Министерство просвещения Приднестровской Молдавской республики

ГОУ СПО «Тираспольский техникум информатики и права»

**КУРСОВАЯ РАБОТА**

по дисциплине: «Прикладное программирование»

на тему: «Конструктор экзаменационных билетов»

**Выполнил обучающийся**:

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

**Специальность** 2.09.02.03

Программирование в

Компьютерных системах

**Руководитель:**

Преподаватель отделения

информационных дисциплин

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

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

(оценка)

«\_\_\_\_» «­­\_\_\_\_\_\_\_\_\_\_\_\_» «­\_\_\_\_\_\_»

(дата, подпись руководителя)

Тирасполь, 2022

**Оглавление**

[**Введение** 3](#_Toc99531424)

[**Глава 1. Теоретическая часть разработки программного обеспечения** 4](#_Toc99531425)

[1.1. Постановка задачи 4](#_Toc99531426)

[1.2. Описание предметной области 4](#_Toc99531427)

[1.3. Обзор существующих решений 5](#_Toc99531428)

[1.4. Формулировка задания 6](#_Toc99531429)

[1.5. Входные и выходные данные 6](#_Toc99531430)

[**Глава 2. Проектирование задач** 7](#_Toc99531431)

[2.1. Разработка алгоритмов решения задачи 7](#_Toc99531432)

[2.2. Выбор и обоснование инструментов разработки 8](#_Toc99531433)

[**Глава 3. Программная реализация** 9](#_Toc99531434)

[3.1. Структура программы 9](#_Toc99531435)

[**Глава 4. Применение программы** 10](#_Toc99531436)

[4.1. Назначение и условия применения программы 10](#_Toc99531437)

[4.2. Руководство пользователя 10](#_Toc99531438)

[**Заключение 21**](#_Toc99531439)

[**Приложение 22**](#_Toc99531440)

# **Введение**

Экзамен — итоговая форма оценки знаний. В высших учебных заведениях проводятся во время экзаменационных сессий. Экзамены бывают совершенно разными: институтскими, школьными, военными, экзамен в ГАИ, online-экзамен и т. д., — но все они объединены одинаковыми традициями.

Экзамены воспитывают в сдающем чувство ответственности, что очень важно в педагогическом отношении. Если они и требуют большой затраты энергии и сил, причиняют воспитанникам много волнений и беспокойства, то и в этом можно видеть прекрасную подготовку к жизни, вовсе не усеянной розами.

Многие студенты в году небрежно относятся к своим обязанностям, надеясь к экзаменам всё подогнать и восполнить все пробелы; уровень познаний, таким образом, понижается, а отнюдь не повышается, как утверждают защитники экзаменов. Решающее значение экзаменов заставляет учеников прибегать к разным уловкам и обманам (шпаргалки и прочее), лишь бы получить удовлетворительную отметку, и в этом смысле экзамены имеют деморализующее влияние на учеников. Впрочем, использование шпаргалки или телефона является поводом для удаления учащегося с экзамена без права переэкзаменовки

В настоящее время большинство преподавателей и учителей составляют экзаменационные билеты по средством обычных текстовых редакторов.

Цель данного курсового проекта – разработать программное средство, позволяющее упростить работу преподавателей по созданию экзаменационных билетов, а именно автоматически генерировать билеты, а так же представлять готовые билеты к печати.

# **Глава 1. Теоретическая часть разработки программного обеспечения**

## **1.1. Постановка задачи**

Целью курсового проекта является разработка программного средства создания экзаменационных билетов, предназначенного для формирования и печати экзаменационных билетов на основе существующих вопросов.

Программное средство должно включать в себя следующее:

создание и редактирование вопросов для экзаменационных билетов; создание и редактирование экзаменационных билетов вручную пользователем генерация заданного числа билетов на основе существующих вопросов; печать созданных билетов.

## **1.2. Описание предметной области**

Существуют различные формы и виды экзаменов. По теоретическим предметам экзамены проводятся в устной и письменной форме. По своим целям экзамены делятся на школьные и государственные. Существуют следующие виды школьных экзаменов: вступительные, переводные, выпускные. При помощи этих экзаменов проводят проверку знаний вновь поступающих, переходящих в следующий класс (курс) и оканчивающих учебное заведение.

Соблюдение всех стандартов в ведении документации является неотъемлемым условием внедрения политики качества в любой сфере деятельности. Одним из важнейших документов, которые должны быть предоставлены преподавателем по правилам приема экзамена, наряду с рабочей программой является комплект экзаменационных билетов. Сам билет должен соответствовать определенным стандартам: в его шапке должны находиться наименование вуза, название кафедры, место для росписи зав. кафедры.

Обычно задача по составлению и хранению билетов решается при помощи текстового процессора, входящего в пакеты «[Microsoft](http://www.pandia.ru/text/category/microsoft/" \o "Microsoft) Office». В этом случае комплект билетов представляет собой документ данного текстового процессора, на каждой странице которого размещаются один или два (в зависимости от выбранного формата страницы А4 или А5) экземпляра билета. При большом объеме материалов (несколько различных дисциплин, разные студенческие группы со своими рабочими программами) поддерживать в актуальном состоянии и, тем более, вносить изменения, становится для преподавателя непростой задачей, требующей от него значительных затрат времени и сил.

## **1.3. Обзор существующих решений**

Программное средство создания экзаменационных билетов, имеет ряд аналогов.

Экзаменационные билеты – программа представляет собой экзаменационную карточку, по которой сдают различные экзамены. Установлен отсчет времени. После завершения сдачи, программа выдает результаты сдачи: количество данных вами правильных ответов, количество ошибок, и сравнение вашей сдачи с идеальной. Недостатком данной программы является отсутствие возможности печати экзаменационных билетов.

Составитель экзаменационных билетов – небольшая и удобная программа, позволяющая формировать билеты из добавленных вопросов. Билеты сохраняются в базу данных, что позволяет их загружать обратно в программу и редактировать. Недостатками данной программы являются отсутствие возможности автоматической генерации билетов и быстрой навигации по вопросам.

## **1.4. Формулировка задания**

Разработать программное средство создания экзаменационных билетов.

Программное средство должно включать в себя следующее:

добавление и редактирование вопросов; добавление и редактирование экзаменационных билетов пользователем; автоматическая генерация экзаменационных билетов на основе существующих вопросов.

## **1.5. Входные и выходные данные**

В данном программном средстве входные данные представлены следующими сущностями:

вопросы к экзаменационным билетам; экзаменационные билеты.

Входные данные хранятся в виде базы данных MS SQL Server. База данных содержит всего две связанные таблицы: билеты и вопросы.

Выходные данные представлены следующими сущностями:

сформированные пользователем вопросы и билеты; отчет, в виде сформированных билетов.

# **Глава 2. Проектирование задач**

## **2.1. Разработка алгоритмов решения задачи**

Общий алгоритм программы:

пользователь добавляет экзаменационные вопросы в программу; для добавления экзаменационных билетов, пользователь может выбрать один из двух вариантов:

создание экзаменационных билетов вручную, используя созданные вопросы (и в случае надобности, добавить вопросы по ходу создания билетов); автоматически сгенерировать билеты, используя созданные вопросы. Пользователю должна предоставляться возможность выбора количества генерируемых билетов.

печать готовых билетов.

В конечном итоге пользователь имеет готовые экзаменационные билеты, которые можно использовать для проверки знаний учащихся.

## **2.2. Выбор и обоснование инструментов разработки**

Для реализации программного средства был выбран язык программирования C#. C# - объектно-ориентированный язык программирования, для реализации приложений на платформе Framework, то есть приложение написанное на языке C# будет может быть легко перенесено на другую систему с платформой Framework.. Для разработки была выбрана среда Microsoft Visual Studio 2022. Данная среда имеет простой и интуитивно понятный интерфейс, имеется встроенный компилятор и отладчик, имеется подсветка синтаксиса.

Операционная система Windows это современная многозадачная многопользовательская операционная система с графическим интерфейсом пользователя.  
Операционные системы семейства Windows являются наиболее распространенными ОС, которые установлены в домашних и офисных ПК.  
        Графическая оболочка ОС Windows обеспечивает взаимодействие пользователя с компьютером в форме диалога с использованием ввода и вывода на экран дисплея графической информации, управления программами с помощью пиктограмм, меню, окон, панелей (управления, задач, инструментов) и других элементов управления.

Операционная система Windows является самой популярной операционной системой в мире. По состоянию 2022 года под управлением операционных систем семейства Windows по данным ресурса NetMarketShare (Net Applications) работает около 91% персональных компьютеров, что подтверждает популярность данной операционной системы. Windows работает на платформах x86, x86-64, IA-64 и ARM.  Большинство программ, которые выпускают компании и разрабатывают любители-программисты, выпускаются в первую очередь на операционную системы Windows, а затем на другие менее востребованные операционные системы.

# **Глава 3. Программная реализация**

## **3.1. Структура программы**

Программное средство имеет несколько логических частей:

база данных, в которой хранятся все экзаменационные билеты и вопросы. Схема базы данных. Приложение в котором осуществляется работа с билетами. В нем реализованы такие функции, как создание и работа с экзаменационными вопросами и билетами, автоматическая генерация билетов, из существующих вопросов, а также печать существующих билетов.

# **Глава 4. Применение программы**

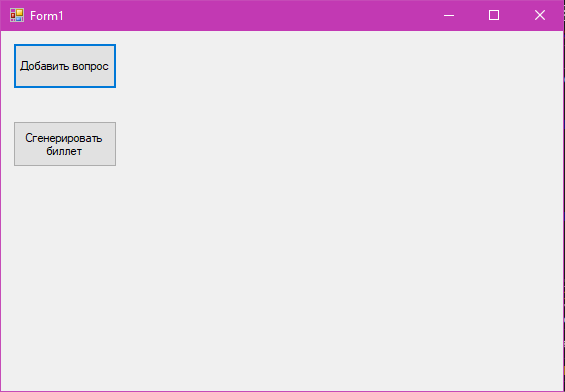
## **4.1. Назначение и условия применения программы**

Целевой аудиторией для данной программы являются преподаватели и учителя, для которых необходимо создавать экзаменационные билеты.

Для нормального функционирования программного средства требуется компьютер под управлением Windows 10 ,с процессором AMD Ryzen 3 1200 или лучше, 8 GB ОЗУ. А также установленный. NET Framework 4 Client Profile и MS SQL Server или выше.

## **4.2. Руководство пользователя**

Для того, чтобы запустить программное средство, необходимо запустить файл Bilet.sln



private void button1\_Click(object sender, EventArgs e)

{

label3.Visible = false;

button5.Visible = false;

textBox2.Visible = false;

button3.Visible = true;

button4.Visible = true;

label1.Visible = true;

label2.Visible = true;

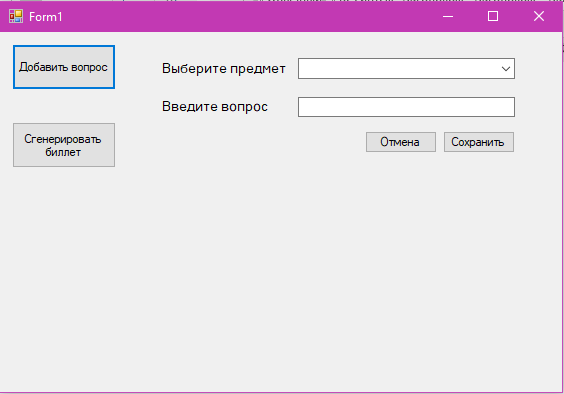
comboBox1.Visible = true;

textBox1.Visible = true;

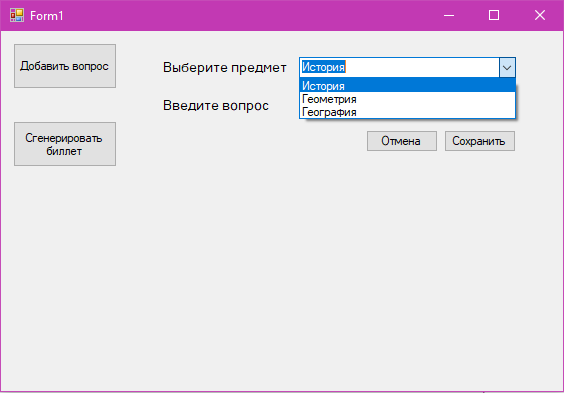
}

Этот метод присваивает значение свойство Visible (т.е. виден элемент или нет, true – виден, false – не виден)

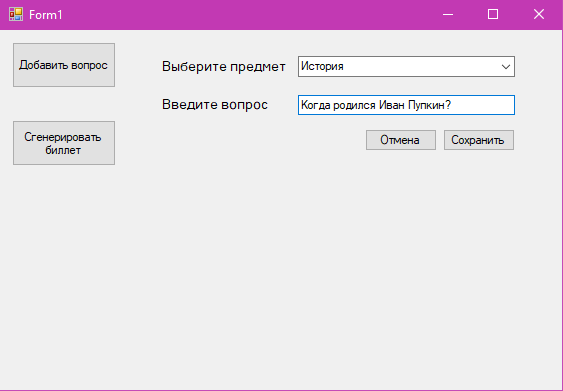
После нажатия у нас стали видны элементы для внесения данных для добавления



Где “выберите предмет”- это выпадающий список предметов (т.е. вопросы по геометрии, географии или любому другому предмету).



Выбираем историю, далее вводим сам вопрос в текстовое поле “Введите вопрос”



Если нажмем отмена то выполнится следующий метод:

private void button4\_Click(object sender, EventArgs e)

{

comboBox1.Text = "";

textBox1.Text = "";

textBox1.Visible = false;

comboBox1.Visible = false;

button4.Visible = false;

button3.Visible = false;

label1.Visible = false;

label2.Visible = false;

label3.Visible=false;

button5.Visible=false;

textBox2.Visible=false;

}

В этом моменте текстовому полю присваивает пустое значение “” и присваивает всем элементам кроме 2 основных первых кнопок (добавить вопрос и сгенерировать билет) свойство Visible = false (Cкрывает кнопки)

Если нажмем на кнопку сохранить то выполнится следующий метод:

private void button3\_Click(object sender, EventArgs e)

{

using (MyDBContext context = new MyDBContext())

{

context.Questions.Add(new Question(comboBox1.Text, textBox1.Text));

context.SaveChanges();

MessageBox.Show("Вопрос добавлен.");

}

comboBox1.Text = "";

textBox1.Text = "";

textBox1.Visible = false;

comboBox1.Visible = false;

button4.Visible = false;

button3.Visible = false;

label1.Visible = false;

label2.Visible = false;

label3.Visible = false;

button5.Visible = false;

textBox2.Visible = false;

}

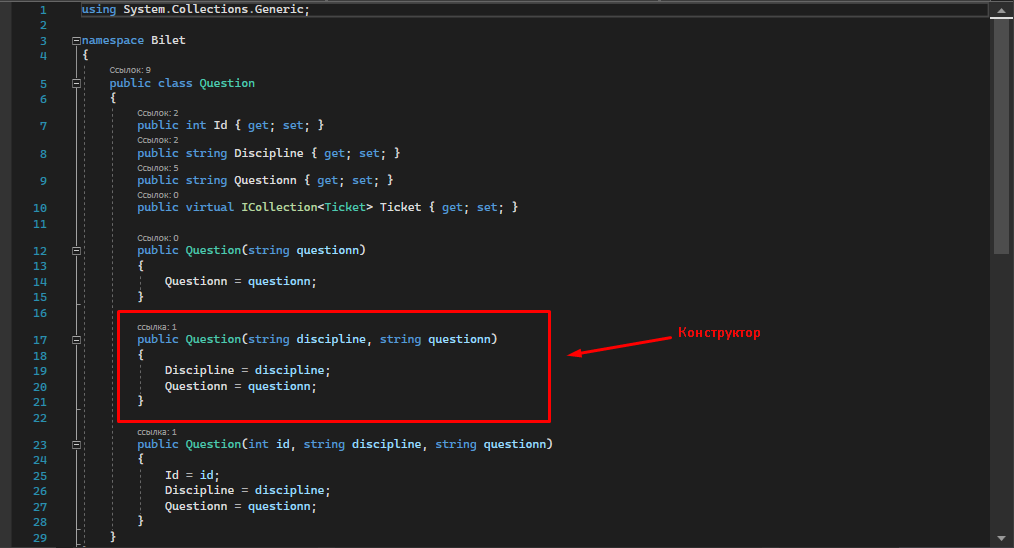
Мы создаем экземпляр нашего класса контекста для взаимодействия с БД и таблицами:

using (MyDBContext context = new MyDBContext())

Далее обращаемся к этому объекту под названием context, при помощи него обращаемся к свойству нашего класса контекста (MyDbContext) – Questions. Это свойство является коллекцией объектов, которая хранит все данные из таблицы Questions в нашей БД.

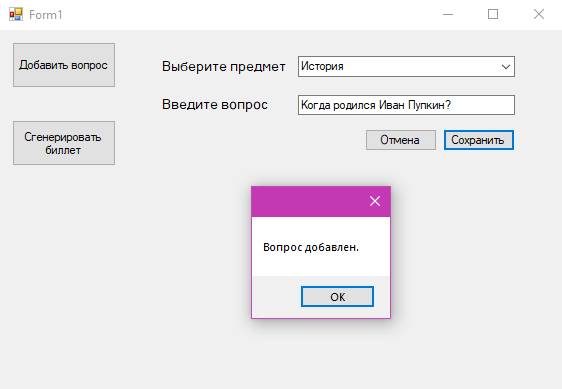
context.Questions.Add(new Question(comboBox1.Text, textBox1.Text));

.Add это встроенный метод который нам добавил класс контекста. Он добавляет новые данные в таблицу Questions. В скобках мы вызываем конструктор который мы объявили в классе Question



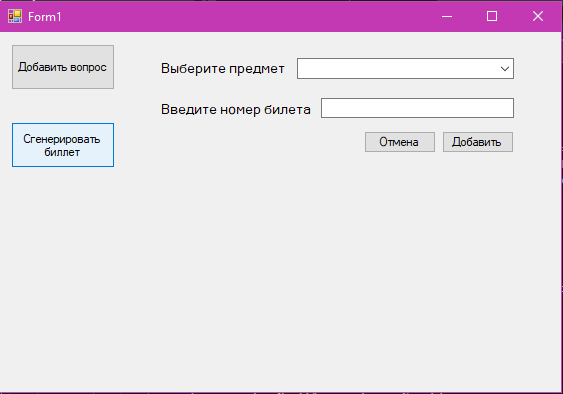
Этот конструктор принимает 2 переменные типа string (строка) и передает их в автосвойства класса, тем самым заполняя данные нашего объекта. После этого сохраняем внесенные изменения в таблицу - context.SaveChanges();

В самом конце у нас вызывается MessageBox.Show("Вопрос добавлен.");Он выводит диалоговое окно в котором написано "Вопрос добавлен."

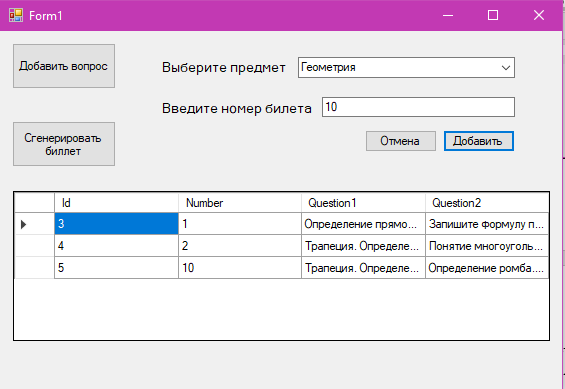


После этого всем ненужным элементам присваивается Visible = false;

Теперь ко второй кнопке: Сгенерировать билет. Нажимая на кнопку она выводит элементы для выбора предмета и ввода номера самого билета.



Предметы там те же что и в прошлом элементе (а точнее это один и тот же элемент, просто данные мы передаем в разные таблицы). Выбрали предмет Геометрия, вписываем номер для билета, допустим 10. Дальше при нажатии на кнопку добавить будет происходить следующее: считываются данные из таблицы где хранятся наши вопросы, предмет у которых так же Геометрия. После этого рандомно выбираются 2 индекса вопросов и вписываются в билет и потом сохраняются изменения. Так же выводится элемент для отображение данных в табличном виде.



private void button5\_Click(object sender, EventArgs e)

{

if (comboBox1.Text != "" && textBox2.Text != "") // если номер билета и предмет не пустые, т.е. их заполнили то будет выполняться код ниже

{

using (MyDBContext context = new MyDBContext()) //создание объекта класса контекста для осуществления взаимодействия с БД и таблицами

{

List<Question> questions = new List<Question>(); //создаем коллекцию объектов класса Question для того чтобы потом туда записать данные из таблицы

var discipline = comboBox1.Text; //просто переменная в которую мы запоминаем предмет который выбрали

string Connection = @"Server=(localdb)\mssqllocaldb;Database=Konstruktor;Trusted\_Connection=True;"; //строка подключения к серверу Sql и твоей БД

string query = $"select \* from Questions where Discipline = N'{discipline}'"; //запрос который передается в sql, выбирает все значения столбцов в таблице Questions если предмет равен тому который мы выбрали

using (SqlConnection connection = new SqlConnection(Connection))

//создаем экземпляр объекта класса который создает подключение к БД и в скобках передаем строку подключения которую объявили раньше

{

connection.Open(); //открываем подключение

SqlCommand cmd = new SqlCommand(query, connection); //создаем команду которой в скобках на первом месте передаем запрос, который будет передан в SQL для выполнения, а на втором месте наша строка подключения

SqlDataReader reader = cmd.ExecuteReader(); //объявляем объект класса, который считывает данные из таблицы и присваиваем ему cmd.ExecuteReader(); - запускает считывание данных по переданному ему ранее запросу - query

while (reader.Read()) //цикл будет выполняться пока происходит считывания

{

questions.Add(new Question(reader.GetInt32(0), reader.GetString(1), reader.GetString(2))); //добавляем в коллекцию, которую объявили в начале метода, считываемые данные при помощи конструктора который принимает Id, название предмета и сам вопрос

}

reader.Close(); //закрываем чтение

var index = random.Next(1, questions.Count);//радомим индекс для того чтобы получить рандомные вопросы которые будут добавлены в наш билет

Ticket ticket = new Ticket(); //создаем объект класса Ticket

ticket.Number = Convert.ToInt32(textBox2.Text);

ticket.Question1 = questions[index].Questionn;

index = random.Next(1, questions.Count);

ticket.Question2 = questions[index].Questionn;

ticket.QuestionId = questions[index].Id; // все от этой строки и выше это присвоение значений его свойствам

context.Tickets.Add(ticket); //добавляем наш объект в таблице Ticket (билеты)

context.SaveChanges(); //сохраняем изменения

connection.Close(); //закрываем соединение

Form1\_Load(sender, e); //загружаем в таблицу данные из нашей таблицы билетов в БД

dataGridView1.Visible = true; //делаем таблицу с данными видимой на форме

}

}

}

else

MessageBox.Show("Не все данные были внесены!"); //если поля пустые выбъет об этом сообщение и надо будет ввести данные для генерации билета

}

# **Заключение**

В данном курсовом проекте было реализовано программное средство создания экзаменационных билетов на языке С# в среде разработке Microsoft Visual Studio, которое включает следующие функции:

создание экзаменационных вопросов и билетов; генерация экзаменационных билетов на основе существующих вопросов.

В ходе выполнения курсового проекта было выполнено следующее:

постановка задачи, в которой были описаны предметная область, формулировка задания, входные и выходные данные, а также существующие аналоги со своими достоинствами и недостатками; проектирование задачи. В ходе проектирования задачи были разработаны и описаны алгоритмы решения задачи, спроектированы диаграммы и выбраны инструменты разработки; программная реализация. Было реализовано программное средство создания экзаменационных билетов, а также описаны структура программы и разработанные классы и методы; описание применения программы, в котором описаны назначение и условия применения программы, а также подробное руководство пользователя.

Данным программным средством могут пользоваться все пользователи, которых он устраивает по функциональным возможностям. Данное программное средство может использоваться в сфере образования для различных учреждений образования.

# **Приложение**

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace Bilet

{

public partial class Form1 : Form

{

public static Random random = new Random();

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

label3.Visible = false;

button5.Visible = false;

textBox2.Visible = false;

button3.Visible = true;

button4.Visible = true;

label1.Visible = true;

label2.Visible = true;

comboBox1.Visible = true;

textBox1.Visible = true;

}

private void button3\_Click(object sender, EventArgs e)

{

using (MyDBContext context = new MyDBContext())

{

context.Questions.Add(new Question(comboBox1.Text, textBox1.Text));

context.SaveChanges();

MessageBox.Show("Вопрос добавлен.");

}

comboBox1.Text = "";

textBox1.Text = "";

textBox1.Visible = false;

comboBox1.Visible = false;

button4.Visible = false;

button3.Visible = false;

label1.Visible = false;

label2.Visible = false;

label3.Visible = false;

button5.Visible = false;

textBox2.Visible = false;

}

private void button4\_Click(object sender, EventArgs e)

{

comboBox1.Text = "";

textBox1.Text = "";

textBox1.Visible = false;

comboBox1.Visible = false;

button4.Visible = false;

button3.Visible = false;

label1.Visible = false;

label2.Visible = false;

label3.Visible=false;

button5.Visible=false;

textBox2.Visible=false;

}

private void button2\_Click(object sender, EventArgs e)

{

label2.Visible = false;

textBox1.Visible=false;

button3.Visible=false;

button4.Visible=true;

button5.Visible = true;

label1.Visible = true;

comboBox1.Visible = true;

label3.Visible = true;

textBox2.Visible = true;

}

private void Form1\_Load(object sender, EventArgs e)

{

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

this.ticketsTableAdapter.Fill(this.konstruktorDataSet.Tickets);

}

private void button5\_Click(object sender, EventArgs e)

{

if (comboBox1.Text != "" && textBox2.Text != "")

{

using (MyDBContext context = new MyDBContext())

{

List<Question> questions = new List<Question>();

var discipline = comboBox1.Text;

string Connection = @"Server=(localdb)\mssqllocaldb;Database=Konstruktor;Trusted\_Connection=True;";

string query = $"select \* from Questions where Discipline = N'{discipline}'";

using (SqlConnection connection = new SqlConnection(Connection))

{

connection.Open();

SqlCommand cmd = new SqlCommand(query, connection);

SqlDataReader reader = cmd.ExecuteReader();

while (reader.Read())

{

questions.Add(new Question(reader.GetInt32(0), reader.GetString(1), reader.GetString(2)));

}

reader.Close();

var index = random.Next(1, questions.Count);

Ticket ticket = new Ticket();

ticket.Number = Convert.ToInt32(textBox2.Text);

ticket.Question1 = questions[index].Questionn;

index = random.Next(1, questions.Count);

ticket.Question2 = questions[index].Questionn;

ticket.QuestionId = questions[index].Id;

context.Tickets.Add(ticket);

context.SaveChanges();

connection.Close();

Form1\_Load(sender, e);

dataGridView1.Visible = true;

}

}

}

else

MessageBox.Show("Не все данные были внесены!");

}

}

}

using System.Data.Entity;

namespace Bilet

{

class MyDBContext : DbContext

{

public MyDBContext() : base("DbConnect")

{

}

public DbSet<Question> Questions { get; set; }

public DbSet<Ticket> Tickets { get; set; }

}

}

namespace Bilet

{

public class Ticket

{

public int Id { get; set; }

public int Number { get; set; }

public string Question1 { get; set; }

public string Question2 { get; set; }

public int QuestionId { get; set; }

public virtual Question Question { get; set; }

}

}

using System.Collections.Generic;

namespace Bilet

{

public class Question

{

public int Id { get; set; }

public string Discipline { get; set; }

public string Questionn { get; set; }

public virtual ICollection<Ticket> Ticket { get; set; }

public Question(string questionn)

{

Questionn = questionn;

}

public Question(string discipline, string questionn)

{

Discipline = discipline;

Questionn = questionn;

}

public Question(int id, string discipline, string questionn)

{

Id = id;

Discipline = discipline;

Questionn = questionn;

}

}

}