**Санкт-Петербургское государственное  
 бюджетное профессиональное образовательное учреждение  
«Колледж информационных технологий»**

**отчёт ПО УЧЕБНОЙ ПРАКТИКЕ**

ПМ 01 Разработка программных модулей программного обеспечения  
для компьютерных сетей

МДК 01.01. Прикладное программирование

**СПЕЦИАЛЬНОСТЬ 09.02.03  
«ПРОГРАММИРОВАНИЕ В КОМПЬЮТЕРНЫХ СИСТЕМАХ»  
КВАЛИФИКАЦИЯ «ПРОГРАММИСТ»**

Руководитель учебной практики:

Преподаватель: Матысик И. А.  
Выполнил студент:  
Климов Иван  
Группа: 374

**ВВЕДЕНИЕ**

Учебная практика по МДК 01.01 «Прикладное программирование» являетсяобязательным учебным элементом для формирования общих и профессиональных компетенцией по модулю ПМ 01 «Разработка программных модулей программногообеспечения для компьютерных систем».

Практика проходила в дистанционной форме. Период прохождения практики с 25.05.2020 по 27.06.2020. Объем учебной практики составлял 50 часов. Тема учебной практики – разработка прикладного приложения «Универсальный тест» в среде Visual Studio C# с использованием СУБД MS SQL Server.

В отчете к учебной практике представлена четкая постановка задачи, представлены диаграммы, описывающие функциональную структуру программы и логическую модель данных, подробная инструкция пользователя и выводы по проделанной работе. Представлен программный код в приложении.

В результате прохождения учебной практики по МДК 01.01 были подтверждены общие и профессиональные компетенции.

Перечень общих компетенций:

ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.

ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.

ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.

ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.

ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.

ОК 6. Работать в коллективе и команде, эффективно общаться с коллегами, руководством, потребителями.

ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), результат выполнения заданий.

ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.

ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности.

Перечень профессиональных компетенций:

ПК 1.1. Выполнять разработку спецификаций отдельных компонент.

ПК 1.2. Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.

ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.4. Выполнять тестирование программных модулей.

ПК 1.5. Осуществлять оптимизацию программного кода модуля.

ПК 1.6. Разрабатывать компоненты проектной и технической документации с использованием графических языков спецификаций.

В результате прохождения учебной практики по модулю, был получен практический опыт в:

− разработки алгоритма поставленной задачи и реализации его средствами автоматизированного проектирования;

− разработки кода программного продукта на основе готовой спецификации на уровне модуля;

− использования инструментальных средств на этапе отладки программного продукта;

− проведения тестирования программного модуля по определенному сценарию.

В результате прохождения учебной практики, были получены умения:

− осуществлять разработку кода программного модуля на современных языках программирования;

− создавать программу по разработанному алгоритму как отдельный модуль;

− выполнять отладку и тестирование программы на уровне модуля;

− оформлять документацию на программные средства;

− использовать инструментальные средства для автоматизации оформления документации.

В результате прохождения учебной практики по модулю, были получены знания по следующим понятиям:

− основные этапы разработки программного обеспечения;

− основные принципы технологии структурного и объектно-ориентированного программирования;

− основные принципы отладки и тестирования программных продуктов;

− методы и средства разработки технической документации.

1. **ПОСТАНОВКА ЗАДАЧИ**

Неоходимо создать проект «Универсальный тест», позволяющий учителям и студентам авторизироваться в информационной системе. После чего дать возможность одним пользователям создавать тесты, и просмотривать их статистику прохождения, а другим юзерам позволить проходить эти тесты.

1. **ИСПОЛЬЗУЕМЫ ПРОГРАММНЫЕ ТЕХНОЛОГИИ**

Для создания приложения я использовал Microsoft Visual Studio, язык программирования С#, интерфейс программирования приложений Windows Forms, а также SQL Server Management для создания локальной базы данных.

**Microsoft Visual Studio**

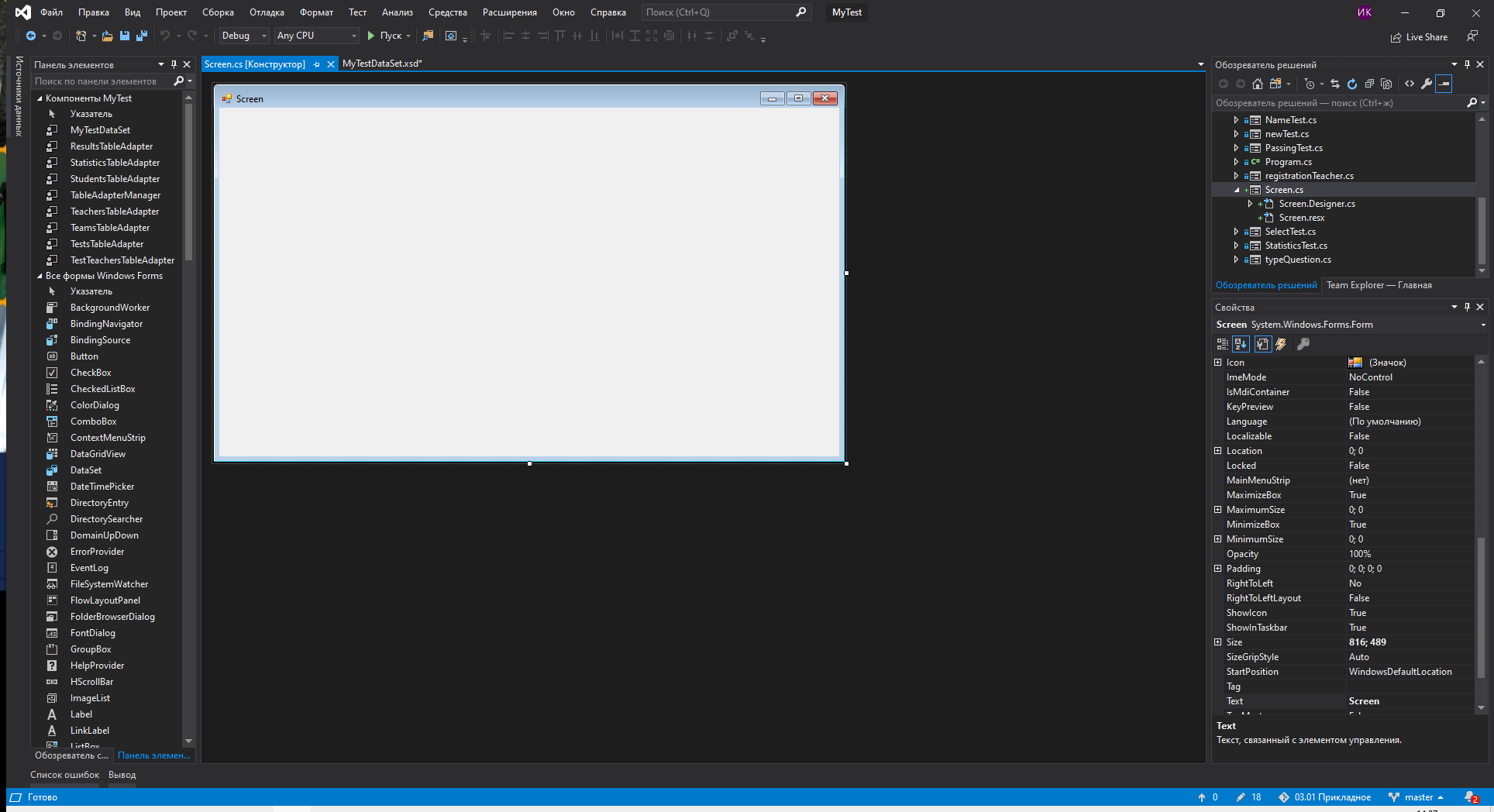
Инструментальная среда разработки, включающая в себя интегрированную среду разработки, редактор исходного кода, встроенный отладчик. Многие другие инструменты возможно получить благодаря подключению плагинов – сторонних расширений. Был выбран по причине наличия интеграции с Unity 3d, поддержки языка программирования С#, наличие возможности написание собственных Unit-тестов. Также это наиболее изученная разработчиком платформа, что позволяет быстрее разрабатывать приложения не тратя время на изучения нового материала (Рисунок 1).

Рисунок 1 - Microsoft Visual Studio

**C#**

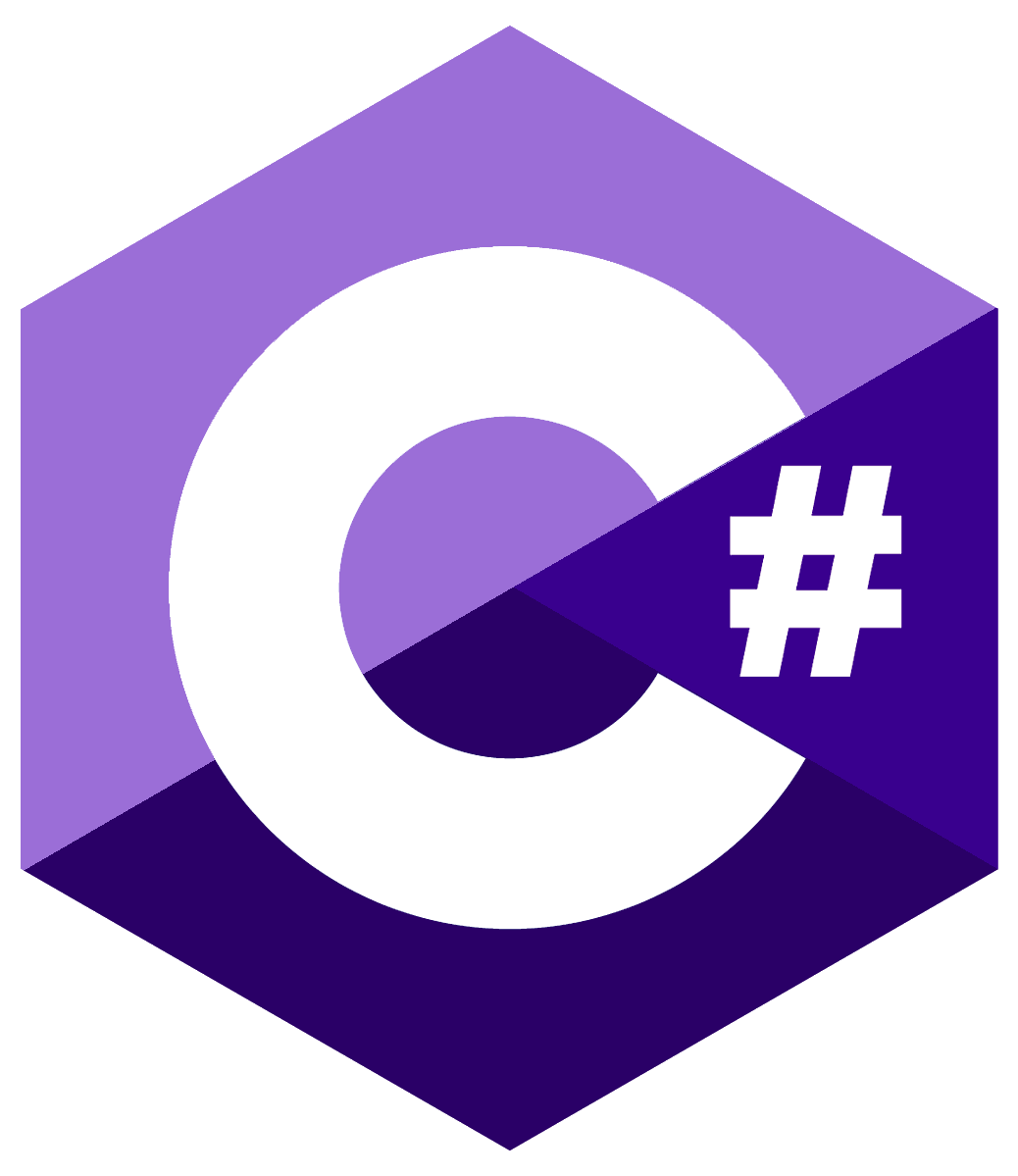
Объектно-ориентированный язык программирования. Разработан в 1998 — 2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework (Рисунок 2).

Рисунок 2 - С# лого

**Windows Forms**

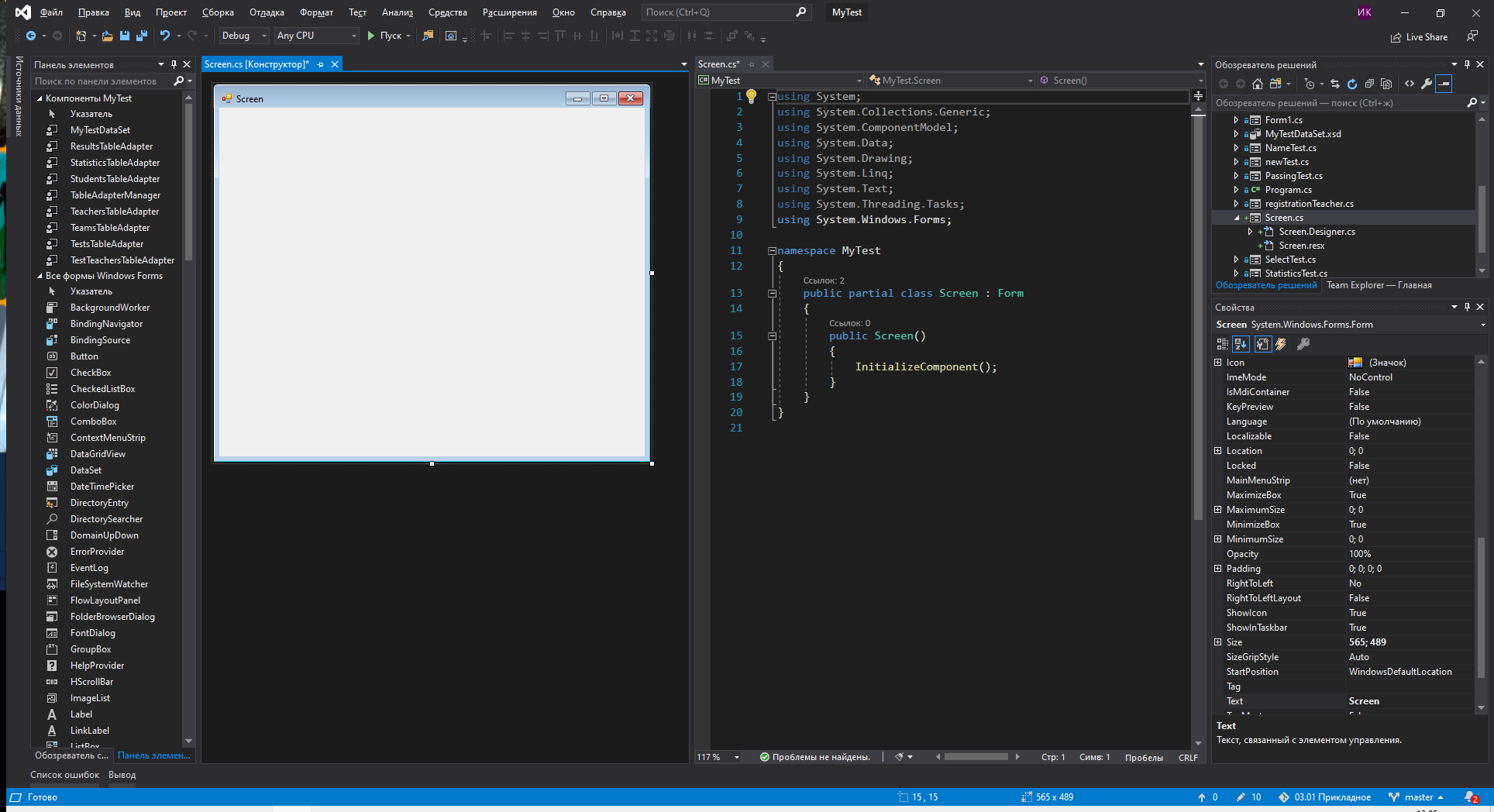
Windows Forms — интерфейс программирования приложений, отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework. Данный интерфейс упрощает доступ к элементам интерфейса Microsoft Windows за счет создания обёртки для существующего Win32 API в управляемом коде. (Рисунок 3).

Рисунок 3 Windows Forms

Рисунок 3 - Windows Forms

**SQL Server Management Studio**

Утилита из Microsoft SQL Server 2005 и более поздних версий для конфигурирования, управления и администрирования всех компонентов Microsoft SQL Server. Утилита включает скриптовый редактор и графическую программу, которая работает с объектами и настройками сервера (Рисунок 4).

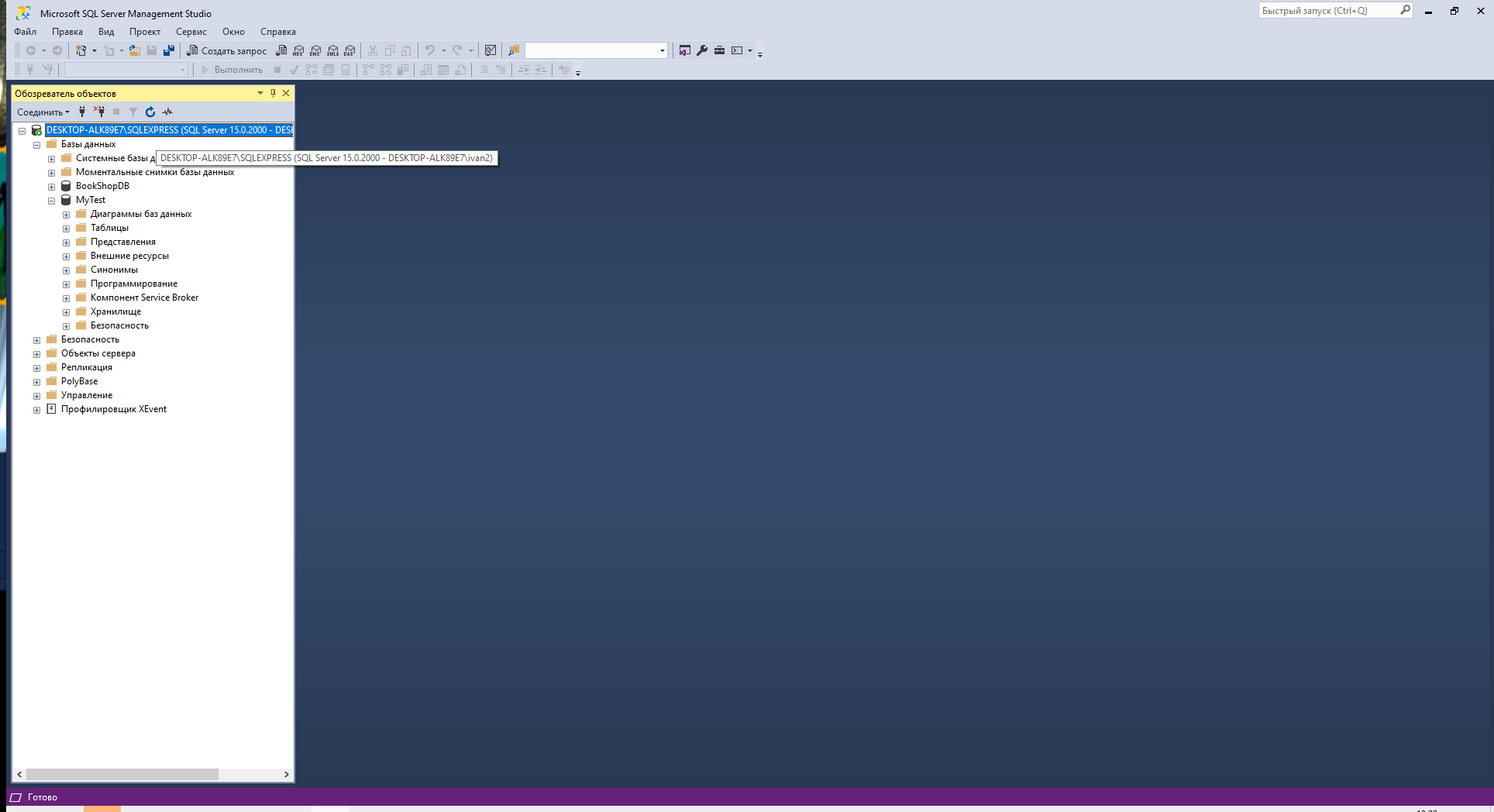


Рисунок 4 SQL Server Managment

1. **ОПИСАНИЕ ФУНКЦИОНАЛЬНО-ЛОГИЧЕСКОЙ СТРУКТУРЫ ПРОГРАММЫ**
   1. **Структурная схема приложения**

Самая «главная» форма – форма авторизации, из нее можно попасть на форму регистрации, а также учителя и студента (в случаи правильного ввода логина и пароля), служит для авторизации. Форма регистрации аналогично переносит на формы учителя и студента (в случаи успешной регистрации), нужна для регистрации пользователей. Форма студента «конечная», то есть из нее можно перейти только обратно на авторизацию, ее функция исключительно прохождение теста. Форма же учителя нужна только для создания тестов, и ради перехода на форму статистики. В свою очередь форма статистики представляет информацию о прохождении тестов (принадлежащим учителю, который зашел) предоставляя фильтры для отбора нужных результатов тестирования студентов, также «конечная».

* 1. **Диаграмма вариантов использования**

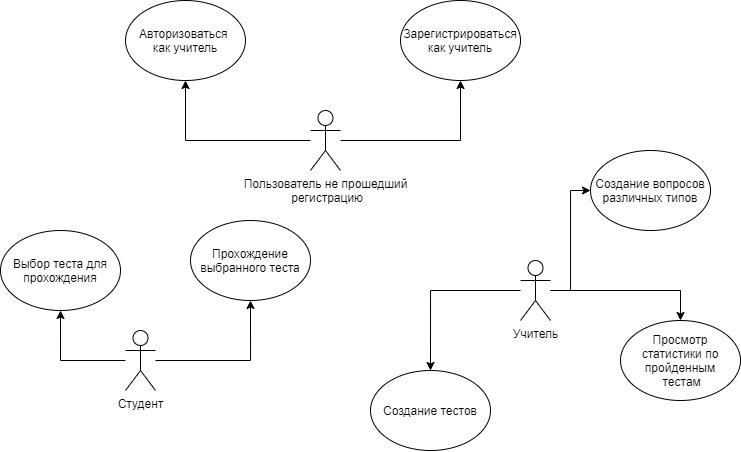
В этом разделе приведена UML диаграмма, показывающая спектр возможностей разных типов пользователей (Рисунок 5). Стоит отметить, что возможности еще невошедшего или незарегистрированного пользователя, присутствуют как у студентов, так и у учителей.

Рисунок 5 UML диаграмма

1. **ЛОГИЧЕСКАЯ МОДЕЛЬ ДАННЫХ**

Для создание такого приложения необходима база данных со следующей структурой (Рисунок 6). Как можно понять из ER диаграммы, вопросы и ответы на тесты хранятся не в таблицах БД, т.к. для них специально создается свои собственные таблицы, для каждого теста отдельная таблица.

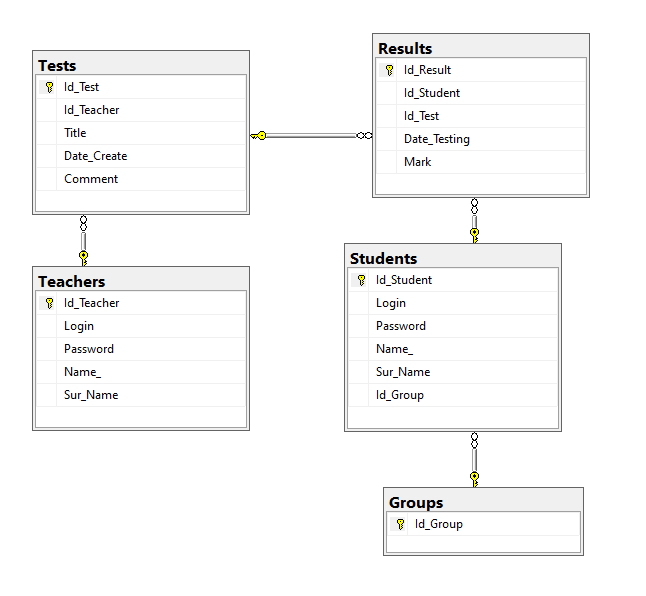


Рисунок 6 ER диаграмм базы данных «My\_Tests»

**Таблица** **Teams**

Эта таблица созданна исключительно для запоминании групп в системе. (Рисунок 7).

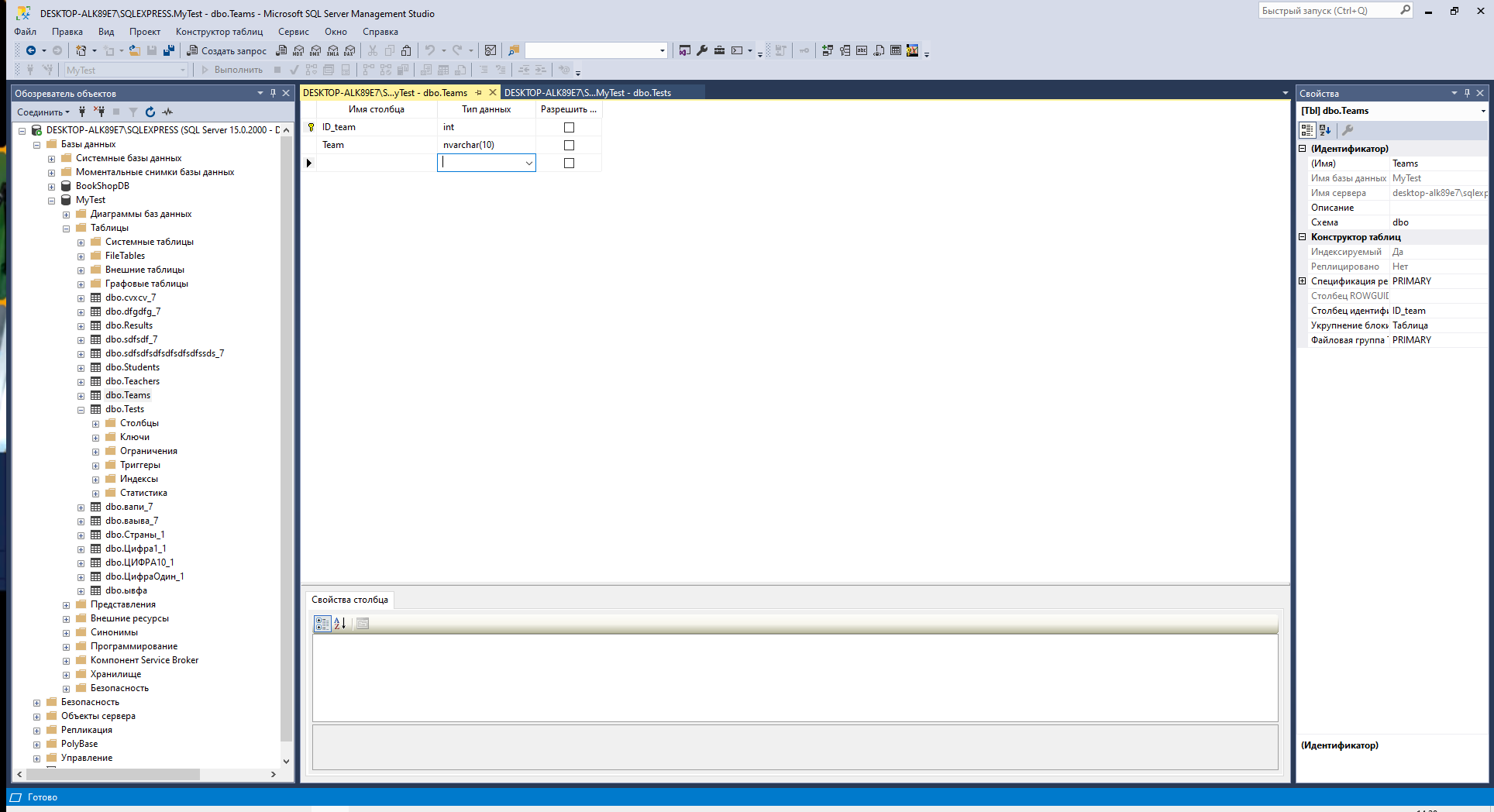


Рисунок 7 Структура таблицы Teams

**Таблица Results**

В этой таблице представлены результаты тестирования студентов (Рисунок 8).

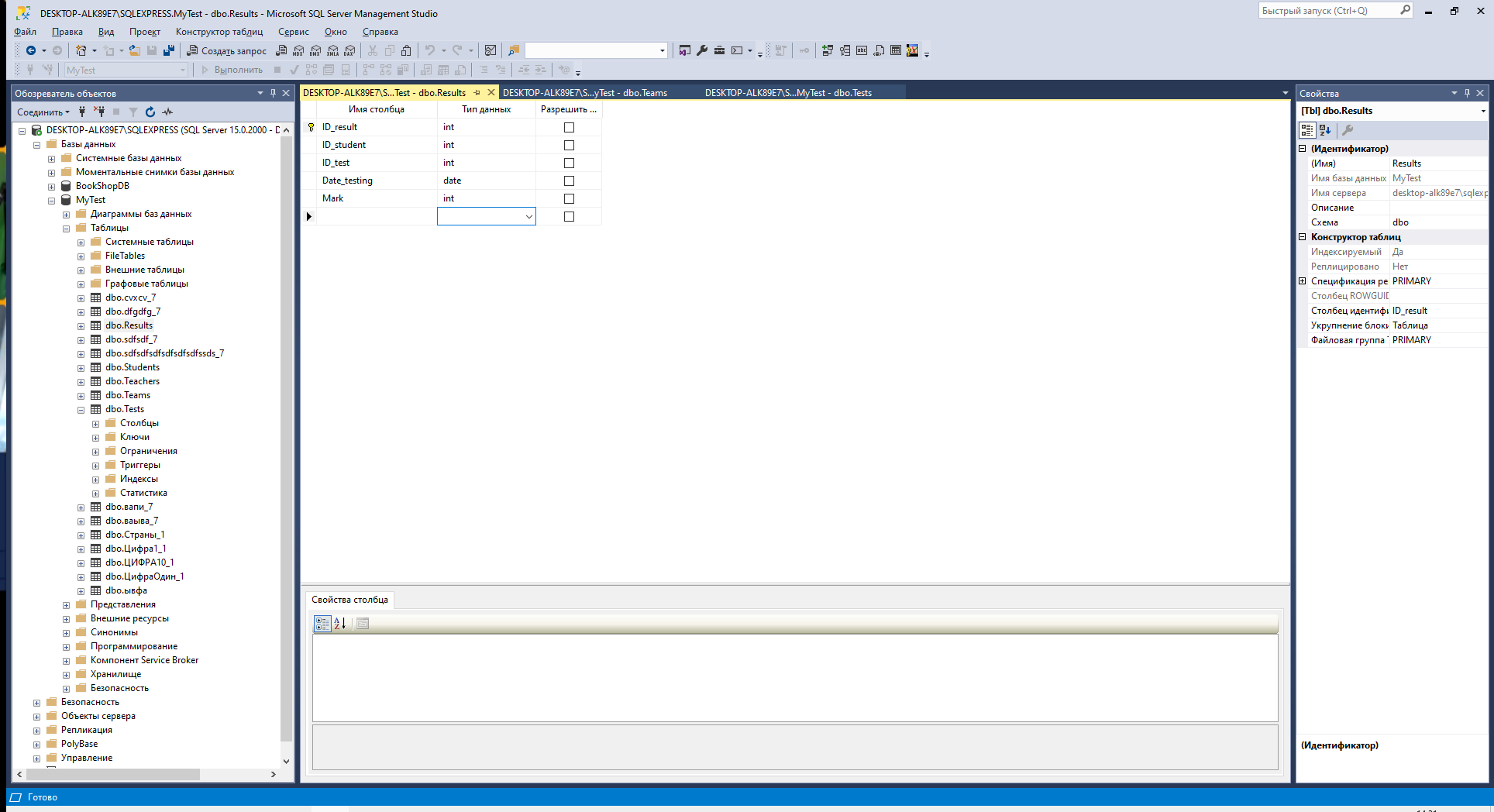


Рисунок 8 Структура таблицы Results

**Таблица Students**

Здесь приведена иформация о зарегистрированных студентах (Рисунок 9).

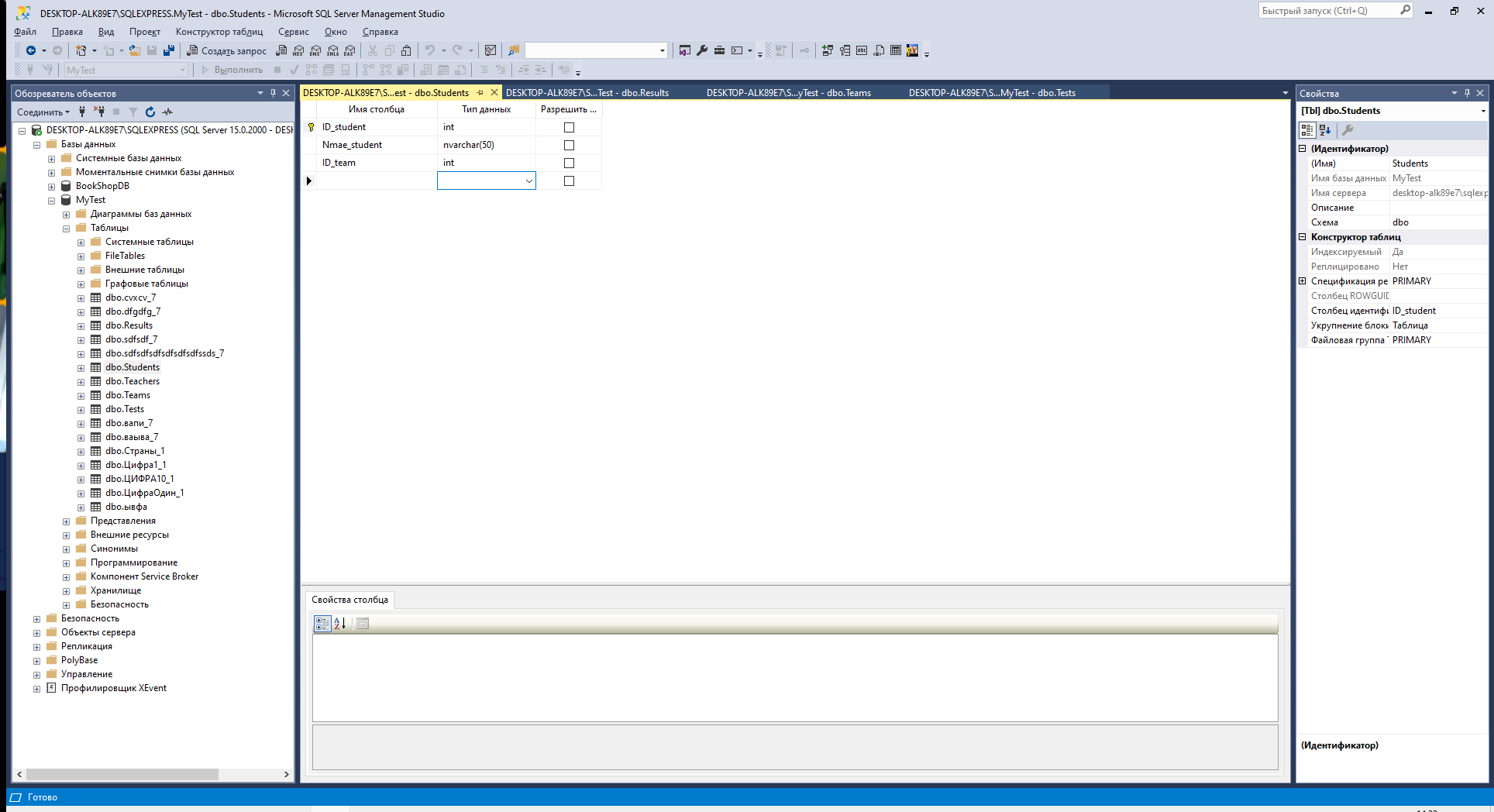


Рисунок 9 Структура таблицы Students

**Таблица Teachers**

Аналогично предыдущий таблице хранит информацию о пользователях, но уже только о зарегистрировавшихся учителях (Рисунок 10).

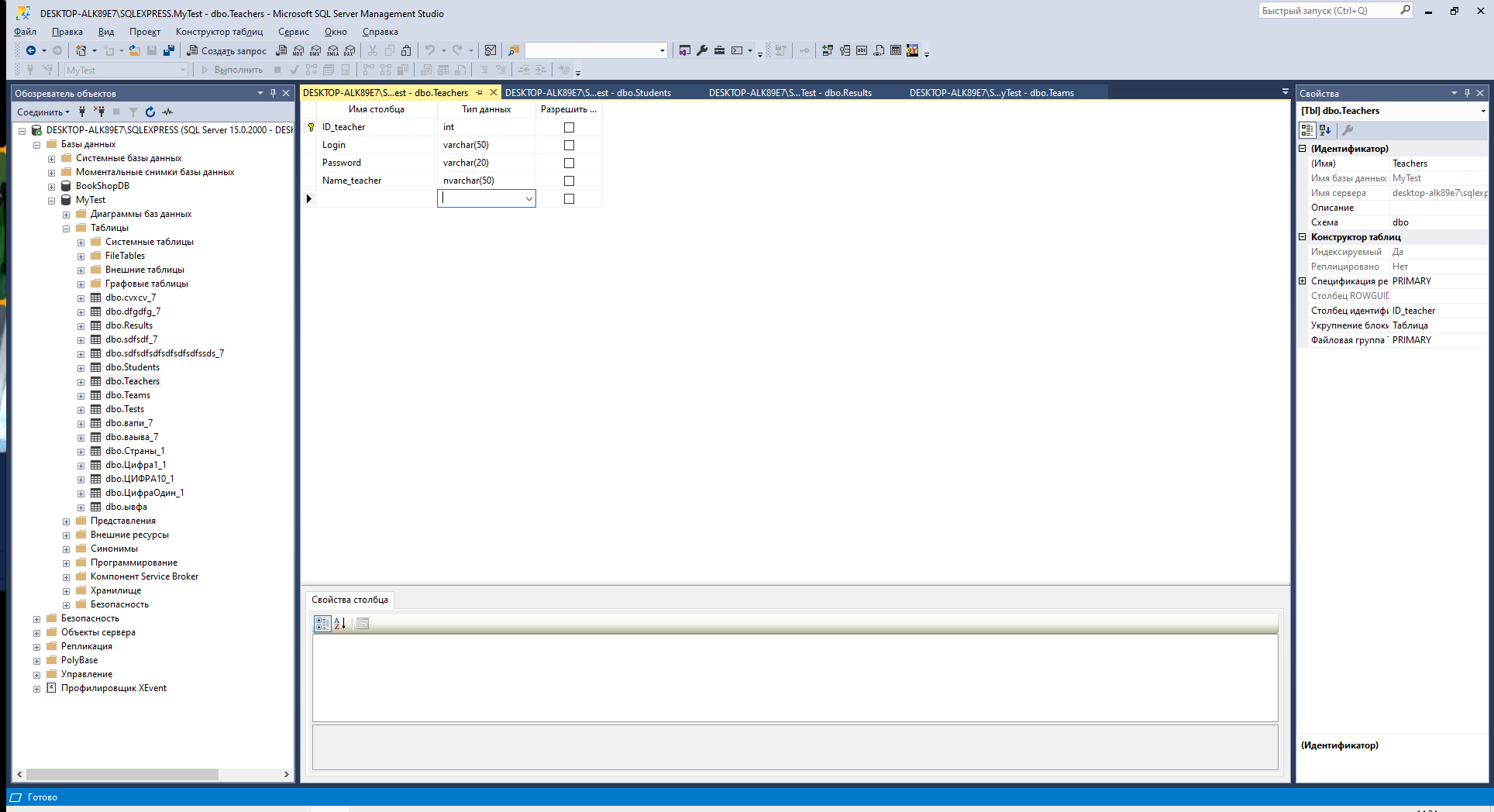


Рисунок 10 Структура таблицы Teachers

**Таблица Tests**

Эта же таблица хранит информацию о созданных тестах (Рисунок 11). Отмечу что вопросы и ответы на тесты хранятся в отдельной таблице, и связывает их айди теста здесь и в названии таблицы с вопросами.

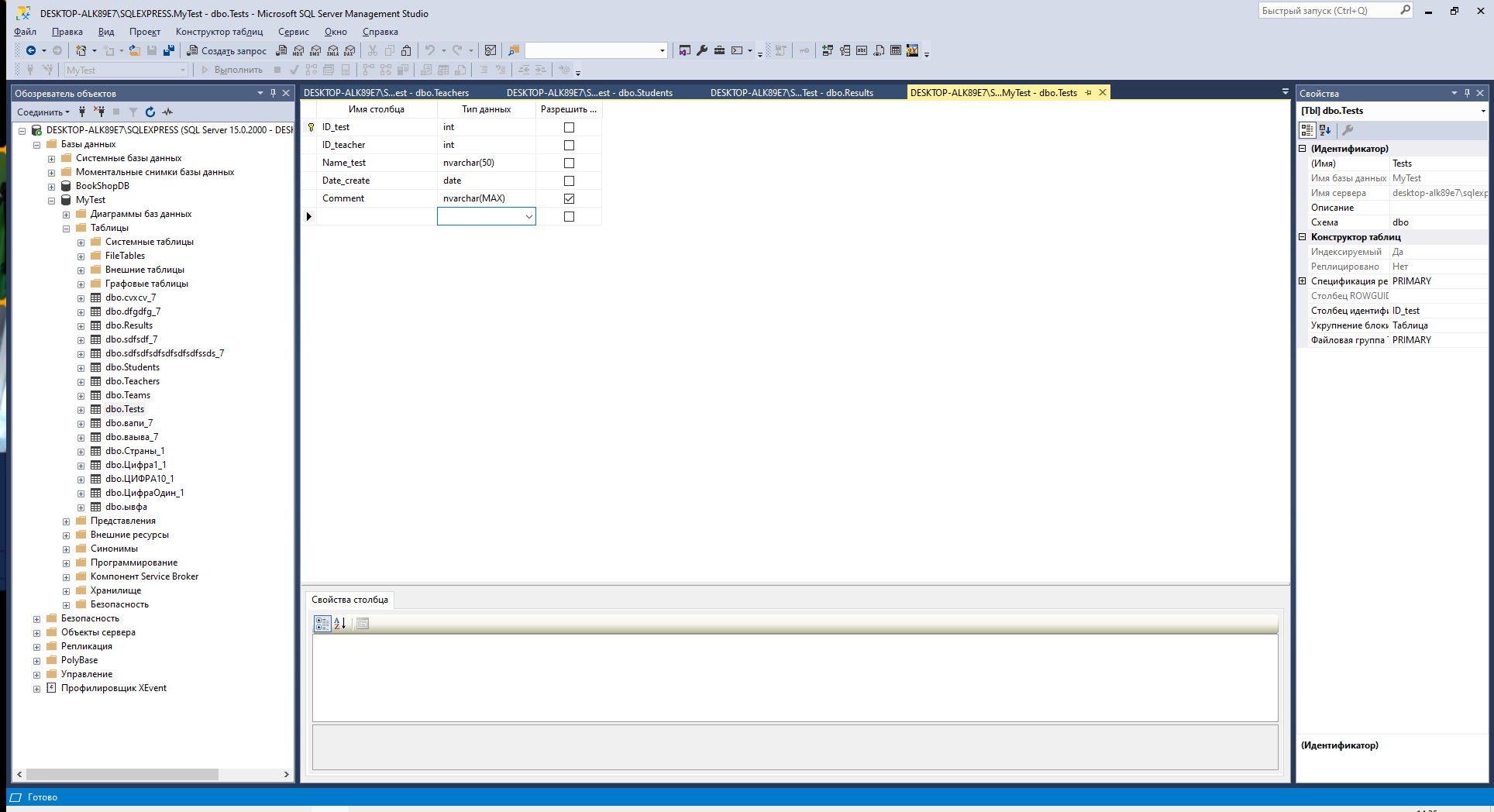


Рисунок 11 Структура таблицы Tests

1. **ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЯ**

**Меню приложения**

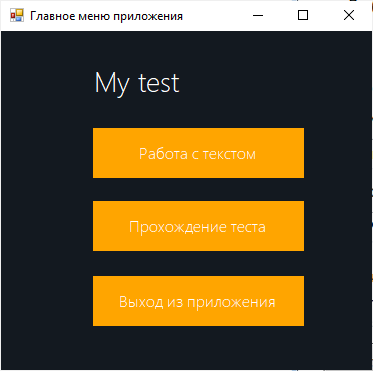
При запуске приложения сначала откроется «главная», из которой мы можем попасть на все остальные формы (Рисунок 12). Далее можно перйти на форму аторизации преподователя или на форму авторизации студента и выйти из приложения.

Рисунок 12 Первая форма

**Форма авторизации**

Форма авторизации, из которой мы можем попасть на все остальные формы (Рисунок 13). Данная форма реагирует на то введены ли все данные в поля, а также на некорректный ввод логина и пароля. В случае ошибки входа приложение оповестит об этом (Рисунок 14).

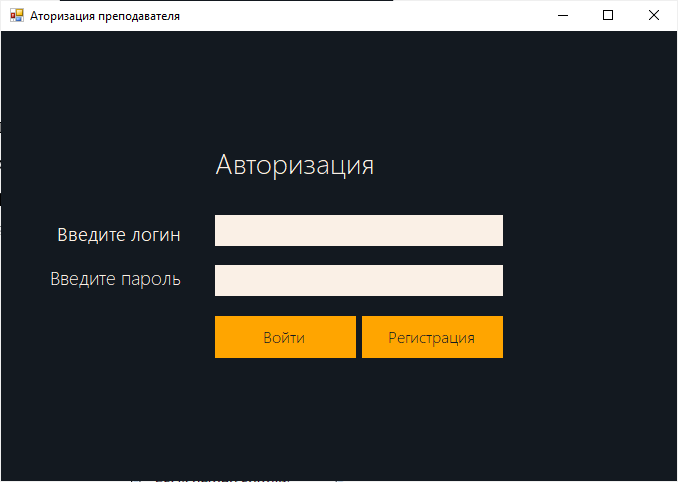
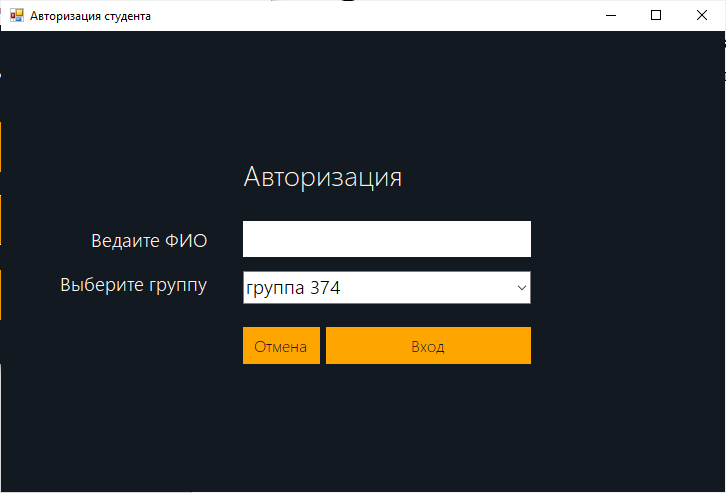


Рисунок 13 Авторизация преподователя и студента

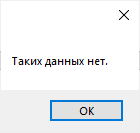


Рисунок 14 Реакция на ввод не корректного логина и пароля

**Форма регистрации**

На данную форму можно попасть в случаи нажатия кнопки «Регистрация» на форме авторизации преподавателя. (Рисунок 15). Также если пользователь: не ввел все данные, использовал уже существующий логин или ввел разные значения в поля «Пароль» и «Потв. пароль» приложение среагирует на это (Рисунок 16, 17, 18). В случае же успешной регистрации всплывает сообщение о успешной регистрации (Рисунок 21).

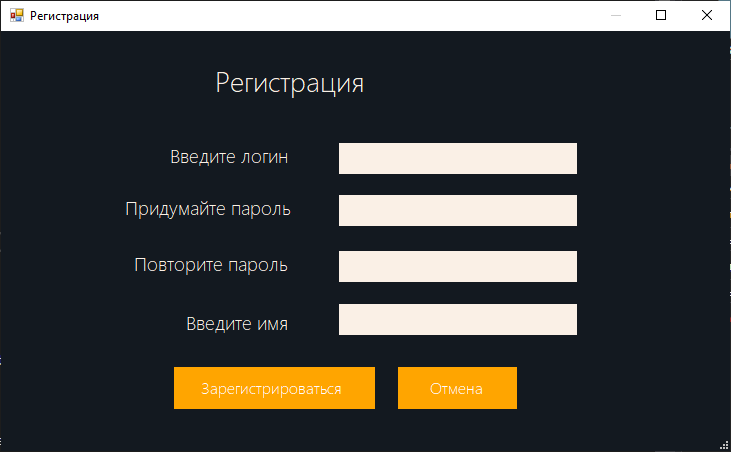


Рисунок 15 Форма регистрации

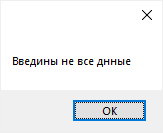
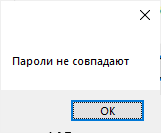


Рисунок 16 Реакция на ввод разных поролей

Рисунок 17 Реакция на ввод не всех данных

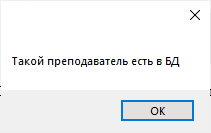


Рисунок 18 Реакци на регистрацию пользователя

с уже существующим логином в БД

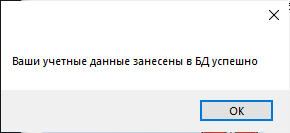


Рисунок 19 Реакция на успешную регистрацию

**Форма учителя**

Если на форме авторизации вы ввели правильные учетные данные и зашли как учитель, создали новый тест и пару вопросов к нему, перед вами будет форма похожего вида (Рисунок 20). Здесь кнопка «Принять изменения» сохраняет изменения в выбранном вопросе, а снизу слева находиться строка состояния, указывающая какое сегодня число, а также кол-во вопросов с индексом текущего. Также слева сверху присутсвует меню позволяющее создавать и сохранять тест в БД, открывать форму статистики, и выходить из текущего «аккаунта» (Рисунок 21).

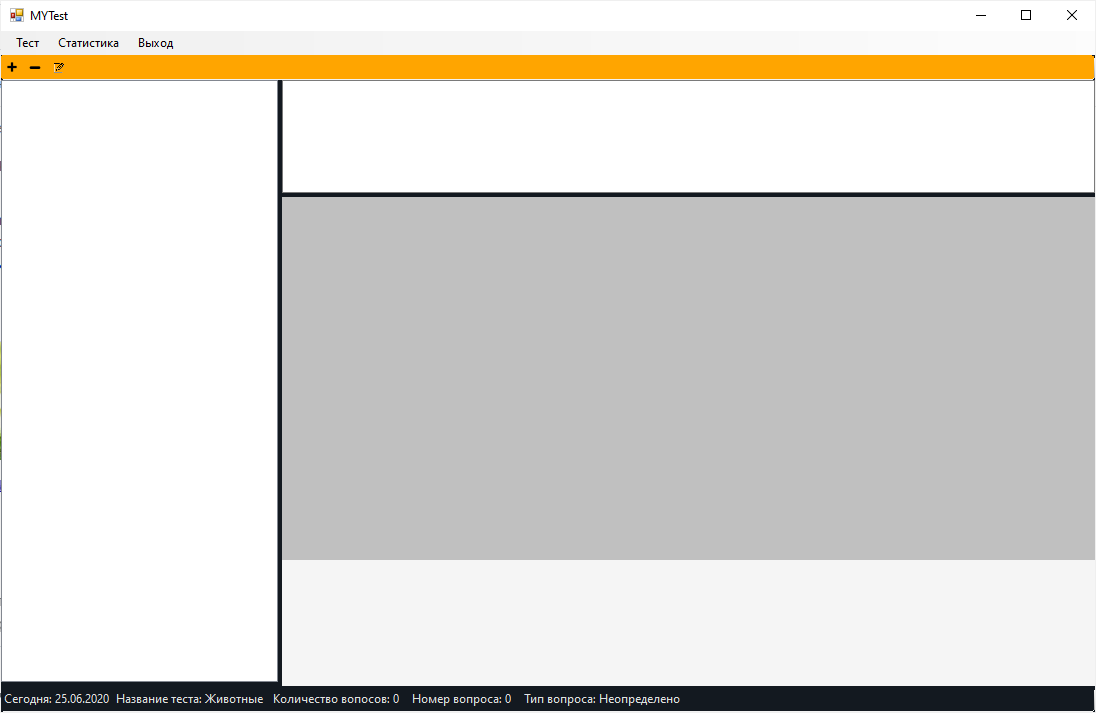
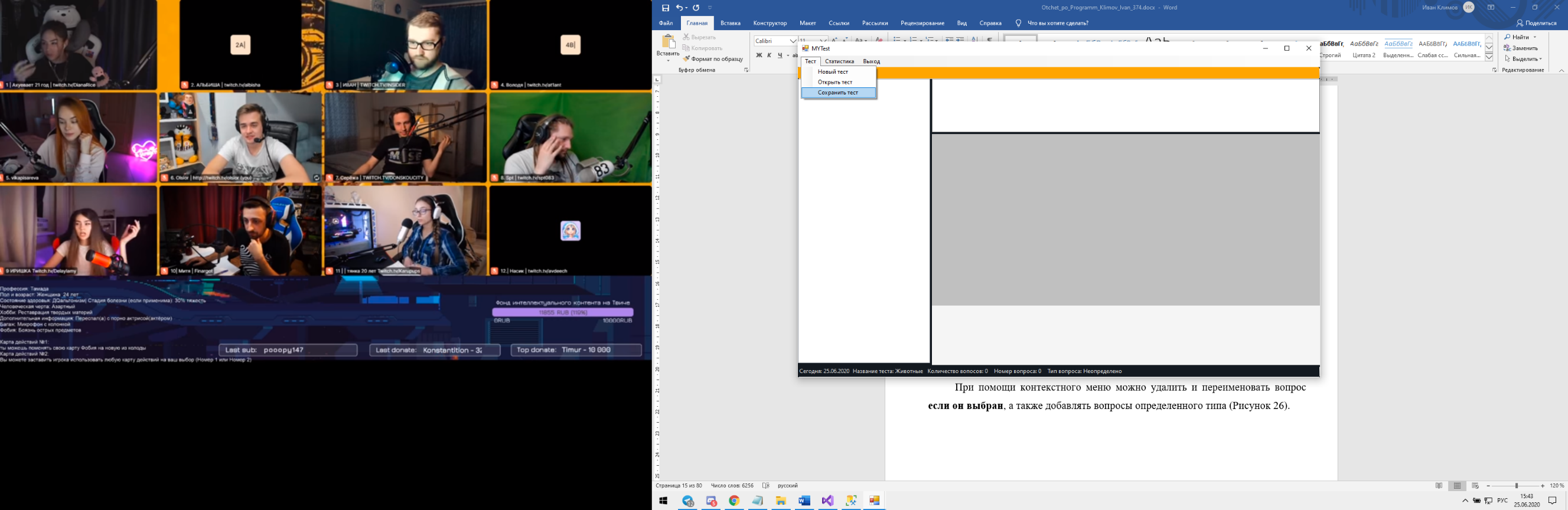


Рисунок 20 Форма учителя

Рисунок 21 Доп. меню

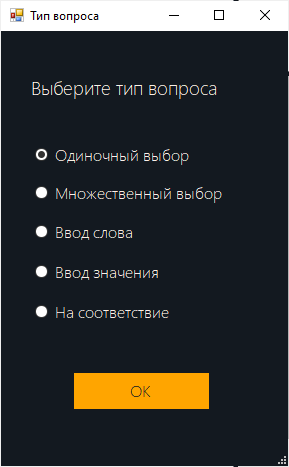
Также можно добавлять вопросы определенного типа (Рисунок 22).

Рисунок 22 Добавление вопроса определенного типа

Дальше в завизимости от созданного вопроса, ему можно будет добавить вопрос и ответ (Рисунок 23).

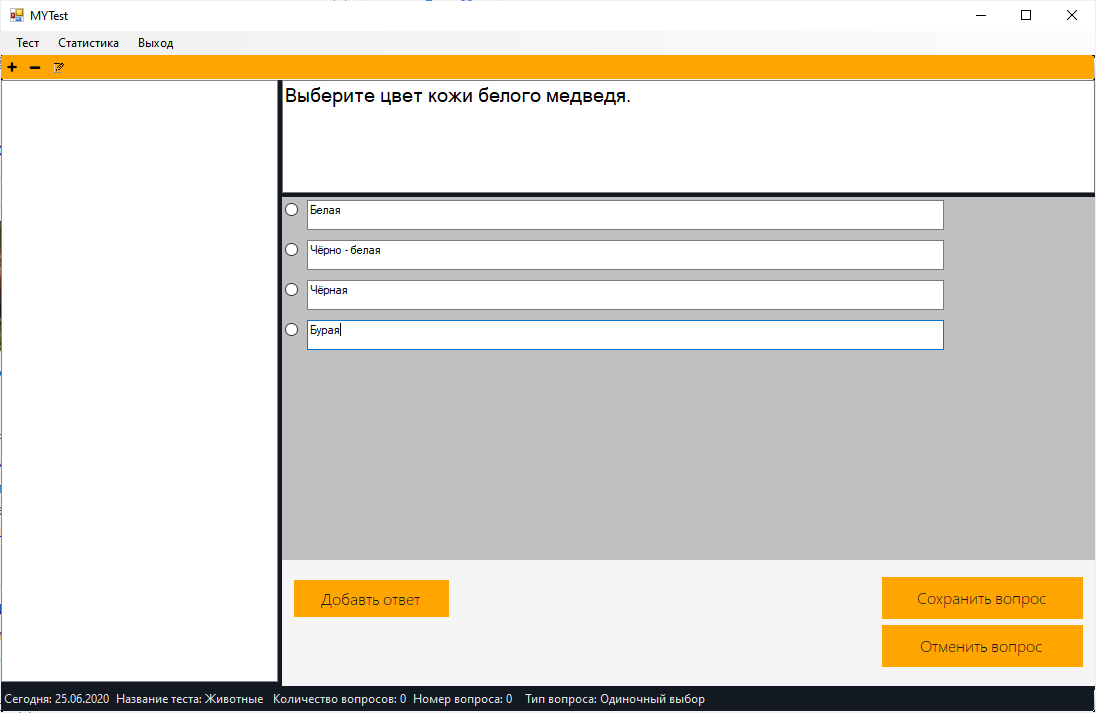


Рисунок 23 Добавление вопроса и ответов

Также в случаи создание учителем двух тестов с одинаковыми названиями приложение предложит поменять название при сохранении. В другом случаи приложение отреагирует на сохранение теста следующим образом (Рисунок 31).

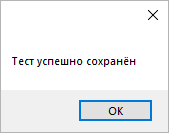
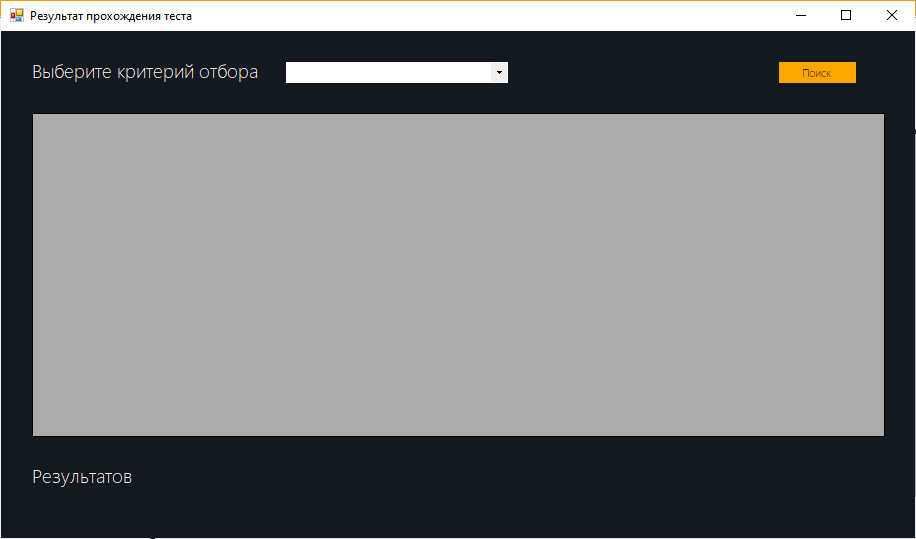


Рисунок 24 Реакция на успешное сохранение теста

**Форма статистики**

Данная форма октрывается во вкладке формы учителя с таким же названием, служащая исключительно для просмотра данных по тестированию и их фильтрации (Рисунок 25). Сверху присутсвует меню, осуществляющее фильтрацию на основе заданных параметров, можно искать по имени и фамилии (ищет вхождение символов, а не конкретной строки), по группе, тесту, оценке (спомощь. ComboBox`ов пример на рисунке), а также по дате (можно выбирать диапозон) (Рисунок 26).



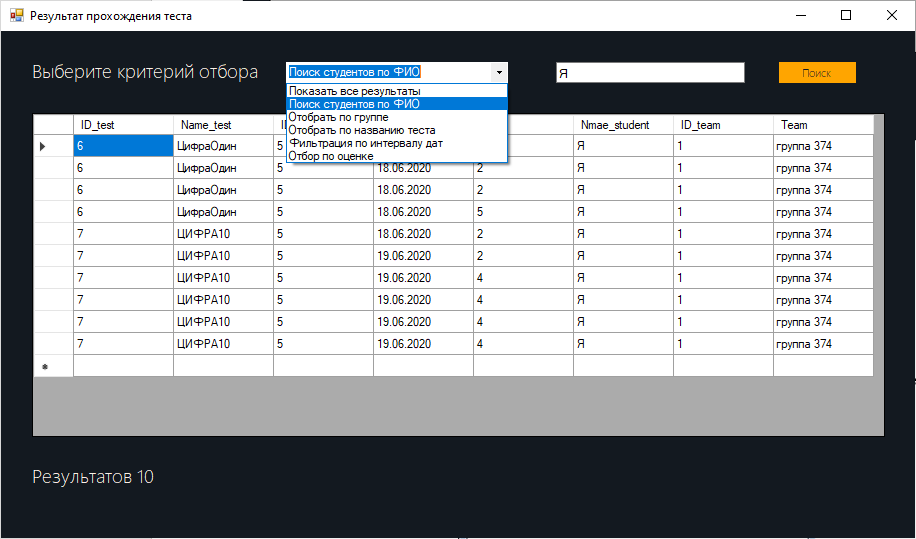


Рисунок 25 Форма статистики

Рисунок 26 Меню фильтрации

**Форма статистики**

После ввода именни студента и выбора группы, выведеться оконо с тестами (Рисунок 27).

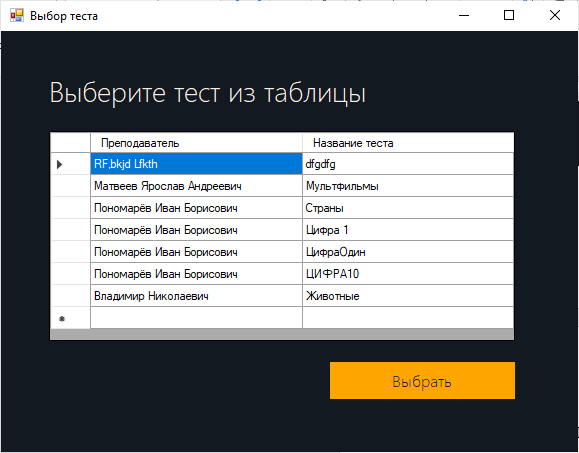


Рисунок 27 Если тесты есть, откроется выбор тестов

После выбора интересующего студента теста, откроется форма его прохождения, примерно следующего содержания (Рисунок 28). На ней присутсвуют кнопки навигации позволяющие выбрать вопрос, а также кнопка по нажатию которой происходит удаление вопроса из очереди и его проверка на правильность. Также есть полоска состояния, указывающая сколько вопросов отвечено из их количества.

Рисунок 28 Форма студента после выбора теста

После нажатия «Закончить тест» приложение подсчитает кол-во правильных вопрос из всех вопросов, и поставит соотвествующую оценку (Рисунок 29).

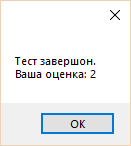


Рисунок 29 Реакция на окончание теста

После чего приложение занесет результаты тестирования в БД.

**ЗАКЛЮЧЕНИЕ**

В заключении стоит отметить, что на выходе получилось приложение соответсвующее требованиям из постановки задач. А то есть авторизация и регистрация как учителей, так и студентов присутствует. Студенты могут выбирать и проходить нужные им тесты, а учителя могут эти тесты создавать и затем просматривать статистику по их прохождению. Следовательно основное задание было выполнено.

**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Сайт по созданию диаграмм: <https://app.diagrams.net/>
2. Сайт с заданием, а также откуда был взять шаблон: <https://sites.google.com/site/iamatisik/studentam/ucebnaa-praktika>

**ПРИЛОЖЕНИЕ. ИСХОДНЫЙ ТЕКСТ ПРОГРАММЫ**

**MainWindow.xaml.cs**

**authorizationTeacher.cs**

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 MyTest

{

public partial class authorizationTeacher : Form

{

public authorizationTeacher()

{

InitializeComponent();

}

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

{

string log, pass;

log = textBoxLogin.Text;

pass = textBoxPassword.Text;

var rec = this.teachersTableAdapter1.GetData();

var filter = rec.Where(p => p.Login == log && p.Password == pass);

if (filter.Count() == 0)

MessageBox.Show("Таких данных нет.");

else {

ClassTotal.idTeacher = filter.ElementAt(0).ID\_teacher;

ClassTotal.nameTeacher = filter.ElementAt(0).Name\_teacher;

//MessageBox.Show("Вы успешно авторизовались как преподаватель.");

newTest newTest = new newTest();

this.Hide();

newTest.ShowDialog();

}

}

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

{

registrationTeacher registrationTeacher = new registrationTeacher();

this.Hide();

registrationTeacher.ShowDialog();

}

}

}

**Exit.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MyTest

{

public class ClassQuestion

{

public string textQuestion; //Текст вопроса

public string typeQuestion; //Тип вопроса

public string answers; //Строка ответов

public ClassQuestion(string textQ, string typeQ, string ans)

{

textQuestion = textQ;

typeQuestion = typeQ;

answers = ans;

}

}

}

**Form1.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace MyTest

{

public partial class FirstForm : Form

{

public FirstForm()

{

InitializeComponent();

}

public void close() {

this.close();

}

private void label1\_Click(object sender, EventArgs e)

{

}

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

{

authorizationTeacher authorizationTeacher = new authorizationTeacher();

authorizationTeacher.Show();

}

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

{

ClassTotal.connectionString = Properties.Settings.Default.MyTestConnectionString;

}

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

{

Exit Exit = new Exit();

Exit.ShowDialog();

}

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

{

authorizationStudents authorizationStudents = new authorizationStudents();

authorizationStudents.Show();

}

}

}

**NameTest.cs**

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 MyTest

{

public partial class NameTest : Form

{

private newTest m\_parent;

public NameTest(newTest newTest)

{

InitializeComponent();

m\_parent = newTest;

}

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

{

this.Close();

}

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

{

if (textBoxNameTest.Text == " " || textBoxNameTest.Text == "")

{

MessageBox.Show("Введите название");

}

else {

ClassTotal.nameTest = textBoxNameTest.Text;

this.Close();

//newTest.Owner = this;

//newTest.toolStripLabelNameTest.Text =$"Название теста {textBoxNameTest.Text}";

//newTest.ShowDialog();

}

}

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

{

m\_parent.Test();

}

}

}\

**PassingTest.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace MyTest

{

public partial class PassingTest : Form

{

public List <ClassQuestion> listQuestion = new List <ClassQuestion>();

public int countAnswers;

public string[] answerMas;

public string trueAnswer;

string typeQuestion;

string answers;

int zazorTop = 25,

comHieght = 20,

zazor = 30;

public int trueNumber,

currentNumber;

public int totalNumber;

public int mark;

private void Show\_Question()

{

ClassQuestion question = listQuestion[currentNumber - 1]; //Объект-текущий вопрос теста

this.textBoxTextQ.Text = question.textQuestion; //Получение информации о вопросе

typeQuestion = question.typeQuestion; //Тип вопроса

answers = question.answers;

answerMas = answers.Split(';'); //Деление ответов на части

countAnswers = answerMas.Length - 1; //Количество ответов

splitContainer2.Panel2.Controls.Clear(); //Готовим панель для динамических компонентов

trueAnswer = ""; //Строка правильных ответов

//Отображаем динамические компоненты в зависимости от типа вопроса

switch (typeQuestion)

{

case "Одиночный выбор":

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

{

trueAnswer += answerMas[i][0]; //Первый символ в ответе хранит правильность: + или –

//Динамические компоненты ответа зависят от типа вопроса

ClassTotal.RadioButtonDynamic(splitContainer2.Panel2, 20, zazorTop + i \* comHieght + i \* zazor, 30, comHieght);

ClassTotal.LabelDynamic(splitContainer2.Panel2, 60, zazorTop + i \* comHieght + i \* zazor, splitContainer2.Panel2.Width - 70, comHieght, answerMas[i].Remove(0, 1));

}

break;

case "Множественный выбор":

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

{

trueAnswer += answerMas[i][0];

ClassTotal.CheckBoxDynamic(splitContainer2.Panel2, 3, 3 + (i \* 40), 17, 20);

ClassTotal.LabelDynamic(splitContainer2.Panel2, 25, 3 + (i \* 40), 637, 20, answerMas[i].Remove(0, 1));

}

break;

case "Ввод значения":

trueAnswer = answerMas[0];

ClassTotal.TextBoxDynamic(splitContainer2.Panel2, 3, 3, 330, 30);

break;

case "Ввод слова":

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

{

trueAnswer += answerMas[i];

ClassTotal.TextBoxDynamic(splitContainer2.Panel2, 3, 3 + (i \* 40), 330, 30);

}

break;

case "На соответствие":

string[] first = new string[countAnswers];

string[] second = new string[countAnswers];

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

{

string[] an = answerMas[i].Split('&');

first[i] = an[0];

second[i] = an[1];

}

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

{

trueAnswer += second[i];

ClassTotal.LabelDynamic(splitContainer2.Panel2, 17, 3 + (i \* 40), 300, 20, first[i]);

ClassTotal.ComboBoxDynamic(splitContainer2.Panel2, 348, 3 + (i \* 40), 300, 20, second);

}

break;

}

}

public PassingTest()

{

InitializeComponent();

}

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

{

this.progressBar.Value++;

//Обработка ответов на текущий вопрос теста

string s = ""; //Строка из + и – по ответам

bool right = false; //Флаг правильности ответа

//Анализ ответа на текущий вопрос в зависимости от типа вопроса

switch (typeQuestion) //Тип вопроса

{

case "Одиночный выбор":

for (int i = 0; i < splitContainer2.Panel2.Controls.Count; i++) //Количество компонент в панели

{

if (splitContainer2.Panel2.Controls[i] is RadioButton) //Если компонент – кнопка

{

RadioButton rb = splitContainer2.Panel2.Controls[i] as RadioButton; //Получить кнопку

if (rb.Checked) s += '+'; //Создание строки с ответами

else s += '-';

}

}

if (s == trueAnswer) right = true; //Сравнить полученные ответы с заданными

break;

case "Множественный выбор":

for (int i = 0; i < splitContainer2.Panel2.Controls.Count; i++) {

if (splitContainer2.Panel2.Controls[i] is CheckBox) //Если компонент – кнопка

{

CheckBox chb = splitContainer2.Panel2.Controls[i] as CheckBox; //Получить кнопку

if (chb.Checked) s += '+'; //Создание строки с ответами

else s += '-';

}

}

if (s == trueAnswer) right = true;

break;

case "Ввод слова":

for (int i = 0; i < splitContainer2.Panel2.Controls.Count; i++)

{

if (splitContainer2.Panel2.Controls[i] is TextBox) //Если компонент – кнопка

{

TextBox tb = splitContainer2.Panel2.Controls[i] as TextBox; //Получить кнопку

s = tb.Text;

}

}

if (s == trueAnswer)

right = true;

break;

case "Ввод значения":

for (int i = 0; i < splitContainer2.Panel2.Controls.Count; i++)

{

if (splitContainer2.Panel2.Controls[i] is TextBox) //Если компонент – кнопка

{

TextBox tb = splitContainer2.Panel2.Controls[i] as TextBox; //Получить кнопку

s = tb.Text;

}

}

if (s == trueAnswer)

right = true;

break;

case "На соответствие":

//MessageBox.Show(trueAnswer);

for (int i = 0; i < splitContainer2.Panel2.Controls.Count; i++)

{

if (splitContainer2.Panel2.Controls[i] is ComboBox)

{

ComboBox tb2 = this.splitContainer2.Panel2.Controls[i] as ComboBox;

s += tb2.Text;

}

}

if (s == trueAnswer)

right = true;

break;

}

//Анализ результата ответа натекущий вопрос

if (right)

{

MessageBox.Show(" Вопрос отвечен верно ");

trueNumber++;

}

else

{

MessageBox.Show(" Вопрос отвечен не верно");

}

//Проверка завершения теста

if (currentNumber < totalNumber) //Еще не все вопросы пройдены

{

currentNumber++; //Номер следующего вопроса

Show\_Question(); //Отобразить компоненты от типа вопроса

}else //Пройдены все вопросы теста

{

//Расчет оценки в зависимости от числа правильно отвеченных вопросов

double proc = trueNumber \* 100.0 / totalNumber;

if (proc > 80) mark = 5;

else

if (proc > 65) mark = 4;

else

if (proc > 50) mark = 3;

else mark = 2;

MessageBox.Show($" Тест завершон. \r\n Ваша оценка: {mark}");

//Занесение данных о результатах прохождения теста в таблицу Results БД

this.resultsTableAdapter1.Insert(ClassTotal.idStud, ClassTotal.idTest, DateTime.Now, mark);

this.Close();

}

}

private void splitContainer1\_Panel2\_Paint(object sender, PaintEventArgs e)

{

}

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

{

listQuestion.Clear();

string query;

string textQuestion,

typeQuestion,

answers;

mark = 0;

SqlConnection connection = new SqlConnection(ClassTotal.connectionString);

connection.Open();

try

{

//Выборка всех данных из динамической таблицы с вопросами теста

query = "SELECT \* FROM " + ClassTotal.nameTest + "\_" + ClassTotal.idTeacher;

SqlCommand command = new SqlCommand(query, connection);

SqlDataReader dataReader = command.ExecuteReader(); //Все записи

//Перенести все вопросы теста в список

while (dataReader.Read()) //Перебрать все записи по одной

{

//Получение полей текущей записи

textQuestion = dataReader["Text\_question"].ToString();

typeQuestion = dataReader["Type\_question"].ToString();

answers = dataReader["Answers"].ToString();

//Создание объекта-вопроса конструктором

ClassQuestion question = new ClassQuestion(textQuestion, typeQuestion, answers);

listQuestion.Add(question); //Добавление вопроса в список

}

dataReader.Close();

connection.Close();

totalNumber = listQuestion.Count; //Общее количество вопросов

this.progressBar.Maximum = totalNumber;

currentNumber = 1; //Текущий номер вопроса

trueNumber = 0; //Количество правильных ответов

Show\_Question();

}

catch

{

MessageBox.Show("Данный тест был удален.\r\nОбратитесь к преподавателю.");

connection.Close();

this.Close();

return;

}

}

}

}

**newTest.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace MyTest

{

public partial class newTest : Form

{

//public string toolStripLabelNameTest

//{

// get { return toolStripLabelNameTest.Text; }

// set { toolStripLabelNameTest.Text = value; }

//}

public bool save = true;

public List<ClassQuestion> listQuestion = new List<ClassQuestion>();

//SqlConnection Conn = new SqlConnection("Data Source=DESKTOP-ALK89E7\\SQLEXPRESS;Initial Catalog=MyTest;Integrated Security=True");

public static int quantitu\_question = 0, number\_question = 0;

public void Test()

{

toolStripLabelNameTest.Text = $"Название теста: {ClassTotal.nameTest}";

toolStripButtonPluses.Visible = true;

toolStripButtonMinuses.Visible = true;

toolStripButtonСheck.Visible = true;

//buttonAddAnswer.Visible = true;

//buttonSaveQuestion.Visible = true;

}

public void Type(string tupe)

{

toolStripLabelTypeQuestion.Text = $"Тип вопроса: {tupe}";

toolStripLabelQuantityQuestions.Text = "Количество вопросов: " + ClassTotal.QuantityQuestions.ToString();

}

public newTest()

{

InitializeComponent();

}

private void новыйТестToolStripMenuItem1\_Click(object sender, EventArgs e)

{

NameTest NameTest = new NameTest(this);

NameTest.ShowDialog();

}

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

{

Exit Exit = new Exit();

Exit.ShowDialog();

}

private void toolStrip1\_ItemClicked(object sender, ToolStripItemClickedEventArgs e)

{

}

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

{

typeQuestion typeQuestion = new typeQuestion(this);

typeQuestion.ShowDialog();

}

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

{

ClassTotal.type = null;

toolStripLabelTypeQuestion.Text = "Тип вопроса: Неопределено";

splitContainer3.Panel1.Controls.Clear();

textBoxQuestion.Text = "";

}

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

{

ClassTotal.ans = "";

int temp\_i = 0;

//Ответы вопроса

bool right = false;

if(textBoxQuestion.Text != "")

{

switch (ClassTotal.type)

{

case "Одиночный выбор":

for (int i = 0; i < this.splitContainer3.Panel1.Controls.Count; i += 2)

{

RadioButton rb = this.splitContainer3.Panel1.Controls[i] as RadioButton;

TextBox tb = this.splitContainer3.Panel1.Controls[i + 1] as TextBox;

if (tb.Text != "") //Сохранять только заполненные вопросы

{

temp\_i++;

if (rb.Checked) //Кнопка включена – вопрос выбран

{

ClassTotal.ans += "+"+tb.Text + ";"; //Учитывается +

right = true; //Одна кнопка выбрана

}

else ClassTotal.ans += "-"+tb.Text + ";"; //Выключена – «-«

}

else

{

MessageBox.Show("Заполнины не все поля");

}

}

break;

case "Множественный выбор":

for (int i = 0; i < this.splitContainer3.Panel1.Controls.Count; i += 2)

{

CheckBox rb = this.splitContainer3.Panel1.Controls[i] as CheckBox;

TextBox tb = this.splitContainer3.Panel1.Controls[i + 1] as TextBox;

if (tb.Text != "") //Сохранять только заполненные вопросы

{

temp\_i++;

if (rb.Checked) //Кнопка включена – вопрос выбран

{

ClassTotal.ans += "+" + tb.Text + ";"; //Учитывается +

right = true; //Одна кнопка выбрана

}

else ClassTotal.ans += "-" + tb.Text + ";"; //Выключена – «-«

}

else {

MessageBox.Show("Заполнины не все поля");

}

}

break;

case "Ввод слова":

for (int i = 0; i < this.splitContainer3.Panel1.Controls.Count; i ++)

{

TextBox tb = this.splitContainer3.Panel1.Controls[i] as TextBox;

if (tb.Text != "") //Сохранять только заполненные вопросы

{

temp\_i = 2;

ClassTotal.ans += tb.Text + ";";

right = true;

}

else

{

MessageBox.Show("Заполнины не все поля");

}

}

break;

case "Ввод значения":

for (int i = 0; i < this.splitContainer3.Panel1.Controls.Count; i++)

{

TextBox tb = this.splitContainer3.Panel1.Controls[i] as TextBox;

if (tb.Text != "") //Сохранять только заполненные вопросы

{

temp\_i = 2;

ClassTotal.ans += tb.Text;

right = true;

}

else

{

MessageBox.Show("Заполнины не все поля");

}

}

break;

case "На соответствие":

for (int i = 0; i < this.splitContainer3.Panel1.Controls.Count; i+=2)

{

TextBox tb = this.splitContainer3.Panel1.Controls[i] as TextBox;

TextBox tb1 = this.splitContainer3.Panel1.Controls[i+1] as TextBox;

if (tb.Text != "" && tb1.Text != "")

{

temp\_i++;

ClassTotal.ans += tb.Text + "&" + tb1.Text +";";

right = true;

}

else

{

MessageBox.Show("Заполнины не все поля");

}

}

break;

}

if(right == true && temp\_i > 1)

{

ClassQuestion question = new ClassQuestion(textBoxQuestion.Text, ClassTotal.type, ClassTotal.ans);

listQuestion.Add(question);

ClassTotal.QuantityQuestions++;

//toolStripLabelNumberQuestion.Text = "Номер вопроса: " + ClassTotal.NumberQuestion;

toolStripLabelQuantityQuestions.Text = "Количество вопросов: " + ClassTotal.QuantityQuestions;

MessageBox.Show(ClassTotal.ans);

listBox.Items.Add(textBoxQuestion.Text);

this.splitContainer3.Panel1.Controls.Clear();

this.textBoxQuestion.Text = "";

ClassTotal.type = null;

}

else

{

MessageBox.Show("Заполнины не все поля");

}

}

else{

MessageBox.Show("Заполните вопрос");

}

}

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

{

save = true;

if (listBox.Items.Count < 1)

{

save = false;

MessageBox.Show("Тест пустой");

}

MessageBox.Show(ClassTotal.nameTest + "\_" + ClassTotal.idTeacher);

string query = "SELECT name FROM sys.tables";

SqlConnection connection = new SqlConnection(ClassTotal.connectionString);

connection.Open();

SqlCommand command = new SqlCommand(query, connection);

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

if (reader[0].ToString() == ClassTotal.nameTest + "\_" + ClassTotal.idTeacher)

{

save = false;

MessageBox.Show("Тест с таким названием уже создан.\nПереименуйте тест");

NameTest ntf = new NameTest(this);

ntf.Show();

}

}

reader.Close();

connection.Close();

if (save == true)

{

query = "CREATE TABLE [" + ClassTotal.nameTest + "\_" + ClassTotal.idTeacher +

"] ([ID\_question] INT IDENTITY(1,1) NOT NULL PRIMARY KEY," +

"[Text\_question] nvarchar(100) NOT NULL," +

"[Type\_question] nvarchar(100) NOT NULL, " +

"[Answers] nvarchar(100) NOT NULL)";

connection.Open();

command = new SqlCommand(query, connection);

command.ExecuteNonQuery();

foreach (ClassQuestion item in listQuestion)

{

string query1 = "INSERT INTO [" + ClassTotal.nameTest + "\_" + ClassTotal.idTeacher +

"] ([Text\_question], [Type\_question], [Answers]) " +

"VALUES ('" + item.textQuestion + "', '" + item.typeQuestion + "', '" + item.answers + "')";

SqlCommand command1 = new SqlCommand(query1, connection);

command1.ExecuteNonQuery();

MessageBox.Show("Тест успешно сохранён");

}

DateTime dt = DateTime.Today;

query = "INSERT INTO [Tests] ([ID\_teacher], [Name\_test], [Date\_create]) VALUES ('" + ClassTotal.idTeacher + "','" + ClassTotal.nameTest + "','" +

dt.ToShortDateString() + "')";

command = new SqlCommand(query, connection);

command.ExecuteNonQuery();

MessageBox.Show("Тест успешно записан в бд");

connection.Close();

}

}

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

{

ClassTotal.temp\_interval++;

switch (ClassTotal.type)

{

case "Одиночный выбор":

ClassTotal.RadioButtonDynamic(splitContainer3.Panel1, 3, 120 + (ClassTotal.temp\_interval \* 40), 17, 20);

ClassTotal.TextBoxDynamic(splitContainer3.Panel1, 25, 120 + (ClassTotal.temp\_interval \* 40), 637, 20);

break;

case "Множественный выбор":

ClassTotal.CheckBoxDynamic(splitContainer3.Panel1, 3, 120 + (ClassTotal.temp\_interval \* 40), 17, 20);

ClassTotal.TextBoxDynamic(splitContainer3.Panel1, 25, 120 + (ClassTotal.temp\_interval \* 40), 637, 20);

break;

case "Ввод слова":

ClassTotal.TextBoxDynamic(splitContainer3.Panel1, 3, 3 + (ClassTotal.temp\_interval \* 40), 330, 20);

break;

case "На соответствие":

ClassTotal.TextBoxDynamic(splitContainer3.Panel1, 17, 120 + (ClassTotal.temp\_interval \* 40), 300, 20);

ClassTotal.TextBoxDynamic(splitContainer3.Panel1, 348, 120 + (ClassTotal.temp\_interval \* 40), 300, 20);

break;

}

}

private void listBox\_SelectedIndexChanged(object sender, EventArgs e)

{

toolStripLabelNumberQuestion.Text = "Номер вопроса: " + (listBox.SelectedIndex + 1);

}

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

{

StatisticsTest StatisticsTest = new StatisticsTest();

StatisticsTest.Show();

}

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

{

toolStripLabelQuantityQuestions.Text += "0 ";

toolStripLabelNumberQuestion.Text += "0 ";

toolStripLabelTypeQuestion.Text += "Неопределено";

toolStripLabelNameTest.Text += "Неопределено";

toolStripButtonPluses.Visible = false;

toolStripButtonMinuses.Visible = false;

toolStripButtonСheck.Visible = false;

buttonAddAnswer.Visible = false;

buttonSaveQuestion.Visible = false;

buttonСancelQuestion.Visible = false;

DateTime dt = DateTime.Today;

toolStripLabelDate.Text += dt.ToShortDateString();

}

}

}

**registrationTeacher.cs**

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 MyTest

{

public partial class registrationTeacher : Form

{

public registrationTeacher()

{

InitializeComponent();

}

private void label2\_Click(object sender, EventArgs e)

{

}

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

{

}

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

{

authorizationTeacher authorizationTeacher = new authorizationTeacher();

this.Hide();

authorizationTeacher.Show();

}

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

{

string log, pass, pass2, name;

log = textBoxNewLogin.Text;

pass = textBoxNewPassword.Text;

pass2 = textBoxRepeatPassword.Text;

name = textBoxNewName.Text;

var rec = this.teachersTableAdapter1.GetData();

var filter = rec.Where(p => p.Login == log && p.Password == pass);

if (filter.Count() > 0)

MessageBox.Show("Такой преподаватель есть в БД");

else {

try

{

if (log == "" || pass == "" || name == "" || pass2 == "")

{

MessageBox.Show("Введины не все днные");

}

else if (pass == pass2)

{

this.teachersTableAdapter1.Insert(log, pass, name);

var temp = this.teachersTableAdapter1.GetData(); //Все записи после добавления

int maxID = temp.Last().ID\_teacher; //ID добавленной записи

ClassTotal.idTeacher = maxID;

ClassTotal.nameTeacher = name;

MessageBox.Show("Ваши учетные данные занесены в БД успешно");

newTest newTest = new newTest();

this.Hide();

newTest.ShowDialog();

MessageBox.Show("Переход в форму работы с текстом");

}

else {

MessageBox.Show("Пароли не совпадают");

}

}

catch

{

MessageBox.Show("Ошибка при внесении Ваших учетных данных.");

}

}

}

}

}

**SelectTest.cs**

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 MyTest

{

public partial class SelectTest : Form

{

public SelectTest()

{

InitializeComponent();

}

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

{

this.dataGridView.DataSource = this.testTeachersTableAdapter1.GetData();

this.dataGridView.Columns["ID\_test"].Visible = false;

this.dataGridView.Columns["ID\_teacher"].Visible = false;

this.dataGridView.Columns["Name\_test"].HeaderText = " Название теста";

this.dataGridView.Columns["Name\_teacher"].HeaderText = " Преподаватель ";

this.dataGridView.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

}

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

{

ClassTotal.idTest = (int)this.dataGridView.CurrentRow.Cells["ID\_test"].Value;

ClassTotal.nameTest = (string)this.dataGridView.CurrentRow.Cells["Name\_test"].Value;

ClassTotal.idTeacher = (int)this.dataGridView.CurrentRow.Cells["ID\_teacher"].Value;

PassingTest PassingTest = new PassingTest();

this.Close();

PassingTest.Show();

}

}

}

**StatisticsTest.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace MyTest

{

public partial class StatisticsTest : Form

{

public MyTestDataSet.StatisticsDataTable records;

public StatisticsTest()

{

InitializeComponent();

}

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

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "myTestDataSet1.Tests". При необходимости она может быть перемещена или удалена.

this.testsTableAdapter.Fill(this.myTestDataSet1.Tests);

// TODO: данная строка кода позволяет загрузить данные в таблицу "myTestDataSet1.Teams". При необходимости она может быть перемещена или удалена.

this.teamsTableAdapter.Fill(this.myTestDataSet1.Teams);

records = statisticsTableAdapter1.GetDataIDTeacher(ClassTotal.idTeacher);

this.textBoxNameStudent.Visible = false;

this.comboBoxGroup.Visible = false;

this.comboBoxNameTest.Visible = false;

this.dateTimePickerStart.Visible = false;

this.dateTimePickerFinish.Visible = false;

this.numericUpDownMark.Visible = false;

}

private void comboBox\_SelectedIndexChanged(object sender, EventArgs e)

{

switch (comboBox.Text)

{

case "Показать все результаты":

this.textBoxNameStudent.Visible = false;

this.comboBoxGroup.Visible = false;

this.comboBoxNameTest.Visible = false;

this.dateTimePickerStart.Visible = false;

this.dateTimePickerFinish.Visible = false;

this.numericUpDownMark.Visible = false;

break;

case "Поиск студентов по ФИО":

this.textBoxNameStudent.Visible = true;

this.comboBoxGroup.Visible = false;

this.comboBoxNameTest.Visible = false;

this.dateTimePickerStart.Visible = false;

this.dateTimePickerFinish.Visible = false;

this.numericUpDownMark.Visible = false;

break;

case "Отобрать по группе":

this.comboBoxGroup.Visible = true;

this.textBoxNameStudent.Visible = false;

this.comboBoxNameTest.Visible = false;

this.dateTimePickerStart.Visible = false;

this.dateTimePickerFinish.Visible = false;

this.numericUpDownMark.Visible = false;

break;

case "Отобрать по названию теста":

this.comboBoxNameTest.Visible = true;

this.textBoxNameStudent.Visible = false;

this.comboBoxGroup.Visible = false;

this.dateTimePickerStart.Visible = false;

this.dateTimePickerFinish.Visible = false;

this.numericUpDownMark.Visible = false;

break;

case "Фильтрация по интервалу дат":

this.dateTimePickerStart.Visible = true;

this.dateTimePickerFinish.Visible = true;

this.comboBoxNameTest.Visible = false;

this.textBoxNameStudent.Visible = false;

this.comboBoxGroup.Visible = false;

this.numericUpDownMark.Visible = false;

break;

case "Отбор по оценке":

this.numericUpDownMark.Visible = true;

this.dateTimePickerStart.Visible = false;

this.dateTimePickerFinish.Visible = false;

this.comboBoxNameTest.Visible = false;

this.textBoxNameStudent.Visible = false;

this.comboBoxGroup.Visible = false;

break;

}

}

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

{

switch (comboBox.Text)

{

case "Показать все результаты":

var recAll = this.statisticsTableAdapter1.GetDataIDTeacher(ClassTotal.idTeacher);

this.dataGridView.DataSource = recAll;

this.labelResults.Text = $"Результатов {recAll.Count()}";

break;

case "Поиск студентов по ФИО":

var recFIO = this.statisticsTableAdapter1.GetDataByFIO(ClassTotal.idTeacher, textBoxNameStudent.Text);

this.dataGridView.DataSource = recFIO;

this.labelResults.Text = $"Результатов {recFIO.Count()}";

break;

case "Отобрать по группе":

var recGroup = this.statisticsTableAdapter1.GetDataByNameGroup(ClassTotal.idTeacher,comboBoxGroup.Text);

this.dataGridView.DataSource = recGroup;

this.labelResults.Text = $"Результатов {recGroup.Count()}";

break;

case "Отобрать по названию теста":

var recNameTest = this.statisticsTableAdapter1.GetDataByNameTest(ClassTotal.idTeacher, comboBoxNameTest.Text);

this.dataGridView.DataSource = recNameTest;

this.labelResults.Text = $"Результатов {recNameTest.Count()}";

break;

case "Фильтрация по интервалу дат":

string Start = Convert.ToString(dateTimePickerStart.Value.Date),

Finifsh = Convert.ToString(dateTimePickerFinish.Value.Date);

var recStartFirsr = this.statisticsTableAdapter1.GetDataByStartFinish(ClassTotal.idTeacher, Start, Finifsh);

this.dataGridView.DataSource = recStartFirsr;

this.labelResults.Text = $"Результатов {recStartFirsr.Count()}";

break;

case "Отбор по оценке":

int mark = Convert.ToInt32(numericUpDownMark.Value);

var recMark = this.statisticsTableAdapter1.GetDataByMark(ClassTotal.idTeacher, mark);

this.dataGridView.DataSource = recMark;

this.labelResults.Text = $"Результатов {recMark.Count()}";

break;

}

}

}

}

**typeQuestion.cs**

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 MyTest

{

public partial class typeQuestion : Form

{

private bool temp = false;

private newTest m\_parent;

public typeQuestion(newTest newTest)

{

InitializeComponent();

m\_parent = newTest;

}

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

{

if (temp == true)

{

newTest newTest = new newTest();

ClassTotal.temp\_interval = 0;

m\_parent.splitContainer3.Panel1.Controls.Clear();

m\_parent.textBoxQuestion.Text = "";

m\_parent.buttonAddAnswer.Visible = true;

m\_parent.buttonSaveQuestion.Visible = true;

m\_parent.buttonСancelQuestion.Visible = true;

switch (ClassTotal.type)

{

case "Одиночный выбор":

for (int i = 0; i < 4; i++)

{

ClassTotal.RadioButtonDynamic(m\_parent.splitContainer3.Panel1, 3, 3 + (i \* 40), 17, 20);

ClassTotal.TextBoxDynamic(m\_parent.splitContainer3.Panel1, 25, 3 + (i \* 40), 637, 30);

}

break;

case "Множественный выбор":

//m\_parent.splitContainer3.Panel1.Controls.Clear();

for (int i = 0; i < 4; i++)

{

ClassTotal.CheckBoxDynamic(m\_parent.splitContainer3.Panel1, 3, 3 + (i \* 40), 17, 30);

ClassTotal.TextBoxDynamic(m\_parent.splitContainer3.Panel1, 25, 3 + (i \* 40), 637, 30);

}

break;

case "Ввод слова":

ClassTotal.TextBoxDynamic(m\_parent.splitContainer3.Panel1, 3, 3, 330, 30);

m\_parent.buttonAddAnswer.Visible = false;

break;

case "Ввод значения":

ClassTotal.TextBoxDynamic(m\_parent.splitContainer3.Panel1, 3, 3, 330, 30);

m\_parent.buttonAddAnswer.Visible = false;

break;

case "На соответствие":

for (int i = 0; i < 4; i++)

{

ClassTotal.TextBoxDynamic(m\_parent.splitContainer3.Panel1, 17, 3 + (i \* 40), 300, 30);

ClassTotal.TextBoxDynamic(m\_parent.splitContainer3.Panel1, 348, 3 + (i \* 40), 300, 30);

}

break;

}

this.Close();

}

}

private void radioButton1\_CheckedChanged(object sender, EventArgs e)

{

ClassTotal.type = "Одиночный выбор";

m\_parent.Type("Одиночный выбор");

temp = true;

}

private void radioButton2\_CheckedChanged(object sender, EventArgs e)

{

ClassTotal.type = "Множественный выбор";

m\_parent.Type("Множественный выбор");

temp = true;

}

private void radioButton3\_CheckedChanged(object sender, EventArgs e)

{

ClassTotal.type = "Ввод слова";

m\_parent.Type("Ввод слова");

temp = true;

}

private void radioButton4\_CheckedChanged(object sender, EventArgs e)

{

ClassTotal.type = "Ввод значения";

m\_parent.Type("Ввод значения");

temp = true;

}

private void radioButton5\_CheckedChanged(object sender, EventArgs e)

{

ClassTotal.type = "На соответствие";

m\_parent.Type("На соответствие");

temp = true;

}

}

}