сервера, для выхода пользователь нажимает «Нет».

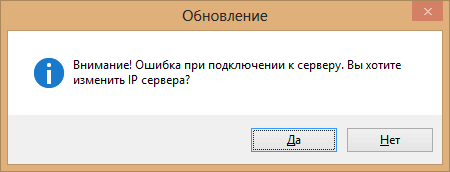


Рисунок 25 – Сообщение выбора IP

В окне настроек, изображенном на рисунке 26, пользователь может изменить IP адрес сервера и проверить работает ли новый адрес перед сохранением. При положительном результате пользователь увидит сообщение изображенное на рисунке 27. После пользователь может сохранить новый IP адрес, который при следующем запуске программы сохранится.

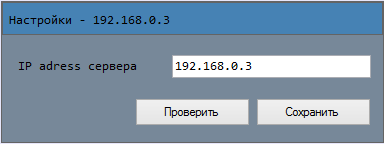


Рисунок 26 – Окно настроек IP

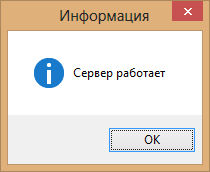


Рисунок 27 – Сообщение о работоспособности сервера

После успешной первоначальной проверки или после изменения настроек пользователь видит начальное окно, которое отображено на рисунке 28. В данном окне пользователь вводит свои персональные данные, а именно, ФИО и табельный номер, который является паролем.

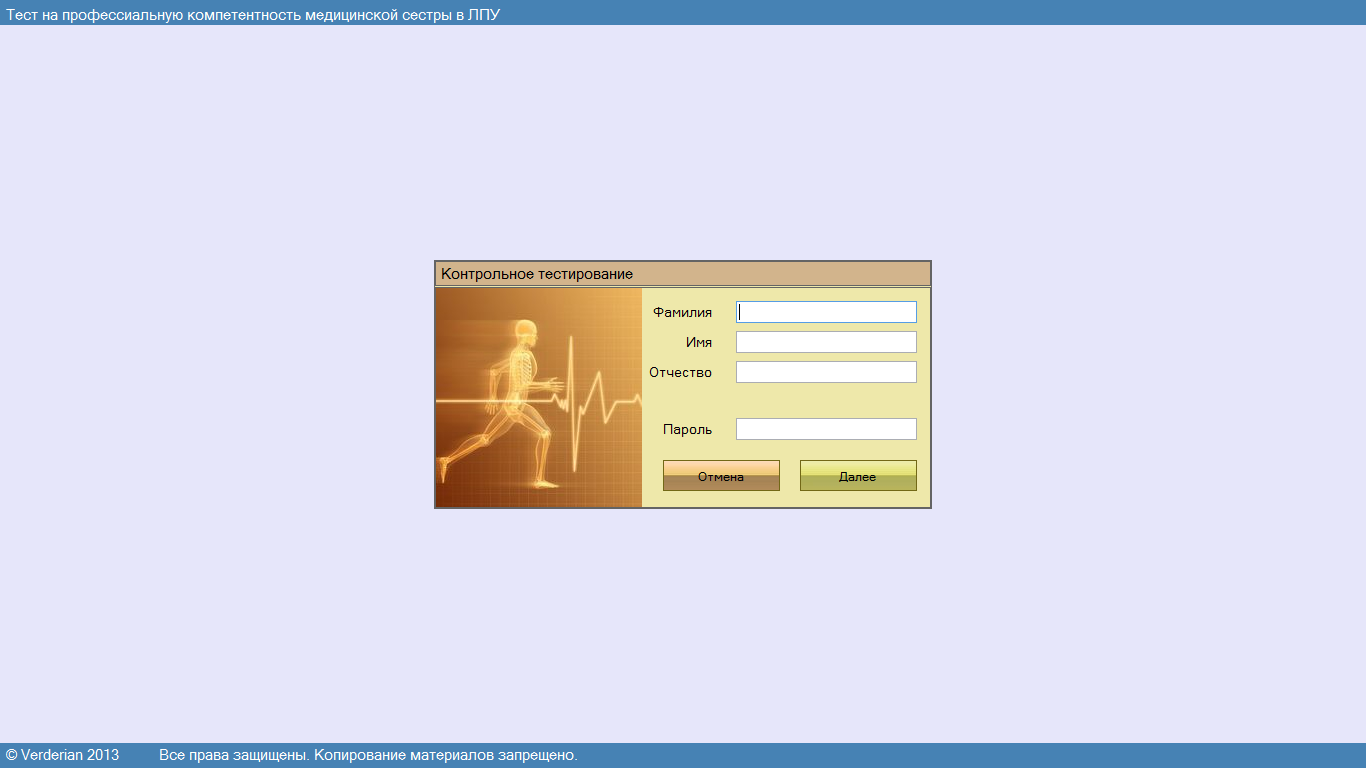


Рисунок 28 – Начальное окно программы (Авторизация)

Пример заполнения персональных данных показан на рисунке 29.

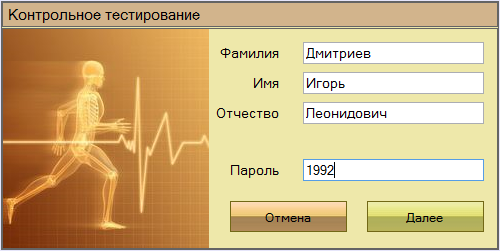


Рисунок 29 – Пример авторизации пользователя

После того как пользователь ввел свои персональные данные, он нажимает клавишу «Далее» и происходит незаметная для пользователя передача сообщения серверу для проверки авторизации пользователя. Если авторизация прошла успешно и сервер дал положительный ответ клиент запрашивает список разделов у сервера и предоставляет возможность клиенту выбрать раздел (рисунок 30), по которому он будет тестироваться (данный раздел ему сообщает экзаменатор).

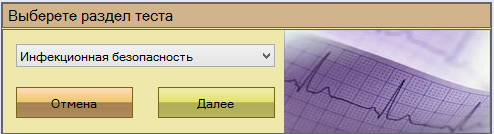


Рисунок 30 – Выбор раздела теста

После выбора раздела клиент запрашивает информацию о выбранном разделе у сервера, а пользователь переходит на следующую страницу, где видит личную информацию о себе и о дальнейшем тестировании (рисунок 31). В данном окне пользователь может также ознакомится с правилами тестирования, нажав клавишу «Помощь» (рисунок 32). Или изучить структуру теста нажав на клавишу «Структура» (рисунок 33). Структура теста содержит список компетенций и количество вопросов на каждую из компетенций в данном тесте.

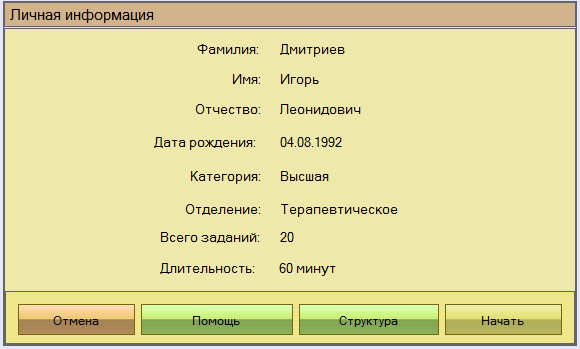


Рисунок 31 – Личная информация об участнике

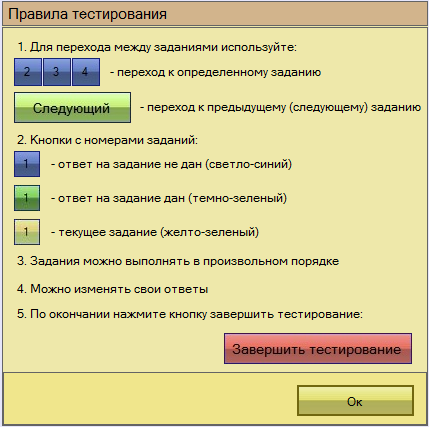


Рисунок 32 – Окно программы «Помощь»

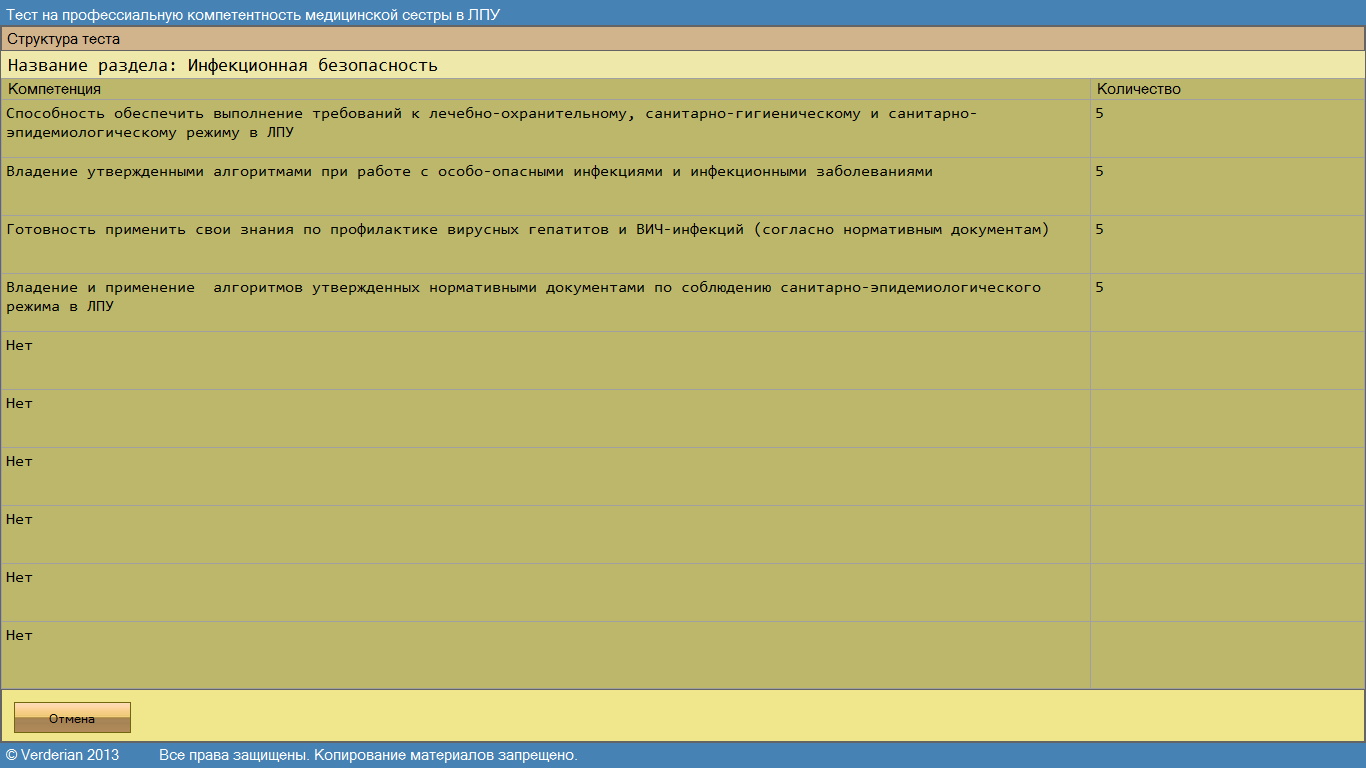


Рисунок 33 – Окно программы «Структура»

Далее пользователь нажимает клавишу «Начать» и тем самым запускает окно тестирования, которое изображено на рисунке 34. Клиент запрашивает список вопросов у сервера по данному разделу и на экране высвечиваются задания, в которых необходимо выбрать правильные варианты ответов. Количество заданий стандартно для всех тестов 20, но эти задания берутся случайным образом из общего количества вопросов в данном разделе (например, из раздела в 100 вопросов берется 20 случайных, но подходящих под определенные компетенции выбранного теста). Пользователь может спокойно переходить с вопроса на вопрос, возвращаться к уже пройденным вопросам, чтобы очередной раз подумать или изменить свою точку зрения, также есть вкладка «Помощь», которая напоминает пользователю о правилах тестирования. Прохождение теста ограничено по времени и по количеству попыток, если пользователь использовал все возможные попытки по данному разделу, но не добился желаемого результата, он может подойти к экзаменатору (руководителю) и договориться о дополнительной попытке или сам экзаменатор предоставит ему эту возможность, видя по локальной сети результат.

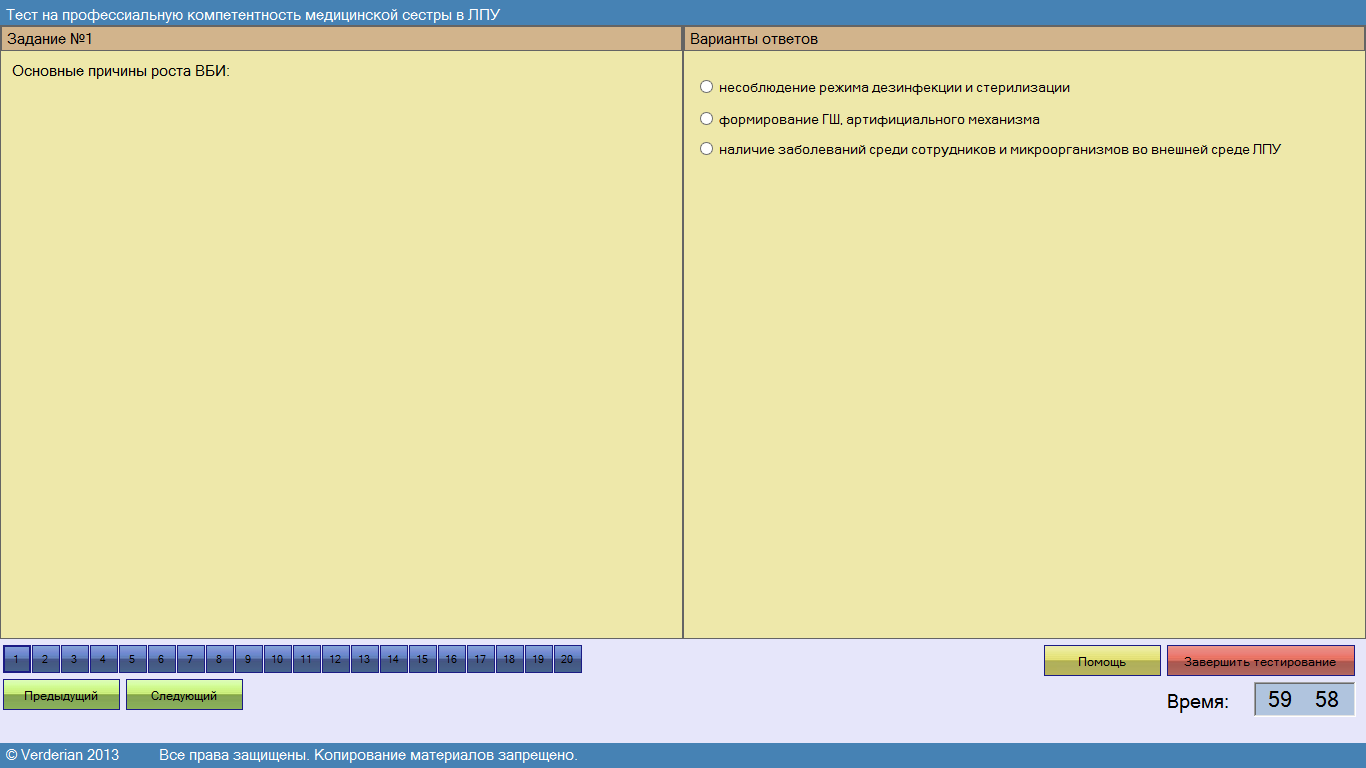


Рисунок 34 – Окно программы «Тест»

По истечении времени тест заканчивается, и пользователь переходит на следующую страницу или если он закончил заранее и нажал клавишу «Завершить тестирование», то происходит процесс передачи данных по локальной сети серверу, а пользователю сразу выдается окно с результатом тестирования, списком полученных компетенций и уровнем их владения (рисунок 35).

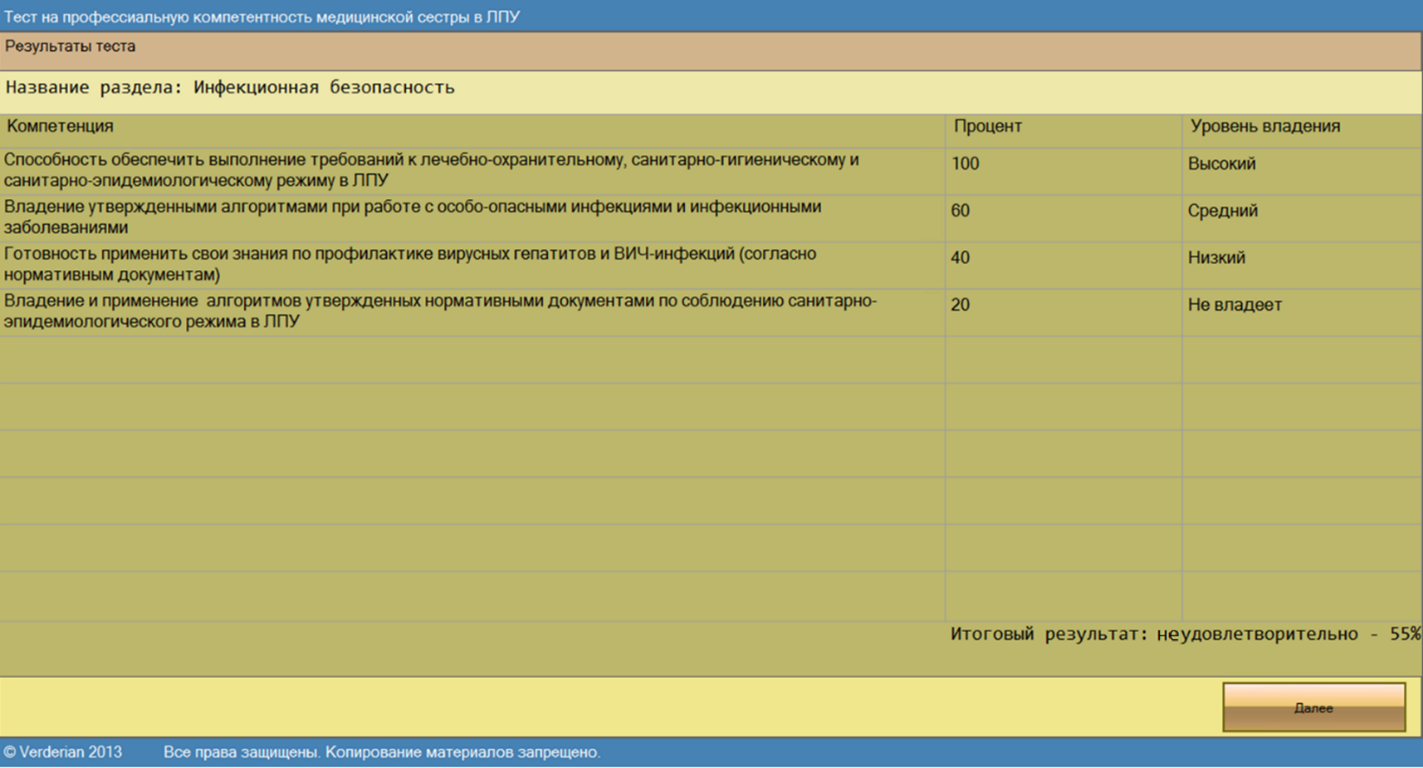


Рисунок 35 – Окно программы «Результаты теста»

После просмотра результата программа готова к следующему тестированию или завершает свою работу (рисунок 36).

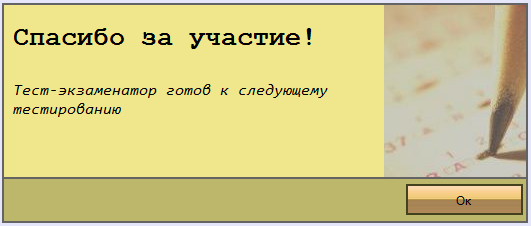


Рисунок 36 – Окно программы «Спасибо»

Также в любой момент пользователь может вывести окно «О программе» с помощью горячей клавиши F1, которое изображено на рисунке 37.

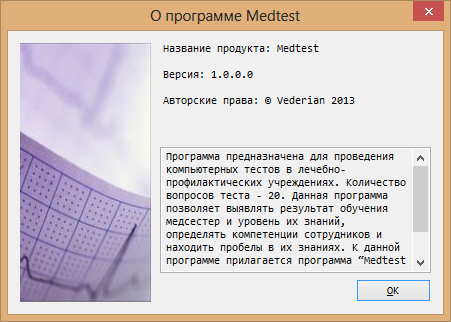


Рисунок 37 – Окно «О программе»

# Результаты работы системы

Разработанная система дает возможность выявления следующих профессиональных компетенций:

1. Способность применять знания общих закономерностей происхождения и развития жизни, строения организма, представления о факторах формирования здоровья, регуляции и саморегуляции в норме и патологии, с целью проведения целенаправленных мероприятий по профилактике заболеваний;
2. Способность оказать качественную медицинскую помощь (с соблюдением правовых, этических норм);
3. Способность вести пропаганду здорового образа жизни на основе научного представления о здоровом образе жизни;
4. Готовность использовать приобретенные профессиональные знания для обучения (консультаций) пациента (семьи) по вопросам профилактики обострений заболеваний, их осложнений, о рациональном питании, физических нагрузках;
5. Готовность использовать в своей работе нормативные документы, определяющие основные задачи, функции и порядок деятельности учреждений здравоохранения;
6. Способность обеспечить выполнение требований к лечебно-охранительному, санитарно-гигиеническому и санитарно-эпидемиологическому режиму в ЛПУ;
7. Владение утвержденными алгоритмами при работе с особо-опасными инфекциями и инфекционными заболеваниями;
8. Готовность применить свои знания по профилактике вирусных гепатитов и ВИЧ-инфекций (согласно нормативным документам);
9. Владение и применение алгоритмов утвержденных нормативными документами по соблюдению санитарно-эпидемиологического режима в ЛПУ;
10. Способность выполнять сестринские манипуляции (согласно стандартам по выполнению манипуляций) - оказывать медицинские услуги, подготавливать пациента к диагностическим процедурам, по назначению врача осуществлять лекарственную терапию;
11. Готовность обеспечить квалифицированный уход за пациентом с учетом его индивидуальных потребностей и проблем, на основе владения методами сбора и оценки данных о состоянии здоровья пациента, методологии сестринского процесса, результатов оценки эффективности оказания медицинской и медико-социальной помощи пациенту;
12. Готовность к взаимодействию медперсонала с пациентом, соблюдая все этические нормы;
13. Готовность оказать медицинскую помощь при неотложных и угрожающих жизни состояниях (зависимые и независимые сестринские вмешательства);
14. Владение и готовность в любой момент использовать знания основ сердечно-легочной реанимации;
15. Умение оценить состояние пациента по клиническим проявлениям и способность оказать по необходимости доврачебную помощь;
16. Способность и готовность проводить противоэпидемические мероприятия, защиту населения в очагах особо опасных инфекций, при ухудшении радиационной обстановки и стихийных бедствиях.

После проведения тестирования администратор тестирования может увидеть результаты с помощью программы «Medtest DB Editor» и построить диаграмму участников (рисунок 38). А пользователь может просмотреть результаты в окне программы «Medtest» - «Результаты теста» (рисунок 35).

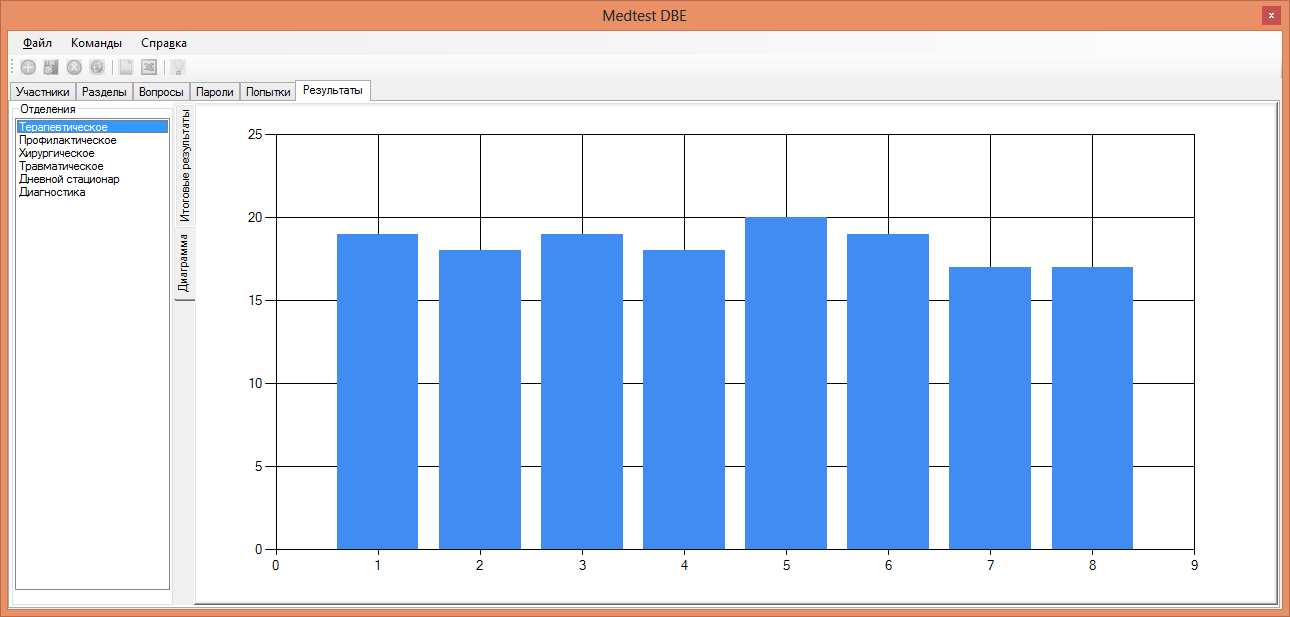


Рисунок 38 - Окно программы «Medtest DB Editor»: раздел «Диаграмма»

# Заключение

Результатом разработки является спроектированная система тестирования включающая базу данных для MS SQL Server 2012, клиентскую программу, две серверные программы (для администратора тестирования). Данная программа не требует установки, но перед ее использованием необходимо создать базу дынных с помощью SQL-скрипта. Программа разработана для .NET Framework 4.0 и не требует дополнительных библиотек и настроек.

Разработанный программный продукт имеет следующие достоинства:

* минимальные системные требования;
* простота использования;
* не требует специальной подготовки пользователя.
* позволяет строить диаграмму результатов и сохранять их в разные форматы

Разработанный программный продукт предоставляет администратору тестирования возможность работать с БД: редактировать, добавлять и удалять данные. Также система предоставляет возможность проводить тестирование сотрудников по локальной сети на их персональных компьютерах в любое удобное время. Она автоматически обрабатывает результаты тестирования и выводит результат и список полученных компетенций как руководителю теста так и сотруднику который прошел тест.

Система полностью выполняет свою поставленную задачу. В плане доработки программы можно сказать что необходимо добавить возможность построения большего количества диаграмм по результатам тестирования, например результат тестирования по нескольким разделам, а также можно добавить возможность вставки изображений в вопросы тестирования.

# Список использованных источников

1. Работа с базами данных под управлением MS SQL Server 2000: Учебное пособие /Ю.П. Парфенов. Екатеринбург: ГОУ ВПО УГТУ-УПИ, 2005. 136с.
2. Базы данных в автоматизированных системах: учебное пособие /Ю.П. Парфенов. Екатеринбург: ГОУ ВПО УГТУ-УПИ, 2006. 108с.
3. Википедия – Свободная энциклопедия. [Электронный ресурс]. – Режим доступа <http://ru.wikipedia.org/wiki>, дата доступа 15.06.2013
4. Библиотека MSDN. [Электронный ресурс]. – Режим доступа <http://msdn.microsoft.com/ru-ru/librar>, дата доступа 14.06.2013
5. Документированная процедура СМК-ДП-8.2А.4-02-2010 «Итоговая государственная аттестация выпускников»
6. ГОСТ 7.32-2001. Отчет о научно-исследовательской работе. Структура и правила оформления.

# Приложение A

USE [master]

GO

/\*\*\*\*\*\* Object: Database [Medtest\_db] Script Date: 17.06.2013 3:42:34 \*\*\*\*\*\*/

CREATE DATABASE [Medtest\_db]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'Medtest\_db', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\Medtest\_db.mdf' , SIZE = 5120KB , MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )

LOG ON

( NAME = N'Medtest\_db\_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\Medtest\_db\_log.ldf' , SIZE = 4672KB , MAXSIZE = 2048GB , FILEGROWTH = 10%)

GO

ALTER DATABASE [Medtest\_db] SET COMPATIBILITY\_LEVEL = 110

GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))

begin

EXEC [Medtest\_db].[dbo].[sp\_fulltext\_database] @action = 'enable'

end

GO

ALTER DATABASE [Medtest\_db] SET ANSI\_NULL\_DEFAULT OFF

GO

ALTER DATABASE [Medtest\_db] SET ANSI\_NULLS OFF

GO

ALTER DATABASE [Medtest\_db] SET ANSI\_PADDING OFF

GO

ALTER DATABASE [Medtest\_db] SET ANSI\_WARNINGS OFF

GO

ALTER DATABASE [Medtest\_db] SET ARITHABORT OFF

GO

ALTER DATABASE [Medtest\_db] SET AUTO\_CLOSE OFF

GO

ALTER DATABASE [Medtest\_db] SET AUTO\_CREATE\_STATISTICS ON

GO

ALTER DATABASE [Medtest\_db] SET AUTO\_SHRINK OFF

GO

ALTER DATABASE [Medtest\_db] SET AUTO\_UPDATE\_STATISTICS ON

GO

ALTER DATABASE [Medtest\_db] SET CURSOR\_CLOSE\_ON\_COMMIT OFF

GO

ALTER DATABASE [Medtest\_db] SET CURSOR\_DEFAULT GLOBAL

GO

ALTER DATABASE [Medtest\_db] SET CONCAT\_NULL\_YIELDS\_NULL OFF

GO

ALTER DATABASE [Medtest\_db] SET NUMERIC\_ROUNDABORT OFF

GO

ALTER DATABASE [Medtest\_db] SET QUOTED\_IDENTIFIER OFF

GO

ALTER DATABASE [Medtest\_db] SET RECURSIVE\_TRIGGERS OFF

GO

ALTER DATABASE [Medtest\_db] SET DISABLE\_BROKER

GO

ALTER DATABASE [Medtest\_db] SET AUTO\_UPDATE\_STATISTICS\_ASYNC OFF

GO

ALTER DATABASE [Medtest\_db] SET DATE\_CORRELATION\_OPTIMIZATION OFF

GO

ALTER DATABASE [Medtest\_db] SET TRUSTWORTHY OFF

GO

ALTER DATABASE [Medtest\_db] SET ALLOW\_SNAPSHOT\_ISOLATION OFF

GO

ALTER DATABASE [Medtest\_db] SET PARAMETERIZATION SIMPLE

GO

ALTER DATABASE [Medtest\_db] SET READ\_COMMITTED\_SNAPSHOT OFF

GO

ALTER DATABASE [Medtest\_db] SET HONOR\_BROKER\_PRIORITY OFF

GO

ALTER DATABASE [Medtest\_db] SET RECOVERY FULL

GO

ALTER DATABASE [Medtest\_db] SET MULTI\_USER

GO

ALTER DATABASE [Medtest\_db] SET PAGE\_VERIFY CHECKSUM

GO

ALTER DATABASE [Medtest\_db] SET DB\_CHAINING OFF

GO

ALTER DATABASE [Medtest\_db] SET FILESTREAM( NON\_TRANSACTED\_ACCESS = OFF )

GO

ALTER DATABASE [Medtest\_db] SET TARGET\_RECOVERY\_TIME = 0 SECONDS

GO

ALTER DATABASE [Medtest\_db] SET READ\_WRITE

GO

USE [Medtest\_db]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Answers] Script Date: 17.06.2013 3:43:15 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[Answers](

[ID] [int] IDENTITY(1,1) NOT NULL,

[Фамилия участника] [int] NOT NULL,

[Название раздела] [int] NOT NULL,

[Вопрос] [int] NOT NULL,

[Результат 1] [bit] NULL,

[Результат 2] [bit] NULL,

[Результат 3] [bit] NULL,

[Результат 4] [bit] NULL,

[Результат 5] [bit] NULL,

[Результат текст] [bit] NULL,

[Ответ текст] [varchar](50) NULL,

[Итог] [bit] NOT NULL,

CONSTRAINT [PK\_Answers] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

ALTER TABLE [dbo].[Answers] WITH CHECK ADD CONSTRAINT [FK\_Answers\_Participants] FOREIGN KEY([Фамилия участника])

REFERENCES [dbo].[Participants] ([ID])

GO

ALTER TABLE [dbo].[Answers] CHECK CONSTRAINT [FK\_Answers\_Participants]

GO

ALTER TABLE [dbo].[Answers] WITH CHECK ADD CONSTRAINT [FK\_Answers\_Questions] FOREIGN KEY([Вопрос])

REFERENCES [dbo].[Questions] ([ID])

GO

ALTER TABLE [dbo].[Answers] CHECK CONSTRAINT [FK\_Answers\_Questions]

GO

ALTER TABLE [dbo].[Answers] WITH CHECK ADD CONSTRAINT [FK\_Answers\_Sections] FOREIGN KEY([Название раздела])

REFERENCES [dbo].[Sections] ([ID])

GO

ALTER TABLE [dbo].[Answers] CHECK CONSTRAINT [FK\_Answers\_Sections]

GO

USE [Medtest\_db]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Attempts] Script Date: 17.06.2013 3:43:37 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Attempts](

[ID] [int] IDENTITY(1,1) NOT NULL,

[Фамилия участника] [int] NOT NULL,

[Название раздела] [int] NOT NULL,

[Количество попыток] [int] NOT NULL,

CONSTRAINT [PK\_Attempts] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Attempts] WITH CHECK ADD CONSTRAINT [FK\_Attempts\_Participants] FOREIGN KEY([Фамилия участника])

REFERENCES [dbo].[Participants] ([ID])

GO

ALTER TABLE [dbo].[Attempts] CHECK CONSTRAINT [FK\_Attempts\_Participants]

GO

ALTER TABLE [dbo].[Attempts] WITH CHECK ADD CONSTRAINT [FK\_Attempts\_Sections] FOREIGN KEY([Название раздела])

REFERENCES [dbo].[Sections] ([ID])

GO

ALTER TABLE [dbo].[Attempts] CHECK CONSTRAINT [FK\_Attempts\_Sections]

GO

USE [Medtest\_db]

GO

/\*\*\*\*\*\* Object: Table [dbo].[DictCateg] Script Date: 17.06.2013 3:44:01 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[DictCateg](

[ID] [int] NOT NULL,

[Категория] [varchar](30) NOT NULL,

CONSTRAINT [PK\_DictCateg] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

USE [Medtest\_db]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Participants] Script Date: 17.06.2013 3:44:16 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[Participants](

[ID] [int] IDENTITY(1,1) NOT NULL,

[Фамилия] [varchar](30) NOT NULL,

[Имя] [varchar](30) NOT NULL,

[Отчество] [varchar](30) NOT NULL,

[Дата рождения] [date] NOT NULL,

[Категория] [int] NOT NULL,

[Отделение] [int] NOT NULL,

[Пароль] [varchar](10) NOT NULL,

CONSTRAINT [PK\_Participants] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

ALTER TABLE [dbo].[Participants] WITH CHECK ADD CONSTRAINT [FK\_Participants\_DictCateg] FOREIGN KEY([Категория])

REFERENCES [dbo].[DictCateg] ([ID])

GO

ALTER TABLE [dbo].[Participants] CHECK CONSTRAINT [FK\_Participants\_DictCateg]

GO

ALTER TABLE [dbo].[Participants] WITH CHECK ADD CONSTRAINT [FK\_Participants\_Passwords] FOREIGN KEY([Отделение])

REFERENCES [dbo].[Passwords] ([ID])

GO

ALTER TABLE [dbo].[Participants] CHECK CONSTRAINT [FK\_Participants\_Passwords]

GO

USE [Medtest\_db]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Passwords] Script Date: 17.06.2013 3:44:49 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[Passwords](

[ID] [int] IDENTITY(1,1) NOT NULL,

[Название отделения] [varchar](30) NOT NULL,

[Пароль] [varchar](10) NOT NULL,

CONSTRAINT [PK\_Passwords] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

USE [Medtest\_db]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Questions] Script Date: 17.06.2013 3:45:07 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[Questions](

[ID] [int] IDENTITY(1,1) NOT NULL,

[Компетенция] [int] NOT NULL,

[Вопрос] [varchar](900) NOT NULL,

[Приложение к вопросу] [varbinary](max) NULL,

[Убран из теста] [bit] NOT NULL,

[Ответ 1] [varchar](150) NULL,

[Цель 1] [bit] NULL,

[Ответ 2] [varchar](150) NULL,

[Цель 2] [bit] NULL,

[Ответ 3] [varchar](150) NULL,

[Цель 3] [bit] NULL,

[Ответ 4] [varchar](150) NULL,

[Цель 4] [bit] NULL,

[Ответ 5] [varchar](150) NULL,

[Цель 5] [bit] NULL,

[Цель текст] [varchar](50) NULL,

CONSTRAINT [PK\_Questions] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

ALTER TABLE [dbo].[Questions] ADD CONSTRAINT [DF\_Questions\_Убран из теста] DEFAULT ((0)) FOR [Убран из теста]

GO

ALTER TABLE [dbo].[Questions] WITH CHECK ADD CONSTRAINT [FK\_Questions\_Сompetence] FOREIGN KEY([Компетенция])

REFERENCES [dbo].[Сompetence] ([ID])

GO

ALTER TABLE [dbo].[Questions] CHECK CONSTRAINT [FK\_Questions\_Сompetence]

GO

ALTER TABLE [dbo].[Questions] WITH CHECK ADD CONSTRAINT [CK\_Questions] CHECK (([Ответ 1] IS NOT NULL AND [Ответ 2] IS NOT NULL AND ([Цель 1] IS NOT NULL AND [Цель 2] IS NOT NULL) OR [Цель текст] IS NOT NULL))

GO

ALTER TABLE [dbo].[Questions] CHECK CONSTRAINT [CK\_Questions]

GO

USE [Medtest\_db]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Results] Script Date: 17.06.2013 3:45:26 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Results](

[ID] [int] IDENTITY(1,1) NOT NULL,

[Фамилия участника] [int] NOT NULL,

[Название раздела] [int] NOT NULL,

[Итог] [int] NOT NULL,

[Процент 1] [int] NULL,

[Процент 2] [int] NULL,

[Процент 3] [int] NULL,

[Процент 4] [int] NULL,

[Процент 5] [int] NULL,

[Процент 6] [int] NULL,

[Процент 7] [int] NULL,

[Процент 8] [int] NULL,

[Процент 9] [int] NULL,

[Процент 10] [int] NULL,

CONSTRAINT [PK\_Results] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Results] WITH CHECK ADD CONSTRAINT [FK\_Results\_Participants] FOREIGN KEY([Фамилия участника])

REFERENCES [dbo].[Participants] ([ID])

GO

ALTER TABLE [dbo].[Results] CHECK CONSTRAINT [FK\_Results\_Participants]

GO

ALTER TABLE [dbo].[Results] WITH CHECK ADD CONSTRAINT [FK\_Results\_Sections] FOREIGN KEY([Название раздела])

REFERENCES [dbo].[Sections] ([ID])

GO

ALTER TABLE [dbo].[Results] CHECK CONSTRAINT [FK\_Results\_Sections]

GO

USE [Medtest\_db]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Sections] Script Date: 17.06.2013 3:45:42 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[Sections](

[ID] [int] IDENTITY(1,1) NOT NULL,

[Название] [varchar](50) NOT NULL,

[Время] [int] NOT NULL,

[Количество попыток] [int] NOT NULL,

[Запустить тестирование] [bit] NOT NULL,

[Компетенция 1] [int] NULL,

[Вопросов 1] [int] NULL,

[Компетенция 2] [int] NULL,

[Вопросов 2] [int] NULL,

[Компетенция 3] [int] NULL,

[Вопросов 3] [int] NULL,

[Компетенция 4] [int] NULL,

[Вопросов 4] [int] NULL,

[Компетенция 5] [int] NULL,

[Вопросов 5] [int] NULL,

[Компетенция 6] [int] NULL,

[Вопросов 6] [int] NULL,

[Компетенция 7] [int] NULL,

[Вопросов 7] [int] NULL,

[Компетенция 8] [int] NULL,

[Вопросов 8] [int] NULL,

[Компетенция 9] [int] NULL,

[Вопросов 9] [int] NULL,

[Компетенция 10] [int] NULL,

[Вопросов 10] [int] NULL,

CONSTRAINT [PK\_Sections] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

ALTER TABLE [dbo].[Sections] ADD CONSTRAINT [DF\_Sections\_Компетенция 1] DEFAULT ((0)) FOR [Компетенция 1]

GO

ALTER TABLE [dbo].[Sections] ADD CONSTRAINT [DF\_Sections\_Компетенция 2] DEFAULT ((0)) FOR [Компетенция 2]

GO

ALTER TABLE [dbo].[Sections] ADD CONSTRAINT [DF\_Sections\_Компетенция 3] DEFAULT ((0)) FOR [Компетенция 3]

GO

ALTER TABLE [dbo].[Sections] ADD CONSTRAINT [DF\_Sections\_Компетенция 4] DEFAULT ((0)) FOR [Компетенция 4]

GO

ALTER TABLE [dbo].[Sections] ADD CONSTRAINT [DF\_Sections\_Компетенция 5] DEFAULT ((0)) FOR [Компетенция 5]

GO

ALTER TABLE [dbo].[Sections] ADD CONSTRAINT [DF\_Sections\_Компетенция 6] DEFAULT ((0)) FOR [Компетенция 6]

GO

ALTER TABLE [dbo].[Sections] ADD CONSTRAINT [DF\_Sections\_Компетенция 7] DEFAULT ((0)) FOR [Компетенция 7]

GO

ALTER TABLE [dbo].[Sections] ADD CONSTRAINT [DF\_Sections\_Компетенция 8] DEFAULT ((0)) FOR [Компетенция 8]

GO

ALTER TABLE [dbo].[Sections] ADD CONSTRAINT [DF\_Sections\_Компетенция 9] DEFAULT ((0)) FOR [Компетенция 9]

GO

ALTER TABLE [dbo].[Sections] ADD CONSTRAINT [DF\_Sections\_Компетенция 10] DEFAULT ((0)) FOR [Компетенция 10]

GO

ALTER TABLE [dbo].[Sections] WITH CHECK ADD CONSTRAINT [FK\_Sections\_Сompetence] FOREIGN KEY([Компетенция 1])

REFERENCES [dbo].[Сompetence] ([ID])

GO

ALTER TABLE [dbo].[Sections] CHECK CONSTRAINT [FK\_Sections\_Сompetence]

GO

ALTER TABLE [dbo].[Sections] WITH CHECK ADD CONSTRAINT [FK\_Sections\_Сompetence1] FOREIGN KEY([Компетенция 2])

REFERENCES [dbo].[Сompetence] ([ID])

GO

ALTER TABLE [dbo].[Sections] CHECK CONSTRAINT [FK\_Sections\_Сompetence1]

GO

ALTER TABLE [dbo].[Sections] WITH CHECK ADD CONSTRAINT [FK\_Sections\_Сompetence2] FOREIGN KEY([Компетенция 3])

REFERENCES [dbo].[Сompetence] ([ID])

GO

ALTER TABLE [dbo].[Sections] CHECK CONSTRAINT [FK\_Sections\_Сompetence2]

GO

ALTER TABLE [dbo].[Sections] WITH CHECK ADD CONSTRAINT [FK\_Sections\_Сompetence3] FOREIGN KEY([Компетенция 4])

REFERENCES [dbo].[Сompetence] ([ID])

GO

ALTER TABLE [dbo].[Sections] CHECK CONSTRAINT [FK\_Sections\_Сompetence3]

GO

ALTER TABLE [dbo].[Sections] WITH CHECK ADD CONSTRAINT [FK\_Sections\_Сompetence4] FOREIGN KEY([Компетенция 5])

REFERENCES [dbo].[Сompetence] ([ID])

GO

ALTER TABLE [dbo].[Sections] CHECK CONSTRAINT [FK\_Sections\_Сompetence4]

GO

ALTER TABLE [dbo].[Sections] WITH CHECK ADD CONSTRAINT [FK\_Sections\_Сompetence5] FOREIGN KEY([Компетенция 6])

REFERENCES [dbo].[Сompetence] ([ID])

GO

ALTER TABLE [dbo].[Sections] CHECK CONSTRAINT [FK\_Sections\_Сompetence5]

GO

ALTER TABLE [dbo].[Sections] WITH CHECK ADD CONSTRAINT [FK\_Sections\_Сompetence6] FOREIGN KEY([Компетенция 7])

REFERENCES [dbo].[Сompetence] ([ID])

GO

ALTER TABLE [dbo].[Sections] CHECK CONSTRAINT [FK\_Sections\_Сompetence6]

GO

ALTER TABLE [dbo].[Sections] WITH CHECK ADD CONSTRAINT [FK\_Sections\_Сompetence7] FOREIGN KEY([Компетенция 8])

REFERENCES [dbo].[Сompetence] ([ID])

GO

ALTER TABLE [dbo].[Sections] CHECK CONSTRAINT [FK\_Sections\_Сompetence7]

GO

ALTER TABLE [dbo].[Sections] WITH CHECK ADD CONSTRAINT [FK\_Sections\_Сompetence8] FOREIGN KEY([Компетенция 9])

REFERENCES [dbo].[Сompetence] ([ID])

GO

ALTER TABLE [dbo].[Sections] CHECK CONSTRAINT [FK\_Sections\_Сompetence8]

GO

ALTER TABLE [dbo].[Sections] WITH CHECK ADD CONSTRAINT [FK\_Sections\_Сompetence9] FOREIGN KEY([Компетенция 10])

REFERENCES [dbo].[Сompetence] ([ID])

GO

ALTER TABLE [dbo].[Sections] CHECK CONSTRAINT [FK\_Sections\_Сompetence9]

GO

USE [Medtest\_db]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Сompetence] Script Date: 17.06.2013 3:46:08 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[Сompetence](

[ID] [int] IDENTITY(1,1) NOT NULL,

[Компетенция] [varchar](400) NOT NULL,

CONSTRAINT [PK\_Сompetence] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO