СОДЕРЖАНИЕ

[Введение 4](#_Toc3233859)

[1 Экономическая характеристика объекта 5](#_Toc3233860)

[2 Постановка задачи 6](#_Toc3233861)

[2.1 Экономическое содержание задачи 6](#_Toc3233862)

[2.2 Входная информация 6](#_Toc3233863)

[3 Технические и инструментальные средства для решения задачи 7](#_Toc3233864)

[3.1 Обоснование выбора языка программирования 7](#_Toc3233865)

[3.2 Технические требования к компьютеру для решения задачи 9](#_Toc3233866)

[4 Информационное обеспечение задачи 10](#_Toc3233867)

[4.1 Классификаторы, используемые для решения задачи 10](#_Toc3233868)

[5 Программное обеспечение задачи 13](#_Toc3233869)

[5.1 Описание методов 13](#_Toc3233870)

[5.2 Алгоритм решения задачи 15](#_Toc3233871)

[5.3 Описание блок – схемы 15](#_Toc3233872)

[Заключение 19](#_Toc3233873)

[Список использованных источников 20](#_Toc3233874)

[Приложение А 21](#_Toc3233875)

[Приложение Б 22](#_Toc3233876)

# Введение

В рамках распоряжения Правительства Российской Федерации от 17 декабря 2009 года № 1993-р утверждён сводный перечень первоочередных государственных и муниципальных услуг, предоставляемых органами исполнительной власти субъектов Российской Федерации и органами местного самоуправления в электронном виде, а также услуг, предоставляемых в электронном виде учреждениями субъектов Российской Федерации и муниципальными учреждениями согласно приложению № 1 этого распоряжения.

Каждое образовательное учреждение имеет какой — либо оборот информации. В каждом образовательном учреждении есть какое — либо количество работников, которые обязаны работать с этой информацией. Чтобы упростить работу со этой информацией её хранят в цифровом формате. Как раз по этому и необходимы электронные журналы для работы с информацией об оценках, учениках(студентах), группах, дисциплинах и т.д.

Актуальность темы обусловлена тем, что она позволяет без проблем осуществлять некоторые действия над студентами, группами, оценками и дисциплинами.

Цель курсового проекта заключается в создании программного модуля оценки знаний студентов на базе ОС Windows.

В курсовом проекте решаются задачи:

* формирование базы данных;
* просмотра оценок по дисциплинам;
* просмотра оценок у определённых учеников(студентов);
* просмотра оценок по группам;
* изменение групп, студентов, дисциплин и оценок с помощью приложения на ПК.

# 1 Экономическая характеристика объекта

Государственное бюджетное профессиональное образовательное учреждение Ростовской области «Волгодонский техникум информационных технологий, бизнеса и дизайна имени В. В. Самарского» основано в 1990 году в г. Волгодонске, как профессионально-техническое училище №105.  
В соответствии с приказом министерства общего и профессионального образования Ростовской области от 17 сентября 2014 года № 335-к «О переименовании учреждения и подтверждении работы в должности» техникум переименован в государственное бюджетное профессиональное образовательное учреждение Ростовской области «Волгодонский техникум информационных технологий, бизнеса и дизайна имени В.В. Самарского».

Информатизация образовательного процесса и управления ГБПОУ РО «ВТИТБиД» реализуется по следующим направлениям:

* информатизация процесса обучения;
* информатизация профориентационной работы;
* информатизация методической работы;
* информатизация управленческой работы администрации и сотрудников.

Информационно-технологическая инфраструктура техникума соответствует современному уровню и включает в себя локальную сеть, объединяющую персональные компьютеры, копировально-множительную технику, презентационное оборудование.

Для подготовки обучающихся к занятиям с использованием сетевых учебных ресурсов и информационных интернет-ресурсов в техникуме имеется 16 компьютерных классов и читальный зал.

Парк компьютерной техники постоянно развивается и качественно и количественно. В настоящее время общий парк вычислительной техники в техникуме насчитывает 438 единиц IBM-совместимой вычислительной техники, из них стационарных персональных компьютеров – 317 ед., ноутбуков – 121 ед.

# 2 Постановка задачи

## 2.1 Экономическое содержание задачи

Цель курсового проекта – создание программного модуля оценки знаний студентов на базе ОС Windows для ГБПОУ РО "Волгодонский техникум информационных технологий, бизнеса и дизайна им. В.В. Самарского"

Разработанный модуль позволяет производить различные действия над студентами, оценками, группами и дисциплинами с помощью компьютера.

Программа создана на языке С#, графический интерфейс программы дружественный и понятный, сама программа проста в использовании и не требует специального образования или прохождение курсов.

## 2.2 Входная информация

Входная информация представляет собой данные об оценках, студентах, дисциплинах и группах, хранящиеся в базе данных.

2.3 Выходная информация

Выходной информацией является уже обновленные данные об оценках, студентах, группах и дисциплинах над которыми были произведены действия доступные в программе.

# 3 Технические и инструментальные средства для решения задачи

## 3.1 Обоснование выбора языка программирования

Данное приложение сделано с помощью языка 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 и .NET Core. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.

В качестве среды для разработки приложения была выбрана программа Microsoft Visual Studio, линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов. Данные продукты позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight. Существует множество сред разработки, но данная была выбрана в ввиду ее удобного графического интерфейса и средств отладки. Microsoft Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Microsoft Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

Главные возможности среды разработки Microsoft Visual Studio:

* включены все «интеллектуальные» возможности по редактированию кода;
* есть возможность визуального просмотра будущего приложения;
* сборка проекта работает быстро;
* удобный конструктор интерфейсов;
* удобное и интуитивно понятное логирование проекта;
* указанные цвета и рисунки, использованные в layout'e отображаются на границе в виде небольших превью, которые легко помогают понять какой конкретно ресурс используется;
* среда разработки является технологиями компании Microsoft;
* при выборе ресурса, его содержимое отображается во всплывающих окнах;
* возможность создания одним кликом новых окон и страниц;
* мониторинг используемой памяти;
* возможность подключать сторонние дополнения(плагины);
* возможность добавления новых инструментов;

## 3.2 Технические требования к компьютеру для решения задачи

Рекомендуемые системные требования к компьютеру на которой будет установлена IDE Android Studio следующие:

* Операционная система – Microsoft® Windows® 7/8/10 (32- или 64-bit);
* оперативная память – 2 ГБ минимум, рекомендуется 4 ГБ;
* пространство на жестком диске – рекомендуется 3 ГБ;
* монитор – 1280 х 720 минимальное разрешение экрана.

# 4 Информационное обеспечение задачи

## 4.1 Классификаторы, используемые для решения задачи

Классификатор представляет собой набор таблиц (справочников).

Объект – некоторая сущность в цифровом пространстве, обладающая определённым состоянием и поведением в данной предметной области.

Атрибут – элемент, описывающий любую характеристику объекта, важную для данной предметной области. Объектом в данной базе данных будет: информация о группах приведенная в таблице 4.1, студентах показанных на таблице 4.2, оценках на таблице 4.3 и дисциплинах на таблице 4.4.

Таблица 4.1 – Атрибуты группы

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Группа | Идентификационный код  Название  Курс |

Таблица 4.2 – Атрибуты студента

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Студент | Идентификационный код  Имя  Фамилия  Отчество  Идентификационный код группы |

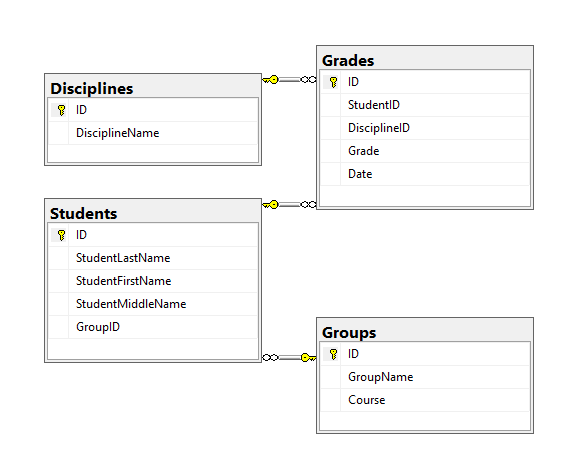
Таблица 4.3 – Атрибуты оценки

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Оценка | Идентификационный код  Идентификационный код студента  Идентификационный код дисциплины  Оценка  Дата выставления оценки |

Таблица 4.4 – Атрибуты дисциплины

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Дисциплина | Идентификационный код  Название |

Для выполнения проекта была выбрана СУБД SQL Server. Схема БД представлена на рисунке 4.1.

Рисунок 4.1 – Связи таблиц в БД

Структура таблиц группы, студенты, оценки и дисциплины базы данных представлена на рисунках 4.2 – 4.5 соответственно.

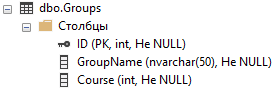


Рисунок 4.2 – Таблица «Группы»

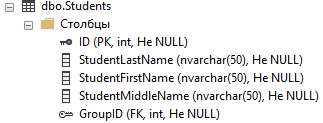


Рисунок 4.3 – Таблица «Студенты»

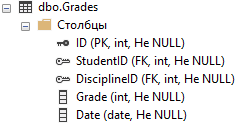


Рисунок 4.4 – Таблица «Оценки»

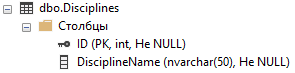


Рисунок 4.5 – Таблица «Дисциплины»

# 5 Программное обеспечение задачи

## 5.1 Описание методов

В примера взята часть кода для входа в приложение под своей учетной записью. При нажатии на кнопку вызывается основное окно, для дальнейшей работы с приложением.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WPF\_StudentsAchievement\_Project.Resources.Profiles;

using WPF\_StudentsAchievement\_Project.Resources.MVVM.View;

using WPF\_StudentsAchievement\_Project.Resources.Checks;

using System.Data;

using System.Data.SqlClient;

namespace WPF\_StudentsAchievement\_Project

{

/// <summary>

/// Interaction logic for LoginWindow.xaml

/// </summary>

public partial class LoginWindow : Window

{

public LoginWindow()

{

InitializeComponent();

}

Check check = new Check();

User user = new User();

DataBase dataBase = new DataBase();

private void RegistrationButton\_Click(object sender, RoutedEventArgs e)

{

RegistrationWindow registrationWindow = new RegistrationWindow();

registrationWindow.Show();

this.Hide();

}

private void LoginButton\_Click(object sender, RoutedEventArgs e)

{

user.Login = LoginTextBox.Text;

user.Password = PasswordTextBox.Password;

try

{

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string queryString = $"SELECT Login,Password FROM Register WHERE Login='{user.Login}' AND Password='{user.Password}'";

SqlCommand command = new SqlCommand(queryString, dataBase.getConnection());

dataBase.openConnection();

adapter.SelectCommand = command;

adapter.Fill(table);

if (table.Rows.Count == 1)

{

MessageBox.Show($"Вход в аккаунт {user.Login}!", "УСПЕШНО", MessageBoxButton.OK, MessageBoxImage.Information);

User.ShowUserLogin = user.Login;

WorkWindow workWindow = new WorkWindow();

workWindow.Show();

this.Close();

}

else

{

MessageBox.Show("Такого аккаунта не существует!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

dataBase.closeConnection();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void MovingWindow(object sender, RoutedEventArgs e)

{

if (Mouse.LeftButton == MouseButtonState.Pressed)

{

this.DragMove();

}

}

private void ExitButton\_Click(object sender, RoutedEventArgs e)

{

Close();

}

}

}

## 5.2 Алгоритм решения задачи

Для проведения каких-либо действий с оценками, студентами, группами или дисциплинами необходимо выбрать действие: удалить, редактировать или добавить.

## 5.3 Описание блок – схемы

Последовательное описание блок – схемы приложения приведено ниже:

Шаг 1. Запуск программы.

Шаг 2. Вопрос: Есть ли у вас аккаунт? Если нет, то переходим к шагу 3. Если есть то переходим к шагу 4.

Шаг 3. Регистрируемся. Возвращаемя к авторизации.

Шаг 4. Вход в приложение под своей учетной записью.

Шаг 5. Вопрос: Выбрано ли действие? Если да, то переходим к шагу 6, если нет, то переходим к 8 шагу.

Шаг 6. Выполнение

Шаг 7. Возвращаемся к 4 шагу.

Шаг 8. Закрытие приложения.

Блок-схема приведена в приложении А.

Листинг программы приведен в приложении Б.

5.4 Описание интерфейса программы

В окне входа в приложение, представленном на рисунке 5.1, находятся два текстовых поля логин и пароль, кнопка авторизации и регистрации. Для дальнейшей работы нужно верно указать логин и пароль. Если логин и пароль верны, то откроется основное окно, представленное на рисунке 5.2 с возможностью выбора действия. При нажатии на кнопку происходит открытие нужного нам окна, которое показана на рисунке 5.3. На верхней панели выбери нужную категорию, как на рисунке 5.4.

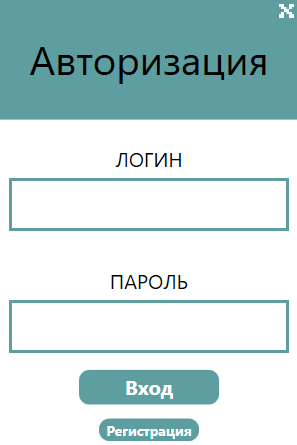


Рисунок 5.1 – Окно входа

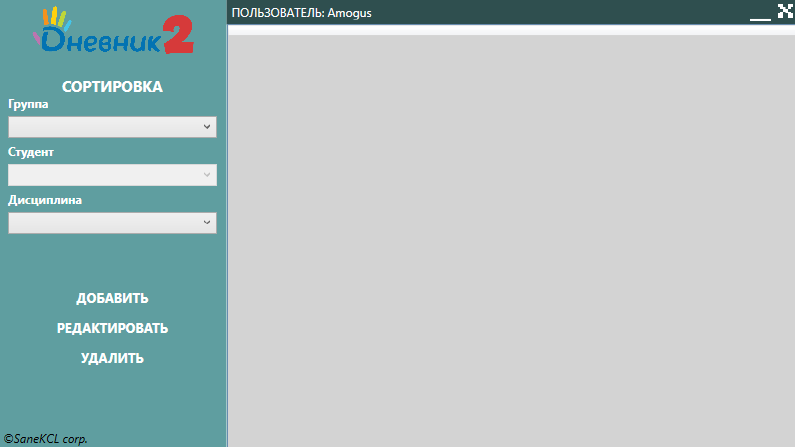


Рисунок 5.2 – Основное окно

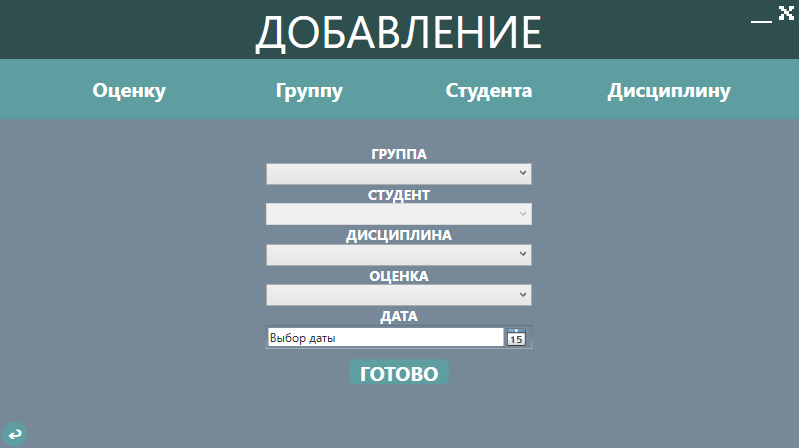


Рисунок 5.3 – Окно согласования

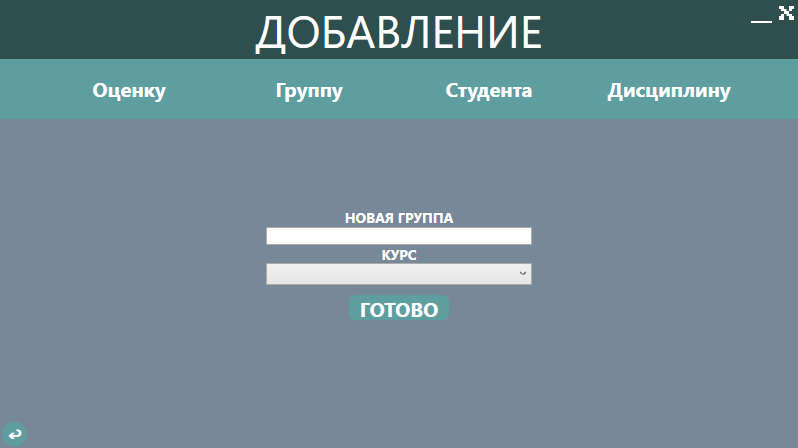


Рисунок 5.4 – Окно согласования

Для успешной работы с программой необходимо:

1. ОС Windows.
2. Соеденение с БД.
3. Учетная запись.

# Заключение

Разработанный программный модуль оценки знаний студентов для ГБПОУ РО "ВТИТБиД" представляет собой простое приложение.

Программа разработана на языке С# в интегрированной среде разработки Microsoft Visual Studio с использованием СУБД SQL Server. Приложение интуитивно понятно даже неопытному пользователю и не затрудняет его работу с оценками, группами, дисциплинами и студентами.

В дальнейшем в приложение будет обновляться и будут добавляться новые функции необходимые для работы с данными.

В процессе написания кода для данной программы были приобретены практические навыки в работе в СУБД SQL Server и языке программирования С#.

Цель курсового проекта достигнута. Задачи выполнены.

# Список использованных источников

Интернет ресурсы:

1. ГБПОУ РО "ВТИТБиД". Сведения компании [Электронный ресурс] – Режим доступа: http://vtitbid.ru/
2. Википедия. Язык программирования C# [Электронный ресурс] -Режим доступа: https://ru.wikipedia.org/wiki/C\_Sharp

# Приложение А

(обязательное)

Блок – схема программы

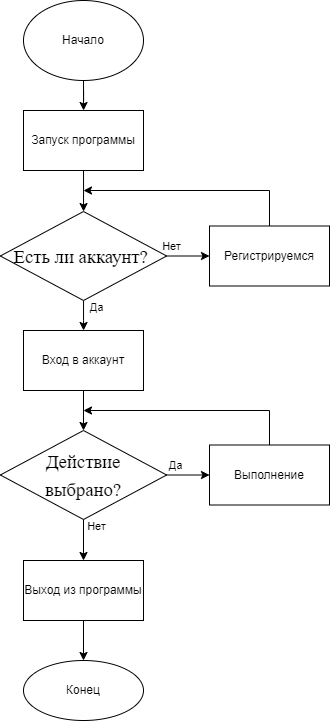


Рисунок А.1–Блок схема программы

# Приложение Б

(обязательное)

Листинг программы

USE StudentsAchievement

GO

CREATE TABLE Groups(

ID INT PRIMARY KEY IDENTITY NOT NULL,

[GroupName] NVARCHAR(50) NOT NULL,

[Course] INT NOT NULL,

)

GO

CREATE TABLE Students(

ID INT PRIMARY KEY IDENTITY NOT NULL,

[StudentLastName] NVARCHAR(50) NOT NULL,

[StudentFirstName] NVARCHAR(50) NOT NULL,

[StudentMiddleName] NVARCHAR(50) NOT NULL,

[GroupID] INT NOT NULL,

CONSTRAINT FK\_Students\_GroupID FOREIGN KEY ([GroupID]) REFERENCES Groups(ID),

)

GO

CREATE TABLE Disciplines(

ID INT PRIMARY KEY IDENTITY NOT NULL,

[DisciplineName] NVARCHAR(50) NOT NULL,

)

GO

CREATE TABLE Grades(

ID INT PRIMARY KEY IDENTITY NOT NULL,

[StudentID] INT NOT NULL,

[DisciplineID] INT NOT NULL,

[Grade] INT NOT NULL,

[Date] Date NOT NULL,

CONSTRAINT FK\_Grades\_StudentID FOREIGN KEY ([StudentID]) REFERENCES Students(ID),

CONSTRAINT FK\_Grades\_DisciplineID FOREIGN KEY ([DisciplineID]) REFERENCES Disciplines(ID),

)

GO

INSERT INTO [Groups]([GroupName],[Course]) VALUES ('ИСП-19',3)

INSERT INTO [Groups]([GroupName],[Course]) VALUES ('ИСП-19К',2)

INSERT INTO [Groups]([GroupName],[Course]) VALUES ('ИСП-24',2)

GO

INSERT INTO [Students]([StudentLastName],[StudentFirstName],[StudentMiddleName],[GroupID]) VALUES ('Аксёнов','Александр','Игоревич',1)

INSERT INTO [Students]([StudentLastName],[StudentFirstName],[StudentMiddleName],[GroupID]) VALUES ('Котлетов','Егор','Сергеевич',2)

INSERT INTO [Students]([StudentLastName],[StudentFirstName],[StudentMiddleName],[GroupID]) VALUES ('Атлетов','Антон','Гераклович',3)

GO

INSERT INTO [Disciplines]([DisciplineName]) VALUES ('Математика')

INSERT INTO [Disciplines]([DisciplineName]) VALUES ('Физ-ра')

INSERT INTO [Disciplines]([DisciplineName]) VALUES ('Русский Язык')

INSERT INTO [Disciplines]([DisciplineName]) VALUES ('Литература')

INSERT INTO [Disciplines]([DisciplineName]) VALUES ('История')

GO

INSERT INTO [Grades]([StudentID],[DisciplineID],[Grade],[Date]) VALUES (1,1,3,'2008-10-24')

INSERT INTO [Grades]([StudentID],[DisciplineID],[Grade],[Date]) VALUES (1,3,5,'2008-10-23')

INSERT INTO [Grades]([StudentID],[DisciplineID],[Grade],[Date]) VALUES (1,2,4,'2008-10-28')

INSERT INTO [Grades]([StudentID],[DisciplineID],[Grade],[Date]) VALUES (2,4,3,'2008-10-26')

INSERT INTO [Grades]([StudentID],[DisciplineID],[Grade],[Date]) VALUES (2,2,5,'2008-10-28')

INSERT INTO [Grades]([StudentID],[DisciplineID],[Grade],[Date]) VALUES (2,1,4,'2008-10-28')

GO

SELECT Groups.ID,Groups.GroupName,Groups.Course FROM Groups

SELECT Students.ID, Students.StudentLastName,Students.StudentFirstName,Students.StudentMiddleName,Groups.GroupName FROM Students

left join Groups on Groups.ID = Students.GroupID

SELECT \* FROM Disciplines

SELECT Students.ID,Students.StudentLastName,Students.StudentFirstName,Disciplines.DisciplineName,Grades.Grade,Grades.[Date] FROM Grades

left join Disciplines on Disciplines.ID = Grades.DisciplineID

left join Students on Students.ID = Grades.StudentID

GO

CREATE TABLE Register(

ID INT PRIMARY KEY IDENTITY NOT NULL,

[Login] VARCHAR(20) NOT NULL,

[Password] VARCHAR(30) NOT NULL

)

INSERT INTO Register([Login],[Password]) VALUES('Amogus','usus')

SELECT \* FROM Register

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WPF\_StudentsAchievement\_Project.Resources.Profiles;

using WPF\_StudentsAchievement\_Project.Resources.MVVM.View;

using WPF\_StudentsAchievement\_Project.Resources.Checks;

using System.Data;

using System.Data.SqlClient;

namespace WPF\_StudentsAchievement\_Project

{

public partial class LoginWindow : Window

{

public LoginWindow()

{

InitializeComponent();

}

Check check = new Check();

User user = new User();

DataBase dataBase = new DataBase();

private void RegistrationButton\_Click(object sender, RoutedEventArgs e)

{

RegistrationWindow registrationWindow = new RegistrationWindow();

registrationWindow.Show();

this.Hide();

}

private void LoginButton\_Click(object sender, RoutedEventArgs e)

{

user.Login = LoginTextBox.Text;

user.Password = PasswordTextBox.Password;

try

{

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string queryString = $"SELECT Login,Password FROM Register WHERE Login='{user.Login}' AND Password='{user.Password}'";

SqlCommand command = new SqlCommand(queryString, dataBase.getConnection());

dataBase.openConnection();

adapter.SelectCommand = command;

adapter.Fill(table);

if (table.Rows.Count == 1)

{

MessageBox.Show($"Вход в аккаунт {user.Login}!", "УСПЕШНО", MessageBoxButton.OK, MessageBoxImage.Information);

User.ShowUserLogin = user.Login;

WorkWindow workWindow = new WorkWindow();

workWindow.Show();

this.Close();

}

else

{

MessageBox.Show("Такого аккаунта не существует!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

dataBase.closeConnection();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void MovingWindow(object sender, RoutedEventArgs e)

{

if (Mouse.LeftButton == MouseButtonState.Pressed)

{

this.DragMove();

}

}

private void ExitButton\_Click(object sender, RoutedEventArgs e)

{

Close();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using System.Data;

using System.Data.SqlClient;

using WPF\_StudentsAchievement\_Project.Resources.Checks;

using WPF\_StudentsAchievement\_Project.Resources.Profiles;

namespace WPF\_StudentsAchievement\_Project.Resources.MVVM.View

{

public partial class RegistrationWindow : Window

{

Check check = new Check();

User user = new User();

DataBase dataBase = new DataBase();

public RegistrationWindow()

{

InitializeComponent();

}

private void MovingWindow(object sender, RoutedEventArgs e)

{

if (Mouse.LeftButton == MouseButtonState.Pressed)

{

this.DragMove();

}

}

private void RegistrationButton\_Click(object sender, RoutedEventArgs e)

{

user.Login = LoginTextBox.Text;

user.Password = PasswordTextBox.Text;

try

{

if (check.User(user.Login, user.Password) == false)

{

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string queryString = $"INSERT INTO Register VALUES('{user.Login}', '{user.Password}')";

SqlCommand command = new SqlCommand(queryString, dataBase.getConnection());

dataBase.openConnection();

if (command.ExecuteNonQuery()== 1)

{

MessageBox.Show($"Аккаунт {user.Login} успешно зарегистрирован!", "УСПЕШНО",MessageBoxButton.OK,MessageBoxImage.Information);

LoginWindow loginWindow = new LoginWindow();

this.Close();

loginWindow.ShowDialog();

}

else

{

MessageBox.Show("ОШИБКА СОЗДАНИЯ АККАУНТА:", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

dataBase.closeConnection();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void LoginButton\_Click(object sender, RoutedEventArgs e)

{

LoginWindow loginWindow = new LoginWindow();

this.Hide();

loginWindow.Show();

}

private void ExitButton\_Click(object sender, RoutedEventArgs e)

{

this.Close();

LoginWindow loginWindow = new LoginWindow();

loginWindow.Close();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using System.Data;

using System.Data.SqlClient;

using WPF\_StudentsAchievement\_Project.Resources.Profiles;

using WPF\_StudentsAchievement\_Project.Resources.Checks;

namespace WPF\_StudentsAchievement\_Project.Resources.MVVM.View

{

public partial class WorkWindow : Window

{

DataBase dataBase = new DataBase();

User user = new User();

DataBaseField baseField = new DataBaseField();

Check check = new Check();

public WorkWindow()

{

InitializeComponent();

}

private void MovingWindow(object sender, RoutedEventArgs e)

{

if (Mouse.LeftButton == MouseButtonState.Pressed)

{

this.DragMove();

}

}

public string standartQuery = $"SELECT Students.ID,Students.StudentLastName,Students.StudentFirstName,Disciplines.DisciplineName,Grades.Grade,Grades.[Date] FROM Grades " +

$"left join Disciplines on Disciplines.ID = Grades.DisciplineID " +

$"left join Students on Students.ID = Grades.StudentID ";

private void ExitButton\_Click(object sender, RoutedEventArgs e)

{

Close();

LoginWindow loginWindow = new LoginWindow();

RegistrationWindow registrationWindow = new RegistrationWindow();

loginWindow.Close();

registrationWindow.Close();

}

private void HideButton\_Click(object sender, RoutedEventArgs e)

{

this.WindowState = WindowState.Minimized;

}

public void CheckComboBoxes()

{

if (GroupComboBox.Text.Length!=0 & StudentComboBox.Text.Length!=0 & DisciplineComboBox.Text.Length!=0)

{

InfoDataGridRefresh(standartQuery + $"WHERE Students.ID ='{baseField.StudentId}' AND Disciplines.ID='{baseField.DisciplineId}'");

}

else if(GroupComboBox.Text.Length!=0 & StudentComboBox.Text.Length==0 & DisciplineComboBox.Text.Length!=0)

{

InfoDataGridRefresh(standartQuery + $"WHERE Students.GroupID ='{baseField.GroupId}' AND Disciplines.ID='{baseField.DisciplineId}'");

}

else if (GroupComboBox.Text.Length != 0 & StudentComboBox.Text.Length != 0 & DisciplineComboBox.Text.Length == 0)

{

InfoDataGridRefresh(standartQuery + $"WHERE Students.ID ='{baseField.StudentId}'");

}

else if (GroupComboBox.Text.Length == 0 & StudentComboBox.Text.Length == 0 & DisciplineComboBox.Text.Length != 0)

{

InfoDataGridRefresh(standartQuery + $"WHERE Disciplines.ID ='{baseField.DisciplineId}'");

}

else if (GroupComboBox.Text.Length != 0 & StudentComboBox.Text.Length == 0 & DisciplineComboBox.Text.Length == 0)

{

InfoDataGridRefresh(standartQuery + $"WHERE Students.GroupID ='{baseField.GroupId}'");

}

}

public void InfoDataGridRefresh(string queryString)

{

dataBase.openConnection();

SqlCommand command = new SqlCommand(queryString, dataBase.getConnection());

command.ExecuteNonQuery();

SqlDataAdapter adapter = new SqlDataAdapter(command);

DataTable table = new DataTable();

adapter.Fill(table);

InfoDataGrid.ItemsSource = table.DefaultView;

adapter.Update(table);

dataBase.closeConnection();

}

public void GroupComboBoxRefresh()

{

dataBase.openConnection();

string queryString = "SELECT ID,GroupName FROM Groups";

SqlCommand command = new SqlCommand(queryString, dataBase.getConnection());

SqlDataReader sqlDataReader = command.ExecuteReader();

GroupComboBox.Items.Clear();

while (sqlDataReader.Read())

{

int GroupId = sqlDataReader.GetInt32(0);

string GroupName = sqlDataReader.GetString(1);

GroupComboBox.Items.Add($"{GroupId}: {GroupName}");

}

dataBase.closeConnection();

}

private void GroupComboBox\_DropDownClosed(object sender, EventArgs e)

{

dataBase.openConnection();

string queryString = $"SELECT ID,GroupName FROM Groups WHERE ID='{check.SearchElementID(GroupComboBox.Text)}'";

SqlCommand command = new SqlCommand(queryString, dataBase.getConnection());

SqlDataReader sqlDataReader = command.ExecuteReader();

while (sqlDataReader.Read())

{

baseField.GroupId = sqlDataReader.GetInt32(0);

}

dataBase.closeConnection();

CheckComboBoxes();

StudentComboBoxRefresh();

CheckGroupComboBox();

}

private void Window\_Loaded(object sender, RoutedEventArgs e)

{

UserName.Content = $"ПОЛЬЗОВАТЕЛЬ: {User.ShowUserLogin}";

GroupComboBoxRefresh();

DisciplineComboBoxRefresh();

CheckGroupComboBox();

}

public void StudentComboBoxRefresh()

{

dataBase.openConnection();

string queryString = $"SELECT ID, StudentLastName,StudentFirstName, GroupID FROM Students WHERE GroupID='{baseField.GroupId}'";

SqlCommand command = new SqlCommand(queryString, dataBase.getConnection());

SqlDataReader sqlDataReader = command.ExecuteReader();

StudentComboBox.Items.Clear();

while (sqlDataReader.Read())

{

int Id = sqlDataReader.GetInt32(0);

string LastName = sqlDataReader.GetString(1);

string FirstName = sqlDataReader.GetString(2);

StudentComboBox.Items.Add($"{Id}: {LastName} {FirstName}");

}

dataBase.closeConnection();

}

private void StudentComboBox\_DropDownClosed(object sender, EventArgs e)

{

dataBase.openConnection();

string queryString = $"SELECT ID FROM Students WHERE ID='{check.SearchElementID(StudentComboBox.Text)}'";

SqlCommand command = new SqlCommand(queryString, dataBase.getConnection());

SqlDataReader sqlDataReader = command.ExecuteReader();

while (sqlDataReader.Read())

{

baseField.StudentId = sqlDataReader.GetInt32(0);

}

dataBase.closeConnection();

CheckComboBoxes();

}

public void DisciplineComboBoxRefresh()

{

dataBase.openConnection();

string queryString = $"SELECT ID, DisciplineName FROM Disciplines";

SqlCommand command = new SqlCommand(queryString, dataBase.getConnection());

SqlDataReader sqlDataReader = command.ExecuteReader();

DisciplineComboBox.Items.Clear();

while (sqlDataReader.Read())

{

int Id = sqlDataReader.GetInt32(0);

string DisciplineName = sqlDataReader.GetString(1);

DisciplineComboBox.Items.Add($"{Id}: {DisciplineName}");

}

dataBase.closeConnection();

}

private void DisciplineComboBox\_DropDownClosed(object sender, EventArgs e)

{

dataBase.openConnection();

string queryString = $"SELECT ID FROM Disciplines WHERE ID='{check.SearchElementID(DisciplineComboBox.Text)}'";

SqlCommand command = new SqlCommand(queryString, dataBase.getConnection());

SqlDataReader sqlDataReader = command.ExecuteReader();

while (sqlDataReader.Read())

{

baseField.DisciplineId = sqlDataReader.GetInt32(0);

}

dataBase.closeConnection();

CheckComboBoxes();

}

public void CheckGroupComboBox()

{

if (baseField.GroupId==0)

{

StudentComboBox.IsEnabled = false;

}

else

{

StudentComboBox.IsEnabled = true;

}

}

private void AddButton\_Click(object sender, RoutedEventArgs e)

{

AddChoiceWindow addChoiceWindow = new AddChoiceWindow();

addChoiceWindow.Show();

this.Close();

}

private void EditButton\_Click(object sender, RoutedEventArgs e)

{

EditChoiceWindow editChoiceWindow = new EditChoiceWindow();

editChoiceWindow.Show();

this.Close();

}

private void DeleteButton\_Click(object sender, RoutedEventArgs e)

{

DeleteChoiceWindow deleteChoiceWindow = new DeleteChoiceWindow();

deleteChoiceWindow.Show();

this.Close();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using WPF\_StudentsAchievement\_Project.Resources.MVVM.ViewModel;

using WPF\_StudentsAchievement\_Project.Resources.Checks;

namespace WPF\_StudentsAchievement\_Project.Resources.MVVM.View

{

public partial class AddChoiceWindow : Window

{

public AddChoiceWindow()

{

InitializeComponent();

}

Check check= new Check();

private void ExitButton\_Click(object sender, RoutedEventArgs e)

{

Close();

}

private void HideButton\_Click(object sender, RoutedEventArgs e)

{

this.WindowState = WindowState.Minimized;

}

private void Window\_Loaded(object sender, RoutedEventArgs e)

{

//ActionTextBlock.Text = $"ДЕЙСТВИЕ: {check.WhatAction(ActionId)}";

}

private void MovingWindow(object sender, RoutedEventArgs e)

{

if (Mouse.LeftButton == MouseButtonState.Pressed)

{

this.DragMove();

}

}

private void BackButton\_Click(object sender, RoutedEventArgs e)

{

WorkWindow workWindow = new WorkWindow();

this.Close();

workWindow.Show();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WPF\_StudentsAchievement\_Project.Resources.Checks;

using WPF\_StudentsAchievement\_Project.Resources.Profiles;

using System.Data;

using System.Data.SqlClient;

namespace WPF\_StudentsAchievement\_Project.Resources.MVVM.View

{

public partial class AddDisciplineView : UserControl

{

DataBase dataBase = new DataBase();

User user = new User();

DataBaseField baseField = new DataBaseField();

Check check = new Check();

public AddDisciplineView()

{

InitializeComponent();

}

private void ReadyButton\_Click(object sender, RoutedEventArgs e)

{

if (baseField.DisciplineName.Length!=0)

{

if (check.Discipline(baseField.DisciplineName) == false)

{

try

{

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string queryString = $"INSERT INTO [Disciplines]([DisciplineName]) VALUES ('{baseField.DisciplineName}')";

SqlCommand command = new SqlCommand(queryString, dataBase.getConnection());

dataBase.openConnection();

if (command.ExecuteNonQuery() == 1)

{

MessageBox.Show($"Дисциплина добавлена!", "УСПЕШНО", MessageBoxButton.OK, MessageBoxImage.Information);

}

else

{

MessageBox.Show("ОШИБКА ДОБАВЛЕНИЯ ДИСИПЛИНЫ:", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

dataBase.closeConnection();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

else

{

MessageBox.Show("Заполните поля!", "ОШИБКА", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void DisciplineTextBox\_TextChanged(object sender, TextChangedEventArgs e)

{

baseField.DisciplineName = DisciplineTextBox.Text;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WPF\_StudentsAchievement\_Project.Resources.Checks;

using WPF\_StudentsAchievement\_Project.Resources.Profiles;

using System.Data;

using System.Data.SqlClient;

namespace WPF\_StudentsAchievement\_Project.Resources.MVVM.View

{

public partial class AddGradeView : UserControl

{

DataBase dataBase = new DataBase();

User user = new User();

DataBaseField baseField = new DataBaseField();

Check check = new Check();

public AddGradeView()

{

InitializeComponent();

}

private void GradeViewControl\_Loaded(object sender, RoutedEventArgs e)

{

DisciplineComboBoxRefresh();

GroupComboBoxRefresh();

CheckGroupComboBox();

StudentComboBoxRefresh();

}

public void DisciplineComboBoxRefresh()

{

dataBase.openConnection();

string queryString = $"SELECT ID, DisciplineName FROM Disciplines";

SqlCommand command = new SqlCommand(queryString, dataBase.getConnection());

SqlDataReader sqlDataReader = command.ExecuteReader();

DisciplineComboBox.Items.Clear();

while (sqlDataReader.Read())

{

int Id = sqlDataReader.GetInt32(0);

string DisciplineName = sqlDataReader.GetString(1);

DisciplineComboBox.Items.Add($"{Id}: {DisciplineName}");

}

dataBase.closeConnection();

}

public void StudentComboBoxRefresh()

{

dataBase.openConnection();

string queryString = $"SELECT ID, StudentLastName,StudentFirstName, GroupID FROM Students WHERE GroupID='{baseField.GroupId}'";

SqlCommand command = new SqlCommand(queryString, dataBase.getConnection());

SqlDataReader sqlDataReader = command.ExecuteReader();

StudentComboBox.Items.Clear();

while (sqlDataReader.Read())

{

int Id = sqlDataReader.GetInt32(0);

string LastName = sqlDataReader.GetString(1);

string FirstName = sqlDataReader.GetString(2);

StudentComboBox.Items.Add($"{Id}: {LastName} {FirstName}");

}

dataBase.closeConnection();

}

public void GroupComboBoxRefresh()

{

dataBase.openConnection();

string queryString = "SELECT ID,GroupName FROM Groups";

SqlCommand command = new SqlCommand(queryString, dataBase.getConnection());

SqlDataReader sqlDataReader = command.ExecuteReader();

GroupComboBox.Items.Clear();

while (sqlDataReader.Read())

{

int GroupId = sqlDataReader.GetInt32(0);

string GroupName = sqlDataReader.GetString(1);

GroupComboBox.Items.Add($"{GroupId}: {GroupName}");

}

dataBase.closeConnection();

}

public void CheckGroupComboBox()

{

if (baseField.GroupId == 0)

{

StudentComboBox.IsEnabled = false;

}

else

{

StudentComboBox.IsEnabled = true;

}

}

private void DisciplineComboBox\_DropDownClosed(object sender, EventArgs e)

{

baseField.DisciplineId = Convert.ToInt32(check.SearchElementID(DisciplineComboBox.Text));

}

private void GroupComboBox\_DropDownClosed(object sender, EventArgs e)

{

baseField.GroupId = Convert.ToInt32(check.SearchElementID(GroupComboBox.Text));

StudentComboBoxRefresh();

CheckGroupComboBox();

}

private void StudentComboBox\_DropDownClosed(object sender, EventArgs e)

{

baseField.StudentId = Convert.ToInt32(check.SearchElementID(StudentComboBox.Text));

}

private void GradeComboBox\_DropDownClosed(object sender, EventArgs e)

{

baseField.Grade = Convert.ToInt32(GradeComboBox.Text);

}

private void DatePick\_CalendarClosed(object sender, RoutedEventArgs e)

{

try

{

baseField.Date = DatePick.SelectedDate.Value;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void ReadyButton\_Click(object sender, RoutedEventArgs e)

{

if (baseField.StudentId!=0 & baseField.DisciplineId!=0 & baseField.Grade!=0 & baseField.Date.Day !=0 & baseField.Date.Month!=0)

{

try

{

string queryString = $"INSERT INTO [Grades]([StudentID],[DisciplineID],[Grade],[Date]) VALUES ({baseField.StudentId},{baseField.DisciplineId},{baseField.Grade},'{baseField.Date.Year}-{baseField.Date.Month}-{baseField.Date.Day}')";

SqlCommand command = new SqlCommand(queryString, dataBase.getConnection());

dataBase.openConnection();

if (command.ExecuteNonQuery() == 1)

{

MessageBox.Show($"Оценка выставлена!", "УСПЕШНО", MessageBoxButton.OK, MessageBoxImage.Information);

}

else

{

MessageBox.Show("ОШИБКА ВЫСТАВЛЕНИЯ ОЦЕНКИ:", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

dataBase.closeConnection();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

else

{

MessageBox.Show("Заполните поля!", "ОШИБКА", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WPF\_StudentsAchievement\_Project.Resources.Checks;

using WPF\_StudentsAchievement\_Project.Resources.Profiles;

using System.Data;

using System.Data.SqlClient;

namespace WPF\_StudentsAchievement\_Project.Resources.MVVM.View

{

public partial class AddGroupView : UserControl

{

DataBase dataBase = new DataBase();

User user = new User();

DataBaseField baseField = new DataBaseField();

Check check = new Check();

public AddGroupView()

{

InitializeComponent();

}

private void ReadyButton\_Click(object sender, RoutedEventArgs e)

{

if (baseField.GroupName.Length != 0 & baseField.Course != 0)

{

if (check.Group(baseField.GroupName) == false)

{

try

{

string queryString = $"INSERT INTO [Groups]([GroupName],[Course]) VALUES ('{baseField.GroupName}',{baseField.Course})";

SqlCommand command = new SqlCommand(queryString, dataBase.getConnection());

dataBase.openConnection();

if (command.ExecuteNonQuery() == 1)

{

MessageBox.Show($"Группа добавлена!", "УСПЕШНО", MessageBoxButton.OK, MessageBoxImage.Information);

}

else

{

MessageBox.Show("ОШИБКА ДОБАВЛЕНИЯ ГРУППЫ:", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

dataBase.closeConnection();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

else

{

MessageBox.Show("Заполните поля!", "ОШИБКА", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void GroupTextBox\_TextChanged(object sender, TextChangedEventArgs e)

{

baseField.GroupName = GroupTextBox.Text;

}

private void CourseComboBox\_DropDownClosed(object sender, EventArgs e)

{

try

{

baseField.Course = Convert.ToInt32(CourseComboBox.Text);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using System.Data;

using System.Data.SqlClient;

using WPF\_StudentsAchievement\_Project.Resources.Profiles;

using WPF\_StudentsAchievement\_Project.Resources.Checks;

namespace WPF\_StudentsAchievement\_Project.Resources.MVVM.View

{

public partial class AddStudentView : UserControl

{

DataBase dataBase = new DataBase();

User user = new User();

DataBaseField baseField = new DataBaseField();

Check check = new Check();

public AddStudentView()

{

InitializeComponent();

}

private void LastNameTextBox\_TextChanged(object sender, TextChangedEventArgs e)

{

baseField.StudentLastName = LastNameTextBox.Text;

}

private void FirstNameTextBox\_TextChanged(object sender, TextChangedEventArgs e)

{

baseField.StudentFirstName = FirstNameTextBox.Text;

}

private void MiddleNameTextBox\_TextChanged(object sender, TextChangedEventArgs e)

{

baseField.StudentMiddleName = MiddleNameTextBox.Text;

}

public void GroupComboBoxRefresh()

{

dataBase.openConnection();

string queryString = "SELECT ID,GroupName FROM Groups";

SqlCommand command = new SqlCommand(queryString, dataBase.getConnection());

SqlDataReader sqlDataReader = command.ExecuteReader();

GroupComboBox.Items.Clear();

while (sqlDataReader.Read())

{

int GroupId = sqlDataReader.GetInt32(0);

string GroupName = sqlDataReader.GetString(1);

GroupComboBox.Items.Add($"{GroupId}: {GroupName}");

}

dataBase.closeConnection();

}

private void GroupComboBox\_DropDownClosed(object sender, EventArgs e)

{

baseField.GroupId = Convert.ToInt32(check.SearchElementID(GroupComboBox.Text));

}

private void ReadyButton\_Click(object sender, RoutedEventArgs e)

{

if (baseField.StudentFirstName.Length != 0 & baseField.StudentLastName.Length != 0 & baseField.StudentMiddleName.Length != 0 & baseField.GroupId != 0)

{

if (check.Student(baseField.StudentLastName,baseField.StudentFirstName,baseField.StudentMiddleName) == false)

{

try

{

string queryString = $"INSERT INTO [Students]([StudentLastName],[StudentFirstName],[StudentMiddleName],[GroupID]) VALUES ('{baseField.StudentLastName}','{baseField.StudentFirstName}','{baseField.StudentMiddleName}',{baseField.GroupId})";

SqlCommand command = new SqlCommand(queryString, dataBase.getConnection());

dataBase.openConnection();

if (command.ExecuteNonQuery() == 1)

{

MessageBox.Show($"Студент добавлен!", "УСПЕШНО", MessageBoxButton.OK, MessageBoxImage.Information);

}

else

{

MessageBox.Show("ОШИБКА ДОБАВЛЕНИЯ СТУДЕНТА:", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

dataBase.closeConnection();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

else

{

MessageBox.Show("Заполните поля!", "ОШИБКА", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void UserControl\_Loaded(object sender, RoutedEventArgs e)

{

GroupComboBoxRefresh();

}

}

}