

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ МОСКОВСКОЙ ОБЛАСТИ**

**Государственное бюджетное профессиональное образовательное**

**учреждение Московской области**

**«Люберецкий техникум имени Героя Советского Союза,**

**лётчика-космонавта Ю.А. Гагарина»**

**КУРСОВАЯ РАБОТА**

**по МДК.01.01 «Разработка программных модулей»**

**по МДК.01.03 «Разработка мобильных приложений»**

специальность

09.02.07 «Информационные системы и программирование»

**Тема:**

Создание клиентского приложения WPF для тестирования на языке С#

Группа 185

Выполнил/а/ студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Журин Артем Вячеславович

(подпись) (ФИО полностью)

Руководитель Жирнова Юлия Витальевна

(подпись) (ФИО полностью)

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Люберцы-2021 год

**Содержание**

[Введение 3](#_heading=h.gjdgxs)

[1.](#_heading=h.30j0zll) Теоретическая часть 4

[1.1.](#_heading=h.1fob9te) Актуальность (кому может понадобиться, зачем нужно) 4

[1.2.](#_heading=h.3znysh7) Теория (что необходимо знать, чтобы сделать приложение) 4

[1.3.](#_heading=h.2et92p0) Особенности IDE (в чем писали приложение) 5

[2.](#_heading=h.tyjcwt) Практическая часть 6

[2.1.](#_heading=h.3dy6vkm) Постановка задачи (Что нужно сделать) 6

[2.2.](#_heading=h.1t3h5sf) Структура приложения (и каких файлов состоит, зачем каждый файл) 7

[2.4.1.](#_heading=h.17dp8vu) [Ваше приложение](#_heading=h.1t3h5sf) (скриншоты результатов, что делает) [8](#_heading=h.17dp8vu)

[Выводы 29](#_heading=h.1ksv4uv)

[Список литературы 30](#_heading=h.44sinio)

**Введение**

В современном мире большое значение имеет контроль знаний студентов, работников и т. д. В данном контексте контроль понимается как система научно обоснованной проверки результатов знаний.

Есть множество форм контроля – экзамен, зачет, аттестация, контрольная. Но в эпоху развития информационного общества, интернет-технологий, глобальной компьютеризации актуальными становятся разнообразные системы тестирования, которые могут заменить традиционные методы контроля и преподавания. Подобные компьютерные системы тестирования намного удобнее использовать с системой образования.

Тест позволяет провести проверку знаний опросив всех участников по вопросам нужного материала в одинаковых условия.

**1.Теоретическая часть**

**1.1. Актуальность**

В настоящее время во многих учебных заведениях широко используются рейтинговые системы оценки знаний студентов, которые направлены на повышение эффективности процесса обучения. Для получения рейтинговой оценки необходимо пройти несколько этапов контроля.

Компьютерная система тестирования, конечно, весьма удобна для применения ее в процессе обучения, но в подавляющем большинстве данные системы разработаны для непрограммирующих пользователей по принципу вопрос ответ.

Для проверки практических умений и навыков студента используется задача, то есть тестовое задание, для ответа на который необходимо провести определенные интеллектуальные действия, связанные с решением задачи. При этом возникает сложность оценки предоставленных ответов в случаях, когда этот ответ может быть неоднозначным: фрагмент кода на языке программирования, структурная схема цифрового устройства.

Кроме того, при проведении тестирования знаний как дистанционно, так и локально, важным является обеспечение уровня безопасности (авторизация, защита базы тестов и т. д.), а также организация работы как со студентами, так и с преподавателями.

Таким образом, работа, направленная на разработку новых способов тестирования знаний и разработку на их основе систем тестирования знаний, является актуальной.

**1.2. Теория**

Необходимо освоить C# Windows Forms или WPF, т.к. они ничем не отличаются. Разница между ними заключается главным образом в масштабируемости и производительности, поскольку WPF может рендериться быстрее по сравнению с формами окон, сложностью и поддержкой. Еще нужно освоить MSSQL. С помощью него пользователь подключает приложение к серверу БД, где находится вся его информация для приложения.

Перед созданием модели подобной программы тестирования нужно определить набор функций и требований:

* реализация эффективного тестирования знаний;
* точная и способная к адаптации оценка результата тестирования;
* способность легко создавать и модифицировать тесты;
* возможность одновременного тестирования неограниченного количества пользователей;
* большое количество тестов, вопросов, заданий;

**1.3. Особенности IDE**

Был использован C++ и интерфейс приложения Windows Forms, а также БД в MSSQL.

C# — объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.

Windows Forms — интерфейс программирования приложений, отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework. Данный интерфейс упрощает доступ к элементам интерфейса Microsoft Windows за счет создания обёртки для существующего Win32 API в управляемом коде.

Microsoft SQL Server — система управления реляционными базами данных, разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов с расширениями.

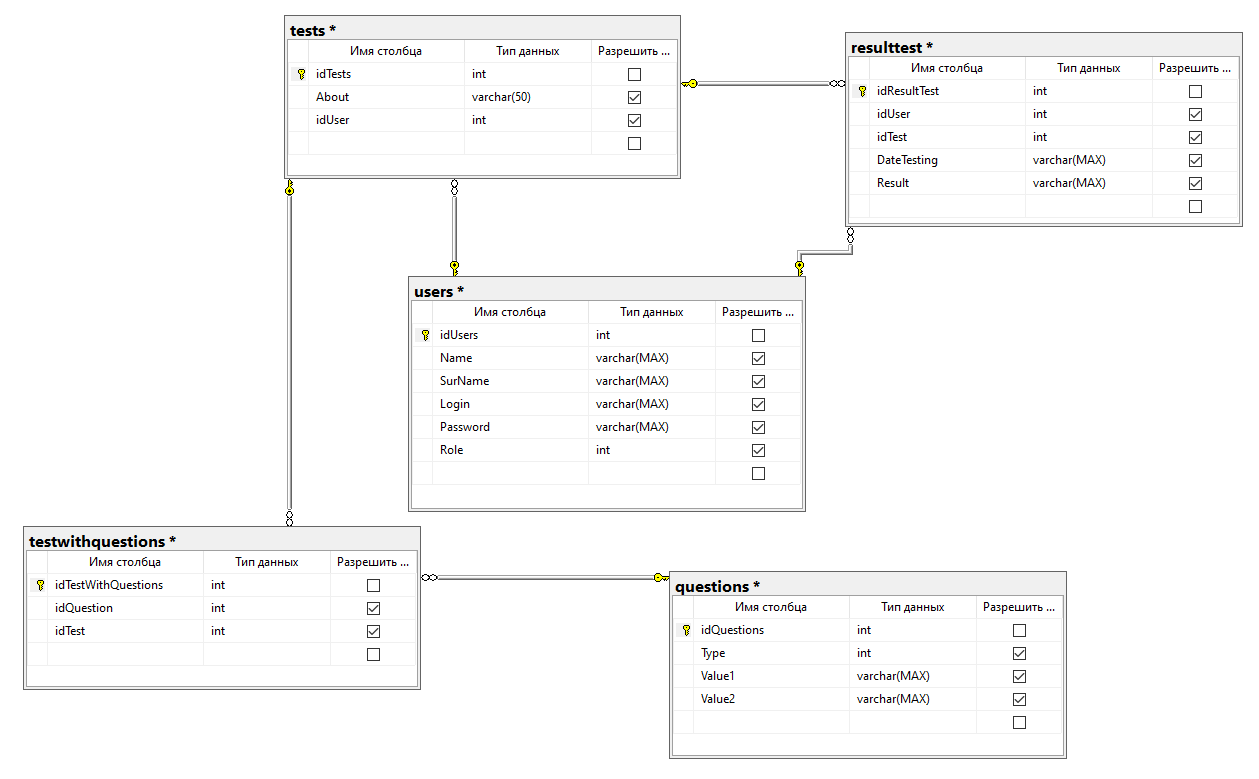
**2. Практическая часть**

**2.1. Постановка задачи**

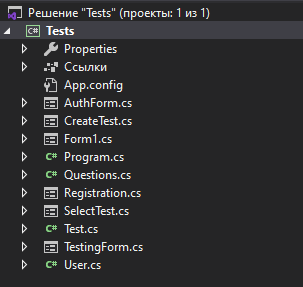
Необходимо разработать WEB-систему тестирования студентов со следующими возможностями:

1. Категорирование страниц по различным терминам. С возможностью для преподавателей создать свои категории сортировки и включать в них термины.
2. Создание теста. С возможностями: сортировки вопросов в случайном порядке по желанию преподавателя, отображения результатов и правильных ответов, вывода оценки по завершению тестирования.
3. Создание вопроса с выбором количества правильных вариантов ответа. С возможностью размещения ответов в случайном порядке и с возможностью добавления данного вопроса в уже существующий тест.
4. Использование групповой политики доступа для обеспечения безопасности. Студент, незарегистрированный пользователь, администратор и преподаватель.
5. Цель – является проверка, анализ и определение уровня знаний. Эта система – прикладная программа, которая должна быть полезна для пользователей в том, что они смогут пройти тест онлайн.
6. Назначение максимального количества баллов за каждый полностью правильный ответ.

**2.2. Структура приложения**



ER-Диаграмма

****

Файлы проекта

AuthForm.cs - форма для авторизации

CreateTest.cs - форма для создания теста

Form1.cs -основная форма с тестами после авторизации

Program.cs - Основная точка для входа в приложение

Questions.cs – Тут находятся категория типов с выбором ответа

Registration.cs - форма для регистрации

SelectTest.cs – форма для выбирания теста

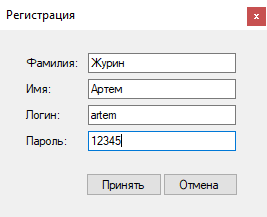
Tests.cs – Класс для заполнения таблицы из БД в приложение

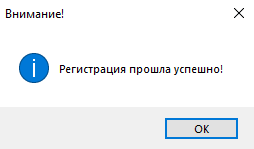
TestingForm.cs – форма с прохождением самого теста

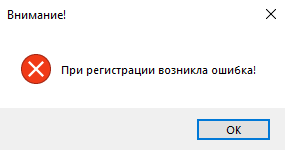
User.cs – Тут находятся 2 роли (admin, user), если пройти авторизацию, то выдает пользователь роль-user, а если надо выставить admin, то надо поменять за место user поставить admin. В БД они помечены как 1(user) и 0(admin)

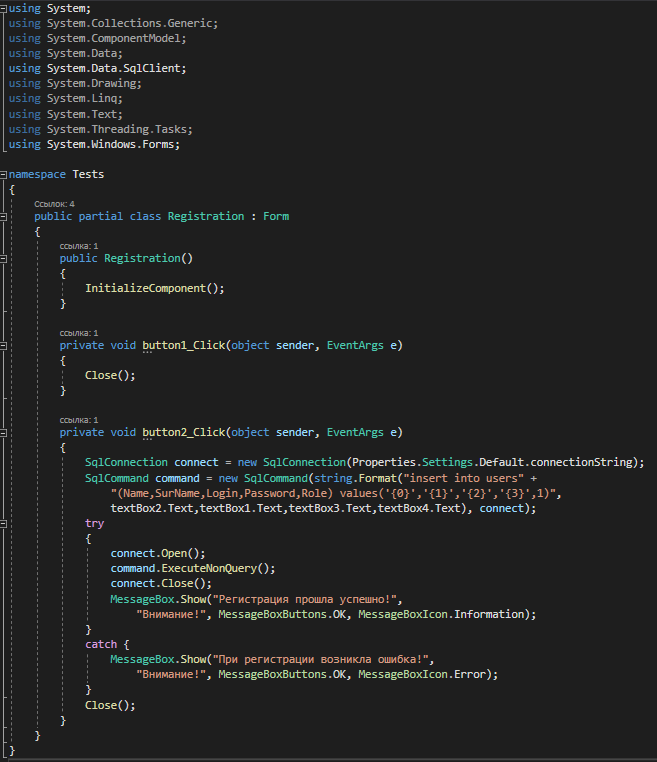
**2.4.1 Ваше приложение**

**Регистрация**

****

****

****

****

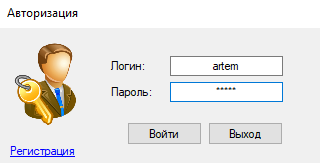
В Авторизации присутствуют 4 Textbox для заполнения информации профиля для пользователя и с помощью кода

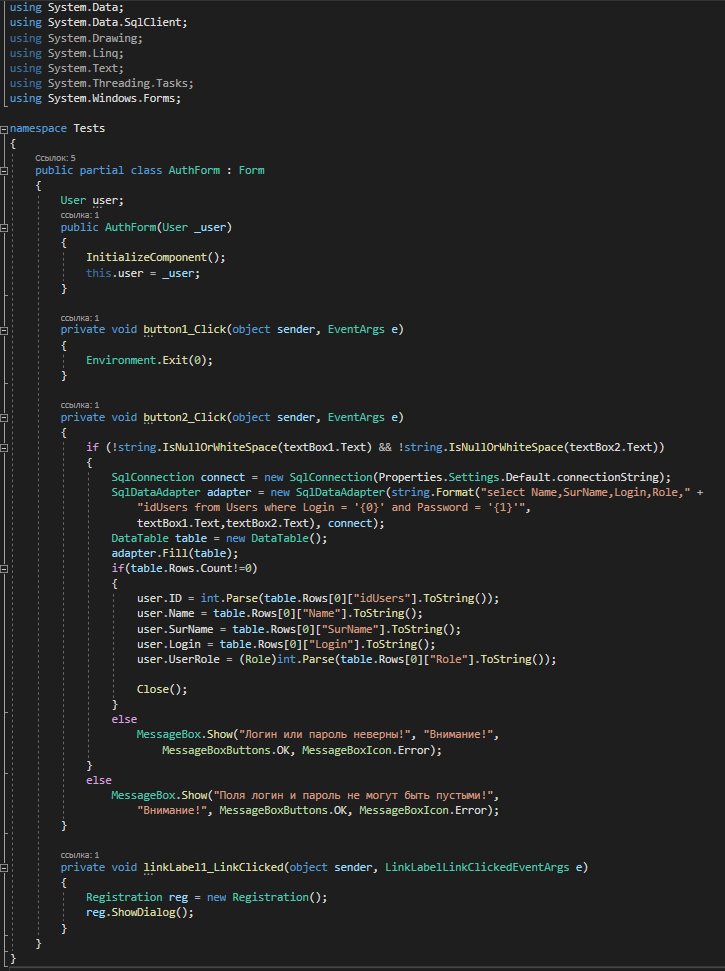
SqlConnection connect = new SqlConnection(Properties.Settings.Default.connectionString) подключаемся к БД, и кода

SqlCommand command = new SqlCommand(string.Format("insert into users" + "(Name,SurName,Login,Password,Role) values('{0}','{1}','{2}','{3}',1)",

textBox2.Text,textBox1.Text,textBox3.Text,textBox4.Text), connect) подключимся уже к самой таблице, чтоб весь введенный текст после нажатия кнопки «Принять» сохранился в табличку БД. Добавляется проверка try и catch, чтоб после всего заполнения Textbox и нажатие кнопки «Принять», появлялось окно «Внимание» Регистрация прошла успешно! Если Textbox не все будут заполнены, то будет появляться окно «Внимание» При регистрации возникла ошибка!

**Авторизация**

****

****

Подключаемся к БД и с помощью кода **SqlDataAdapter adapter = new SqlDataAdapter(string.Format("select Name,SurName,Login,Role," +**

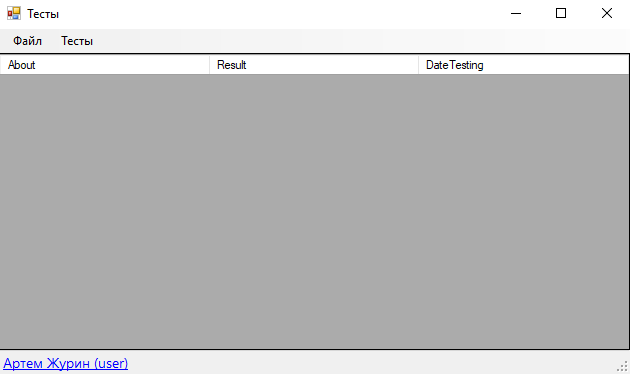
**"idUsers from Users where Login = '{0}' and Password = '{1}'",**

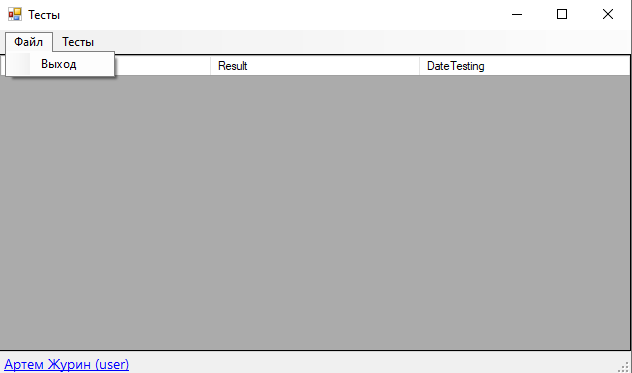
**textBox1.Text,textBox2.Text), connect) выберет запись из БД и заполнит ее, после нажатия кнопки «Войти».**

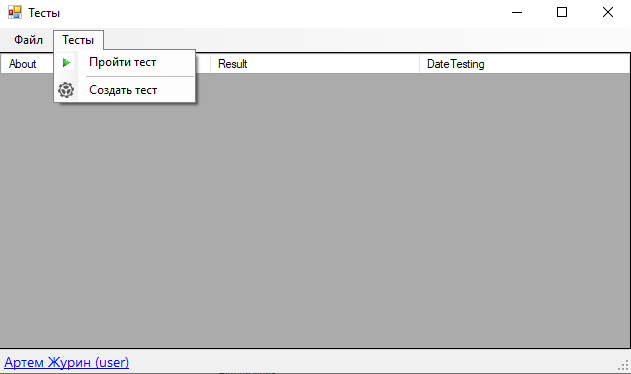
**DataTable это представляет одну таблицу данных в памяти, а Fill обновляет строки в указанном диапазоне с использованием имени DataTable.**

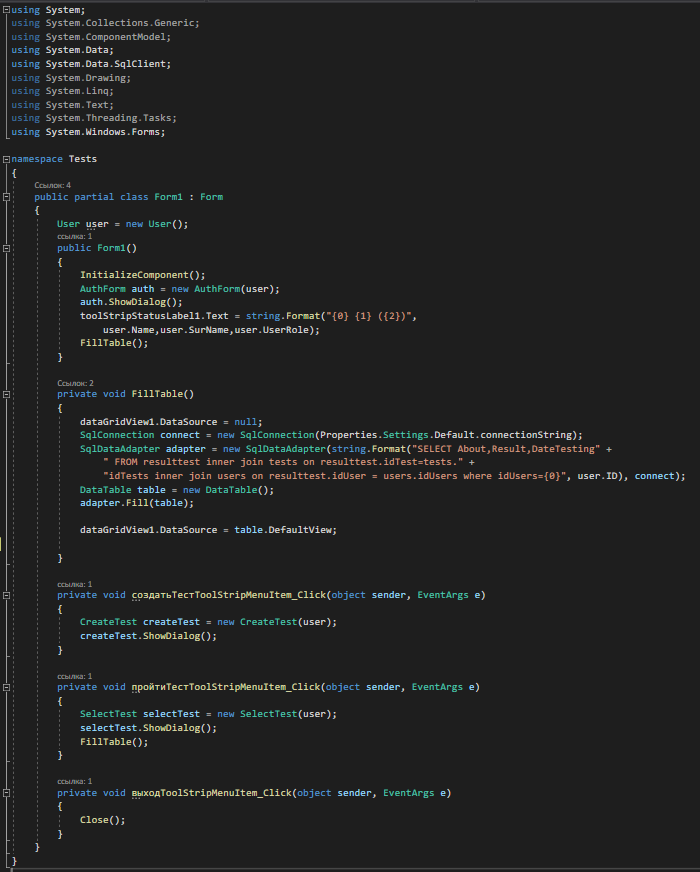
Нажатие на ссылку «Регистрация» откроется окно самой регистрации, а срабатывает это с помощью команды ShowDialog()

**Основная форма**

****

****

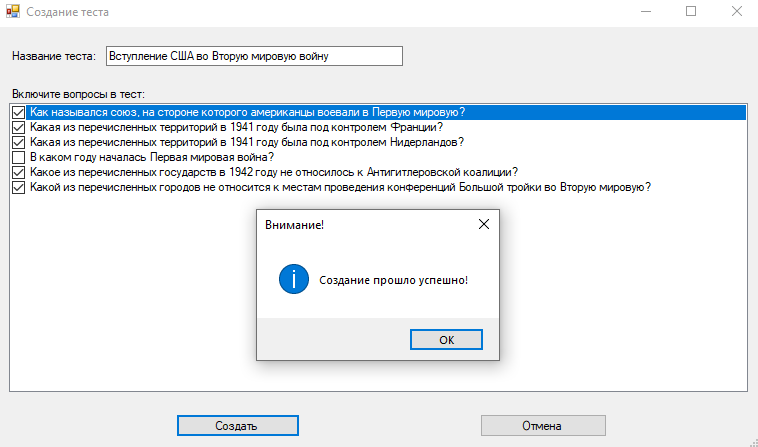
****

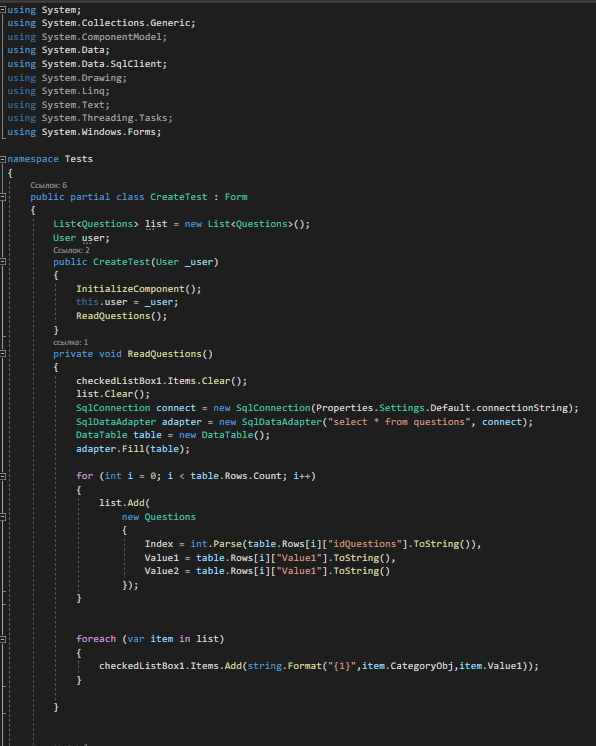
****

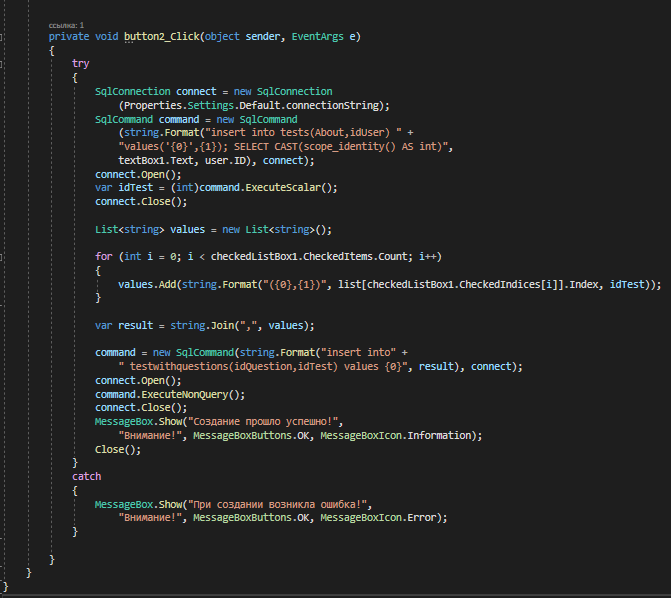
С помощью кода dataGridView1.DataSource = null список результатов будет пустым т.к. тестирование еще не пройдено никем.

SqlDataAdapter adapter = new SqlDataAdapter(string.Format("SELECT About,Result,DateTesting" + " FROM resulttest inner join tests on resulttest.idTest=tests." + "idTests inner join users on resulttest.idUser = users.idUsers where idUsers={0}", user.ID), connect). Тут будет выбирать About, Result, DateTesting из таблицы resulttest и добавлять в dataGridView1 каждый раз при прохождении тестирования результаты.

**Создание теста**

****

****

****

Название текста это textbox, при написания любого текста и нажатии кнопки создать он его сохранит.

Вопросы выводиться с помощью элемента списка checkedListBox1 и кода:

**foreach (var item in list)**

**{**

**checkedListBox1.Items.Add(String.Format("{1}",item.CategoryObj,item.Value1);**

**}**

**С помощьюкода для Button «Создать»**  SqlCommand command = new SqlComma (string.Format("insert into tests(About,idUser) " +" values('{0}',{1}); SELECT CAST(scope\_identity() AS int)", textBox1.Text, user.ID), connect);

Будет отправлять в БД данные и сохранять в таблицу «tests» по нажатию этой кнопки.

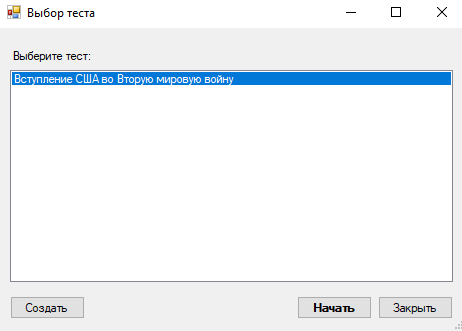
var idTest = (int)command.ExecuteScalar() - выполняет команду и возвращает значение из первого столбца первой строки любого результирующего набора.

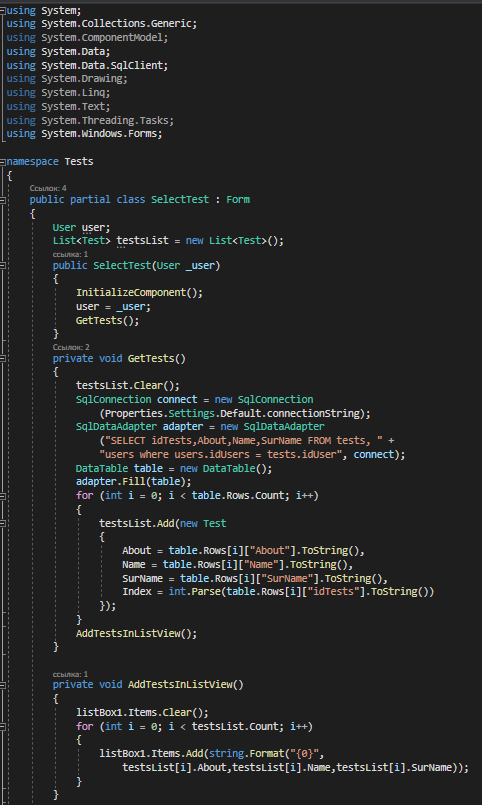
var result = string.Join(",", values) - сцепляет элементы созданной коллекцией.

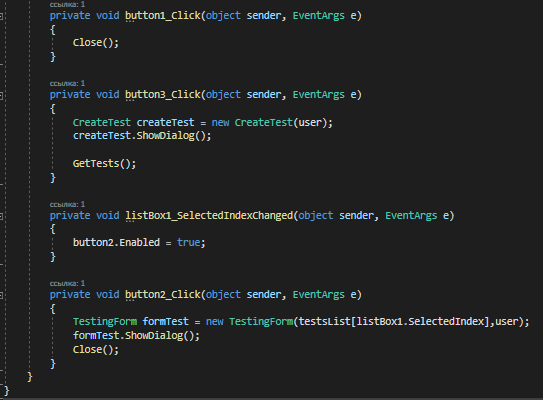
command = new SqlCommand(string.Format("insert into" +

" testwithquestions(idQuestion,idTest) values {0}", result), connect) – добавляет в БД таблицу testwithquestions записи, IdTest – создает номер теста который был создан, а IdQuestion какие тесты были внесены в этот тест.

**Выбора теста**

****

****

****

listBox1 добавляет уже созданный тест в форму «Выбор теста»

Чтоб добавить тест в listBox1, нужен код:

private void AddTestsInListView()

{

listBox1.Items.Clear();

for (int i = 0; i < testsList.Count; i++)

{

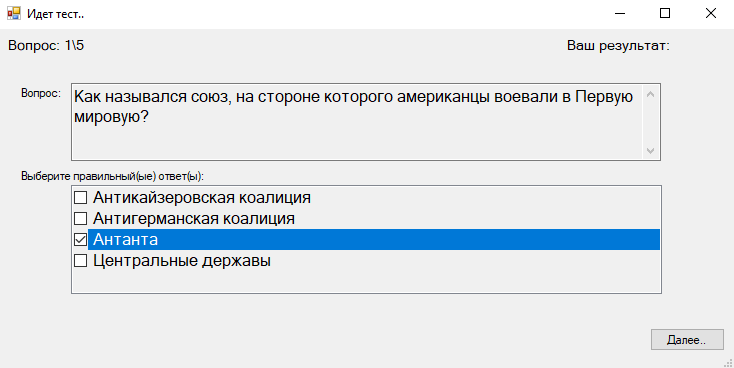
listBox1.Items.Add(string.Format("{0}",

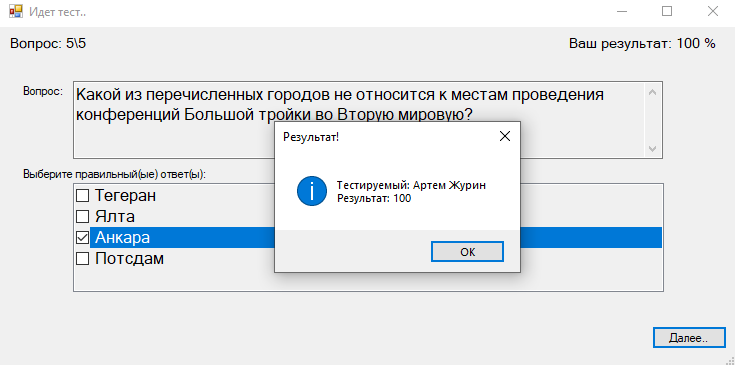
testsList[i].About,testsList[i].Name,testsList[i].SurName));

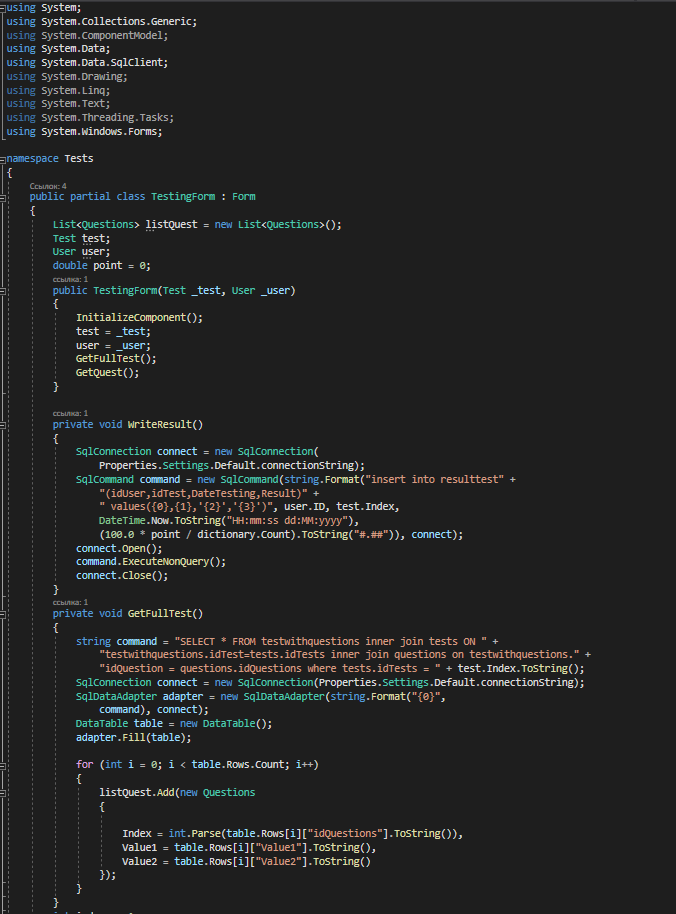
}

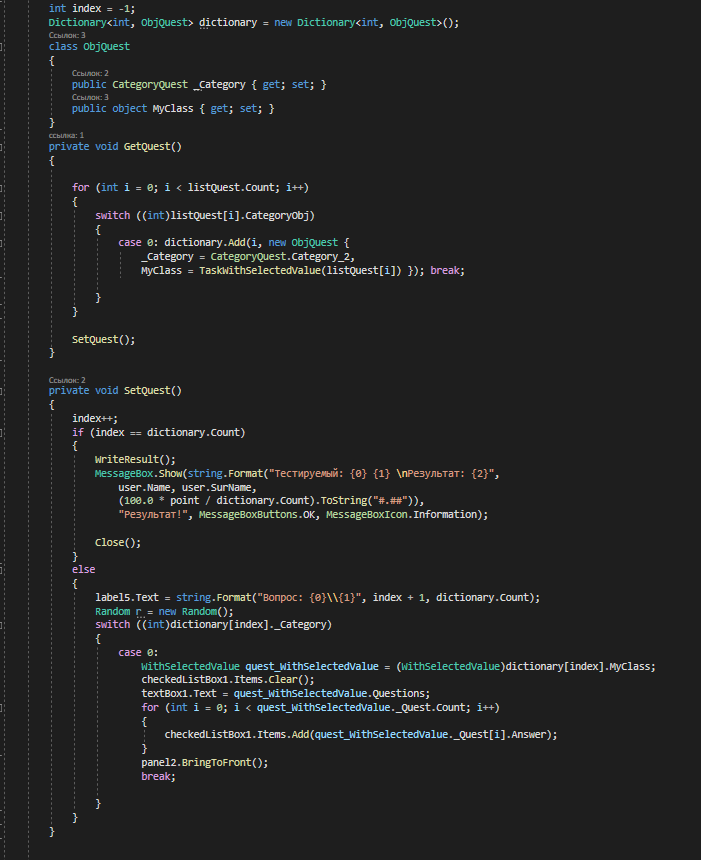
}

**Процесс тестирование**

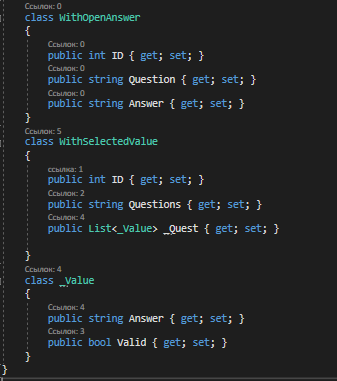
****

****

****

****

****

****

Для выполнения запроса программы:

for (int i = 0; i < listQuest.Count; i++)

{

switch ((int)listQuest[i].CategoryObj)

{

case 0: dictionary.Add(i, new ObjQuest {

\_Category = CategoryQuest.Category\_2,

MyClass = TaskWithSelectedValue(listQuest[i]) }); break;

}

}

string command = "SELECT \* FROM testwithquestions inner join tests ON " +     "testwithquestions.idTest=tests.idTests inner join questions on testwithquestions." +  idQuestion = questions.idQuestions where tests.idTests = " + test.Index.ToString() - Присоединяет вопросы и ответы по ID и добавляет в тестирование по порядку

Для добавления результата с расчетом и имя фамилия тестируемого, понадобиться код:

index++;

if (index == dictionary.Count)

{

WriteResult();

MessageBox.Show(string.Format("Тестируемый: {0} {1} \nРезультат: {2}",

user.Name, user.SurName,

(100.0 \* point / dictionary.Count).ToString("#.##")),

"Результат!", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

Продолжения кода с результатом, также добавляется подсчет вопросов где находится пользователь и вопрос с выбором ответа:

**else**

**{**

**label5.Text = string.Format("Вопрос: {0}\\{1}", index + 1, dictionary.Count);**

**Random r = new Random();**

**switch ((int)dictionary[index].\_Category)**

**{**

**case 0:**

**WithSelectedValue quest\_WithSelectedValue = (WithSelectedValue)dictionary[index].MyClass;**

**checkedListBox1.Items.Clear();**

**textBox1.Text = quest\_WithSelectedValue.Questions;**

**for (int i = 0; i < quest\_WithSelectedValue.\_Quest.Count; i++)**

**{**

**checkedListBox1.Items.Add(quest\_WithSelectedValue.\_Quest[i].Answer);**

**}**

**panel2.BringToFront();**

**break;**

**}**

**}**

**}**

**Данный код нужен, чтоб выбирался правильный ответ и неправильный ответ, в БД это помечено как - (неправильный) и + (правильный).**

**private WithSelectedValue TaskWithSelectedValue(Questions quest)**

**{**

**var array = quest.Value2.Split(new char[] { '\n', '\r' }, StringSplitOptions.RemoveEmptyEntries);**

**var listt = new List<\_Value>();**

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

**{**

**if (array[i][0] == '+')**

**listt.Add(new \_Value { Answer = array[i].Remove(0, 1), Valid = true });**

**else**

**listt.Add(new \_Value { Answer = array[i].Remove(0, 1), Valid = false });**

**}**

**return new WithSelectedValue**

**{**

**ID = quest.Index,**

**Questions = quest.Value1,**

**\_Quest = listt**

**};**

**}**

Чтоб сделать переход на другой вопрос на кнопку «Далее», нужен код:

var \_cl = ((WithSelectedValue)dictionary[index].MyClass);

for (int i = 0; i < checkedListBox1.CheckedItems.Count; i++)

{

If (!\_cl.\_Quest.Find(x => x.Answer == checkedListBox1.Items[checkedListBox1.CheckedIndices[i]]).Valid)

{

break;

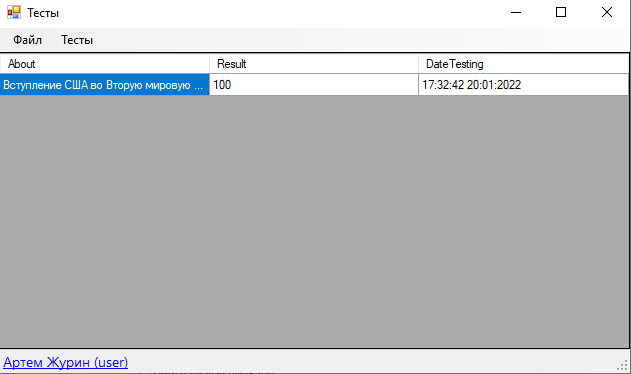
}

if (i == checkedListBox1.CheckedItems.Count - 1)

point++;

}

**Основная форма с итогом тестирования**

****

Отображение результатов из БД работает с помощью команды dataGridView1.DataSource = table.DefaultView, которая находится в коде Основной формы

**Выводы**

В ходе работы было проанализировано множество систем интернет тестирования, выявлено множество плюсов и минусов подобных систем. В дальнейшем будет разработан программный продукт, отвечающий всем выдвинутым требованиям, и включает в себя:

1. Создание теста с гибкими настройками.
2. Создание различных типов вопроса.
3. Назначение максимального количества баллов за каждый полностью правильный ответ.

**Список Литературы**

1. Компьютерные системы тестирования знаний студентов на различных этапах оценки успеваемости [Электронный ресурс].  [https://cyberleninka.ru...](https://cyberleninka.ru/article/v/kompyuternye-sistemy-testirovaniya-znaniy-studentov-na-razlichnyh-etapah-otsenki-uspevaemosti).
2. <https://www.youtube.com/watch?v=VBrvvBPYbj4>
3. <https://www.youtube.com/watch?v=VT76ruswmHo>
4. <https://www.youtube.com/watch?v=OQYSv_fcudI>
5. https://www.youtube.com/watch?v=\_GX7A2jIovI