Государственное бюджетное профессиональное образовательное учреждение

Самарской области

«Тольяттинский социально-экономический колледж»

**КУРСОВАЯ РАБОТА**

**РАЗРАБОТКА И ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «УЧЁТ ОПЛАТЫ ЗА ОБУЧЕНИЕ В ХУДОЖЕСТВЕННОЙ ШКОЛЕ»**

**ПМ.05 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ**

**МДК 05.02 Разработка кода информационных систем**

**09.02.07 Информационные системы и программирование**

|  |  |  |  |
| --- | --- | --- | --- |
| Студент |  | **/** | О.А.Федорова |
|  | *подпись* |  | *И.О. Фамилия* |
| \_\_.\_\_.2022 г. |  |  |  |
| Оценка выполнения и защиты курсовой работы | | |  |
|  | | |  |
| Руководитель |  | **/** | Е.В. Плюснина |
|  | *подпись* |  | *И.О. Фамилия* |
| \_\_.\_\_.2022 г. |  |  |  |

Тольятти, 2022

Государственное бюджетное профессиональное образовательное учреждение  
Самарской области  
«Тольяттинский социально-экономический колледж»

|  |  |  |
| --- | --- | --- |
|  |  | Утверждаю:  Заместитель директора по УР  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.С. Киронова  *« » 202 г.* |

**ЗАДАНИЕ**

на курсовую работу

**по ПМ.05 Проектирование и разработка информационных систем модуля, выполняемой в рамках МДК.05.02 Разработка кода информационных систем**

студента группы ИСП-32

Фёдоровой Ольги Андреевны

*Фамилия Имя отчество студента*

Тема курсовой работы: «Разработка информационной системы **«**Учёт оплаты за обучение»

1. Содержание задания:

1.1 Разработать физическую структуру информационной системы.

1.2 Разработать запросы к информационной системе.

1.3 Разработать приложение.

1. Исходные данные:

Исходные данные для практической реализации автоматизированной информационной системы (АИС) берутся из различных информационных источников (Интернет-ресурсы, печатные издания, периодика и др.).

1. Содержание курсовой работы

Введение

1 Разработка базы данных информационной системы (Учёт оплаты за обучение)

1.1 Разработка физической структуры БД

1.2 Разработка запросов ИС

2 Разработка интерфейса информационной системы (Учёт оплаты за обучение)

Заключение

Список использованных источников

Приложения

Приложение А Листинг программы

Дата выдачи задания: «13» января 2022 г.

Дата сдачи работы на отделение: «\_\_»\_\_\_\_\_\_\_\_2022 г.

Руководитель курсового(ой) проекта(работы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Е.В. Плюснина

подпись расшифровка подписи

**КАЛЕНДАРНЫЙ ПЛАН**

выполнения курсовой работы

Студентом 3 курса группы ИСП-32

По теме Разработка и проектирование ИС «Учёт оплаты за обучение»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  этапа  работы | Содержание этапов работы | Плановый срок выполнения этапа | Планируемый объем выполнения  этапа, % | Отметка  о  выполнении  этапа |
| 1 | Выбор, обоснование темы и объекта исследования | Январь 2022 | 5% |  |
| 2 | Утверждение темы, согласование плана. Введение, библиография | Январь 2022 | 10% |  |
| 3 | Изучение и анализ информационных материалов по теме | Февраль 2022 | 15% |  |
| 4 | Обоснование актуальности выбранной темы применительно к профессиональной деятельности (введение) | Февраль 2022 | 20% |  |
| 5 | Изложение материала основной части по теме курсовой работы | Февраль 2022 | 20% |  |
| 6 | Подведение итогов проведенного анализа, формулировка выводов УИР применительно к профессиональной деятельности (заключение) | Март 2022 | 20% |  |
| 7 | Оформление работы и сдача на проверку | Март 2022 | 10% |  |
| 8 | Защита работы |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Студент |  | **/** | О.А. Федорова |
|  | *подпись* |  | *И.О. Фамилия* |
| \_\_.\_\_.2022 г. |  |  |  |
| Руководитель |  | **/** | Е.В. Плюснина |
|  | *подпись* |  | *И.О. Фамилия* |
| \_\_.\_\_.2022 г. |  |  |  |

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 6](#_Toc101606519)

[1 РАЗРАБОТКА БАЗЫ ДАННЫХ ИНФОРМАЦИОННОЙ СИСТЕМЫ РАСЧЕТ СТОИМОСТИ УСЛУГ, ПРЕДОСТАВЛЯЕМЫХ КЛИЕНТАМ 7](#_Toc101606530)

[1.1 Разработка физической структуры БД 7](#_Toc101606531)

[Создание базы данных в SQL Server Management Studio 7](#_Toc101606532)

[1.2 Разработка запросов ИС 10](#_Toc101606533)

[2.1 Разработка интерфейса 12](#_Toc101606534)

[ЗАКЛЮЧЕНИЕ 25](#_Toc101606535)

[1. Руководство. создание первого приложения WPF в Visual Studio 2019 25](#_Toc101606536)

[2. Кормышев В.М. Исследование систем управления при проектировании информационных систем: учебное пособие. 25](#_Toc101606537)

[3. Кормышев В.М. Управление знаниями в информационных системах. 25](#_Toc101606538)

[ПРИЛОЖЕНИЕ А 26](#_Toc101606539)

[ЛИСТИНГ ПРОГРАММЫ 26](#_Toc101606540)

# ВВЕДЕНИЕ

# В настоящее время вычислительная техника постоянно развивается высокими темпами. Мы просто-напросто теперь не можем обойтись без компьютеров, потому что они позволяют автоматизировать нашу жизнь, то есть редактировать документы, проектировать чертежи и так далее. Также не стоит упускать развлекательный фактор, связанный с просмотром фильмов, прослушиванием музыки, прохождением компьютерных игр и многими другими занятиями.

Актуальность исследования.Учёт оплаты за обучение в художественной школе — магазин, торгующий компьютерными комплектующие. Позволит облегчить работу сотрудникам и повысить их эффективность.

# Предмет исследования. – информационная система «Учёт оплаты за обучение»

# Проблема исследования. При изучении предметной области, выяснилось, что информационная система «Учёт оплаты за обучение» на первый взгляд простая, но для ее реализации необходимо обратится к некоторым разделам, таким как создание базы данных и создание приложения для взаимодействия с этой базой данных.

# Цель исследования Целью курсовой работы является создание информационной системы «Учёт оплаты за обучение». Для достижения поставленной цели были определены следующие задачи:

# Описать предметную область;

# Спроектировать программный комплекс;

# Выполнить программную реализацию;

# Проанализировать качество программного обеспечения.

# Практическая значимость исследования. Практическая значимость заключается в использовании разработанной системы на различных предприятиях по «Учёт оплаты за обучение».

# Структура работы. Структура работы соответствует логике исследования и включает в себя введение, две главы, заключение, список использованной литературы, приложение.

# 1 РАЗРАБОТКА БАЗЫ ДАННЫХ ИНФОРМАЦИОННОЙ СИСТЕМЫ РАСЧЕТ СТОИМОСТИ УСЛУГ, ПРЕДОСТАВЛЯЕМЫХ КЛИЕНТАМ

## 1.1 Разработка физической структуры БД

## Создание базы данных в SQL Server Management Studio

Первое, что Вам нужно сделать, это запустить среду SQL Server Management Studio и подключиться к SQL серверу.

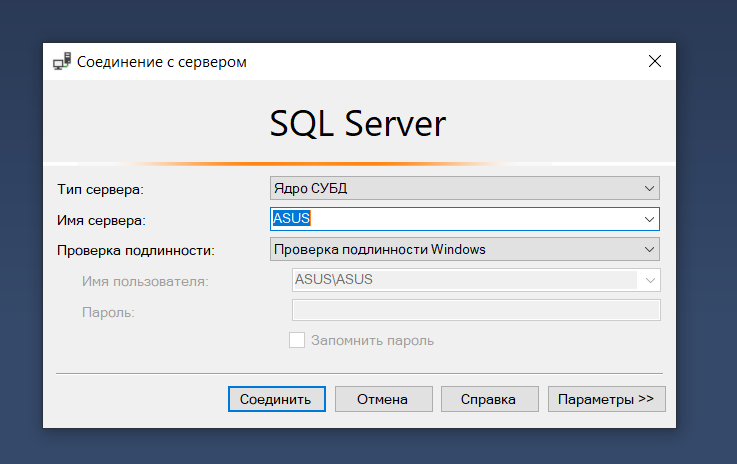


Рисунок 1- Окно подключения

Затем в обозревателе объектов щелкнуть по контейнеру «Базы данных» правой кнопкой мыши и выбрать пункт «Создать базу данных»*.*

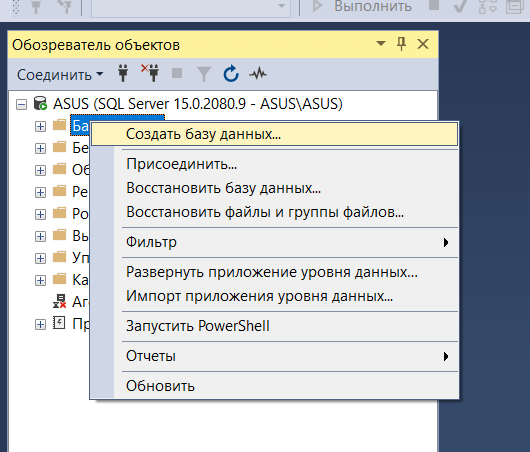


Рисунок 2 – Окно Object Explorer

В обозревателе объектов открываем контейнер «Базы данных», затем открываем нужную базу данных и щелкаем правой кнопкой мыши по пункту «Таблицы», и выбираем «Таблица».

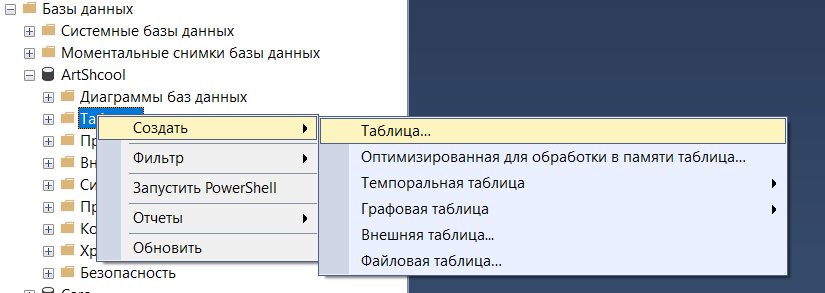


Рисунок 3 – Окно Object Explorer

Введите имена столбцов, выберите типы данных и определите для каждого столбца, могут ли в нем присутствовать значения NULL, как показано на следующей иллюстрации:

Таблица создается в рамках текущей базы данных.

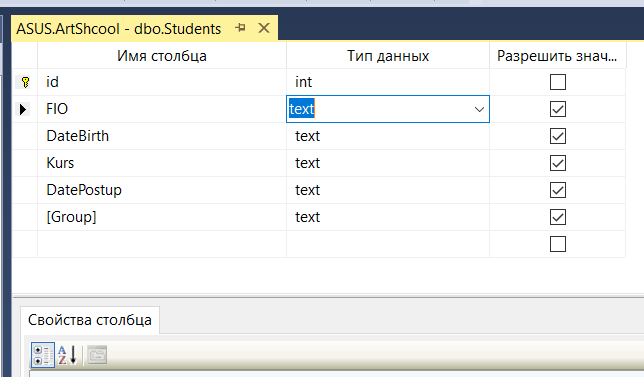


Рисунок 4 – Таблица «Учащиеся»

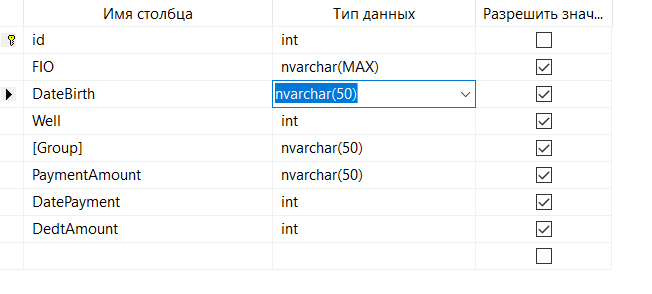
**

Рисунок 5 - Таблица «Отчёт»

1. Также можете задать другие свойства столбца, например является ли этот столбец столбцом идентификаторов или вычисляемым столбцом. Для этого щелкните столбец на вкладке свойств столбцов. Дополнительные сведения о свойствах столбцов см. в разделе Свойства столбца таблицы (среда SQL Server Management Studio).

2. Чтобы указать, что столбец является столбцом первичного ключа, щелкните его правой кнопкой мыши и выберите. Задать первичный ключ.

3. Чтобы создать связи по внешнему ключу, проверочные ограничения или индексы, щелкните правой кнопкой мыши панель конструктора таблиц и выберите в списке объект, как показано на следующей иллюстрации:

1. В меню Файл выберите команду Сохранить имя таблицы.

2. В диалоговом окне Выбор имени введите имя таблицы и нажмите кнопку OK.

3. Чтобы просмотреть новую таблицу, в обозревателе объектов разверните узел Таблицы, а затем нажмите клавишу F5, чтобы обновить список объектов. Новая таблица будет отображена в списке таблиц*.*

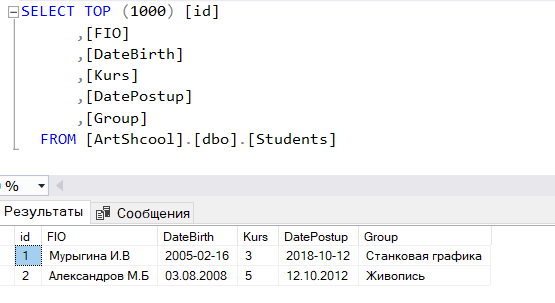
**

Рисунок 6 - Запрос «Учащиеся»

## 1.2 Разработка запросов ИС

Все операции, которые вы можете выполнять с данными, следуют аббревиатуре CRUD.

CRUD обозначает 4 основные операции, которые мы выполняем при запросе базы данных: Create (создание), Update (обновление) и Delete (удаление).

Delete (удаление).

USE [ArtShcool]

GO

DELETE FROM [dbo].[Учащиеся]

WHERE <Условия поиска,,>

GO

**U**pdate (обновление)

USE [ArtShcool]

GO

UPDATE [dbo].[Учащиеся]

SET [ФИО] = <ФИО, nvarchar(50),>

,[Дата рождения] = < Дата рождения, nvarchar(50),>

,[Курс] = <Курс, nvarchar(50),>

,[Группа] = <Группа, int,>

WHERE <Условия поиска,,>

GO

**C**reate (создание)

USE [ArtShcool]

GO

Object: Table [dbo].[Отчёт]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Отчёт](

[ФИО] [nvarchar](50) NULL,

[Дата рождения] [nvarchar](50) NULL,

[Курс] [nvarchar](50) NULL,

[Группа] [int] NULL

) ON [PRIMARY]

GO

Для получения данных применяется команда SELECT. В упрощенном виде она имеет следующий синтаксис:

SELECT список\_столбцов FROM имя\_таблицы

Для изменения уже имеющихся строк в таблице применяется команда UPDATE. Она имеет следующий формальный синтаксис:

UPDATE имя\_таблицы

SET столбец1 = значение1, столбец2 = значение2, ... столбецN = значениеN

[FROM выборка AS псевдоним\_выборки]

[WHERE условие\_обновления]

Для удаления применяется команда DELETE:

DELETE [FROM] имя\_таблицы

WHERE условие\_удаления

**2. РАЗРАБОТКА ИНТЕРФЕЙСА ИНФОРМАЦИОННОЙ СИСТЕМЫ «Учёт оплаты за обучение»**

## 2.1 Разработка интерфейса

Visual Studio использует *проекты*, чтобы упорядочить код для приложения, и *решения*, чтобы упорядочить проекты. Проект содержит все параметры, конфигурации и правила, используемые для сборки приложения. Он управляет связью между всеми файлами проекта и любыми внешними файлами. Чтобы создать приложение, сначала создайте проект и решение.

Порядок загрузки данных и проверка работоспособности

Порядок загрузки:

1. Открыть программу «Visual Studio»;
2. Выбрать пункт «запустить…»;

В Visual Studio откройте меню "Файл" и выберите команду "Создать > Project", чтобы открыть диалоговое окно "Создание нового Project". Выберите шаблон Консольное приложение с тегами C# , Windows и Консоль, а затем нажмите кнопку Далее.

1. В открывшемся окне выбрать пункт «Создание проекта»

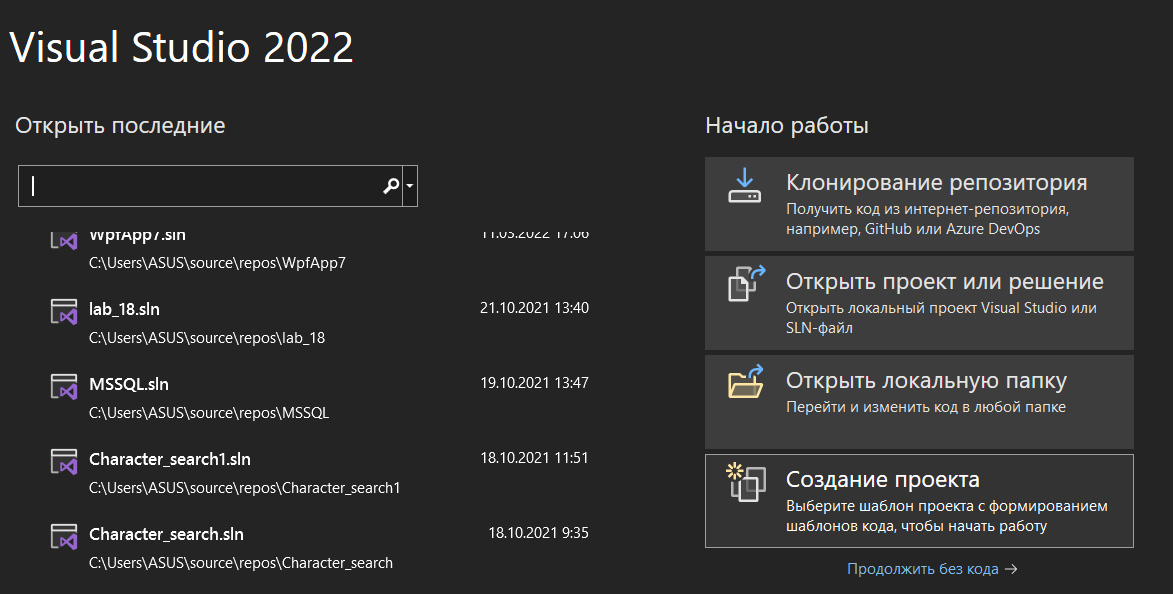


Рисунок 2.1 – Создания проекта

1. Вводим в поиске WPF и ищем приложение WPF (.Net.Framework) создаём проект и нажимаем далее

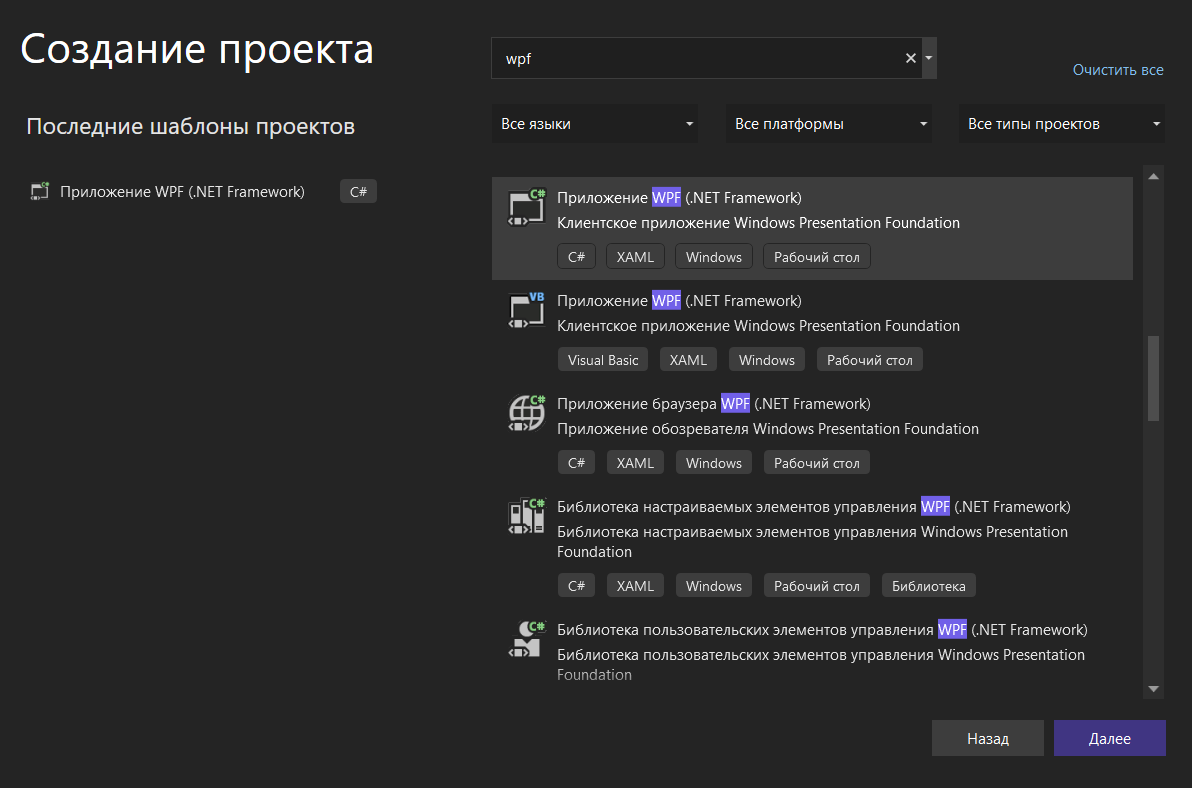


Рисунок 2.2 - приложение «wpf»

В диалоговом окне настроить новый проект в поле Имя проекта введите HelloWorld. Выберите Создать, чтобы создать проект.

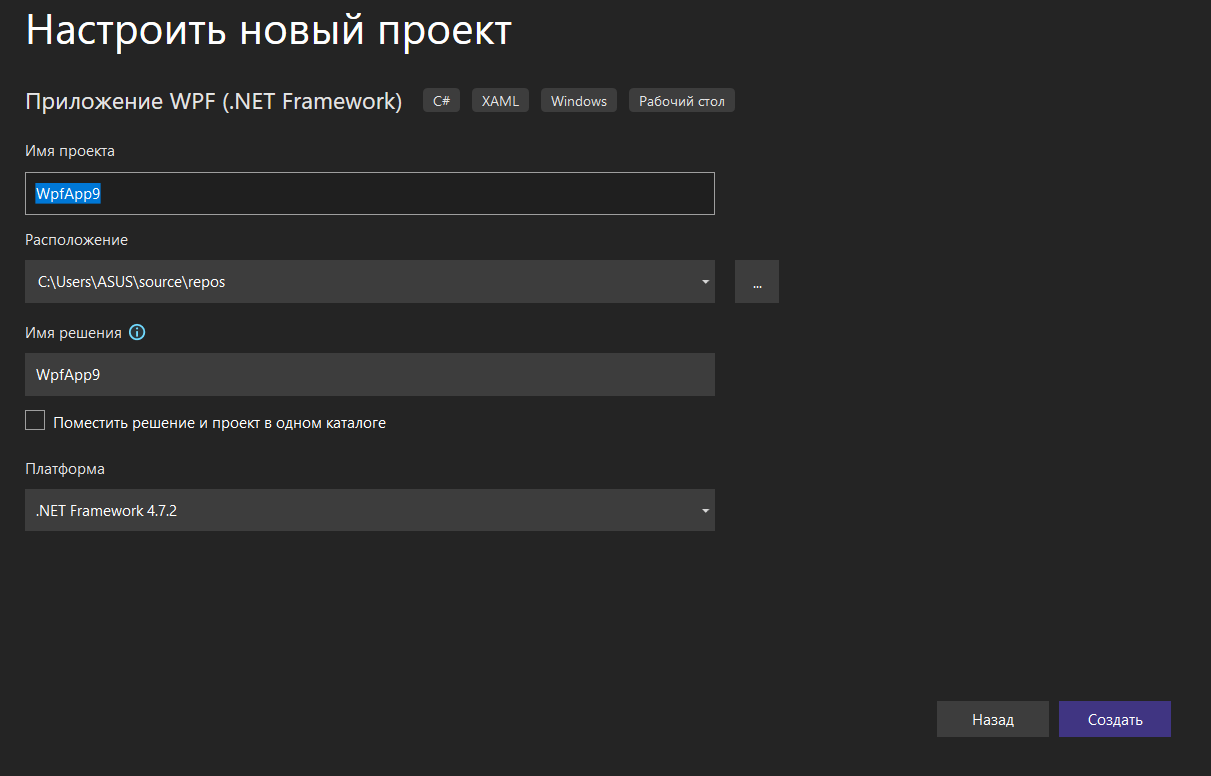


Рисунок 2.3 – Название «АИС»

1. После загрузки проекта, в правом верхнем углу выбираем файл-открыть-решение или проект

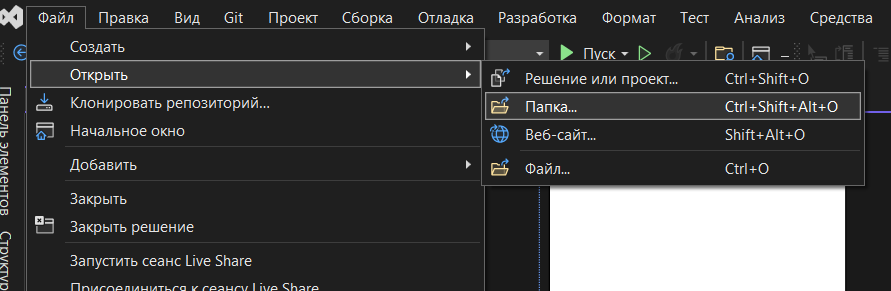


Рисунок 2.4 – Открытие проекта

1. Выбрать путь к выгруженному проекту «Учёт оплаты за обучение» процесс продемонстрирован на рисунке 2.5;

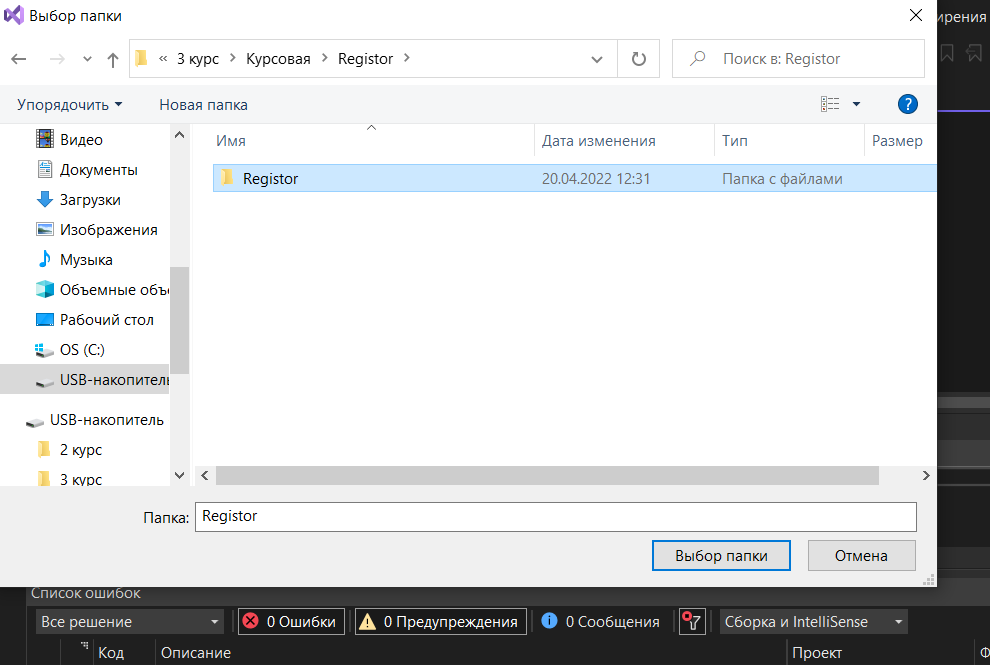


Рисунок 2.5 – Путь к проекту

1. Дождаться окончания загрузки и можно приступать к работе.

откройте Application. xaml (Visual Basic) или App. xaml (C#).

Этот XAML-файл определяет приложение WPF и все ресурсы приложения. Этот файл также используется для указания пользовательского интерфейса, в данном случае MainWindow. *XAML*, который автоматически отображается при запуске приложения.

КОД XAML должен выглядеть следующим образом в Visual Basic:

<Application x:Class="Application"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

StartupUri="MainWindow.xaml">

<Application.Resources>

</Application.Resources>

</Application>

Аналогично следующему в C#:

<Application x:Class="ExpenseIt.App"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

StartupUri="MainWindow.xaml">

<Application.Resources>

</Application.Resources>

</Application>

Откройте файл *MainWindow.xaml*.

Этот XAML-файл является главным окном приложения и отображает содержимое, созданное на страницах. [Window](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.window)Класс определяет свойства окна, такие как заголовок, размер или значок, а также обрабатывает события, такие как закрытие или скрытие.

[Window](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.window)Измените элемент на [NavigationWindow](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.navigation.navigationwindow) , как показано в следующем коде XAML:

<NavigationWindow x:Class="ExpenseIt.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

</NavigationWindow>

Это приложение переходит к другому содержимому в зависимости от введенных пользователем данных. Именно поэтому необходимо изменить основное [Window](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.window) значение на [NavigationWindow](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.navigation.navigationwindow) . [NavigationWindow](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.navigation.navigationwindow) наследует все свойства [Window](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.window) . [NavigationWindow](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.navigation.navigationwindow)Элемент в файле XAML создает экземпляр [NavigationWindow](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.navigation.navigationwindow) класса. Дополнительные сведения см. в разделе [Общие сведения о навигации](https://docs.microsoft.com/ru-ru/dotnet/desktop/wpf/app-development/navigation-overview?view=netframeworkdesktop-4.8).

[Grid](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.grid) Удалите элементы из между [NavigationWindow](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.navigation.navigationwindow) тегами.

Измените следующие свойства в коде XAML для [Navigation Window](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.navigation.navigationwindow) элемента:

[Title](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.window.title) Присвойте свойству значение " ExpenseIt ".

[Height](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.frameworkelement.height) Задайте для свойства значение 350 пикселей.

[Width](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.frameworkelement.width) Задайте для свойства значение 500 пикселей.

КОД XAML должен выглядеть следующим образом для Visual Basic:

XAMLКопировать

<NavigationWindow x:Class="MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="ExpenseIt" Height="350" Width="500">

</NavigationWindow>

И, как и в случае с C#:

<NavigationWindow x:Class="ExpenseIt.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="ExpenseIt" Height="350" Width="500">

</NavigationWindow>

Откройте файл MainWindow*. XAML.* vb или MainWindow*. XAML. CS*.

Этот файл является файлом кода программной части, который содержит код для работы с событиями, объявленными в файле MainWindow*. XAML*. Этот файл содержит разделяемый класс для окна, определенного в XAML-коде.

Если вы используете C#, измените MainWindow класс на производный от [NavigationWindow](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.navigation.navigationwindow) . (в Visual Basic это происходит автоматически при изменении окна в XAML.) Код C# теперь должен выглядеть следующим образом:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

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;

namespace ExpenseIt

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : NavigationWindow

{

public MainWindow()

{

InitializeComponent();

}

}

}

Чтобы выгрузить базу данных в проект нужно:

1. Нажать правой кнопкой на решения проекта, после чего создаем элемент

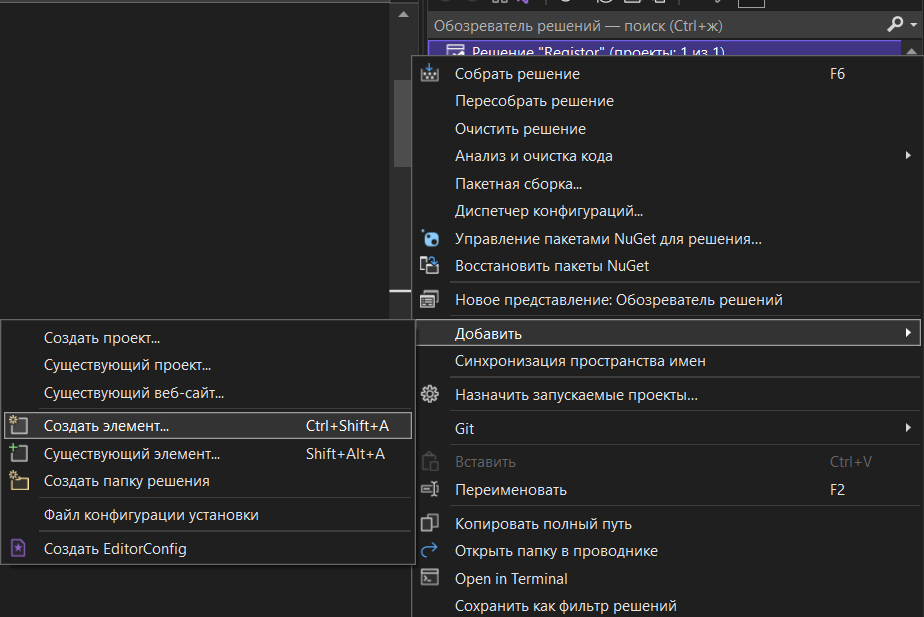


Рисунок 2.6 - добавление базы данных

1. В окне «Добавление нового элемента» выбираем пункт данные. Выбираем модель ADO.NET DEM

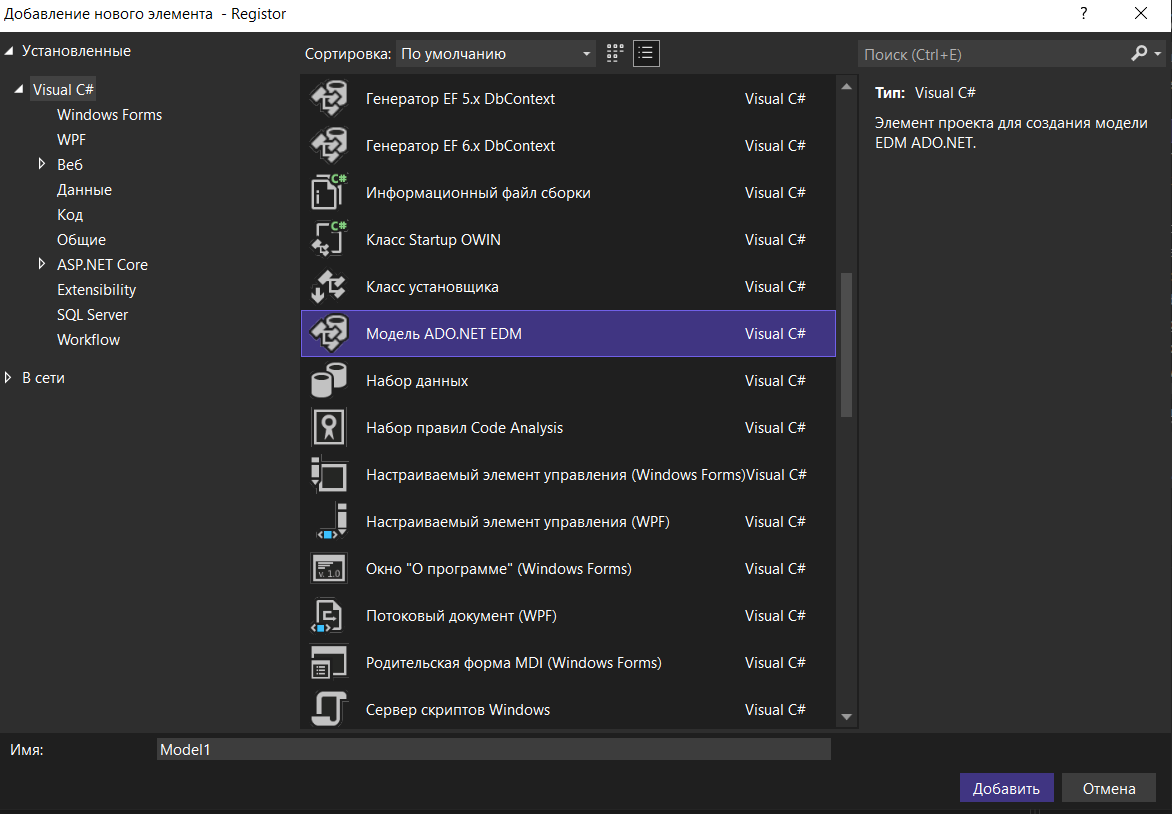


Рисунок 2.7 - добавление базы данных

1. Из содержания модели выбираем Конструктор EF из базы данных, после чего создаем новое соединение

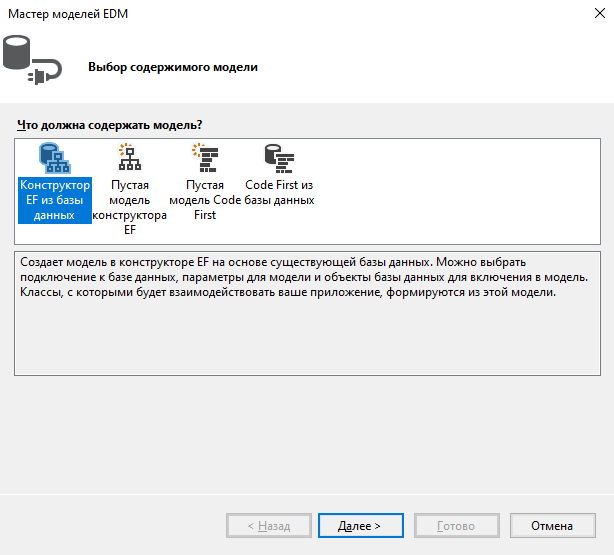


Рисунок 2.8 - добавление базы данных

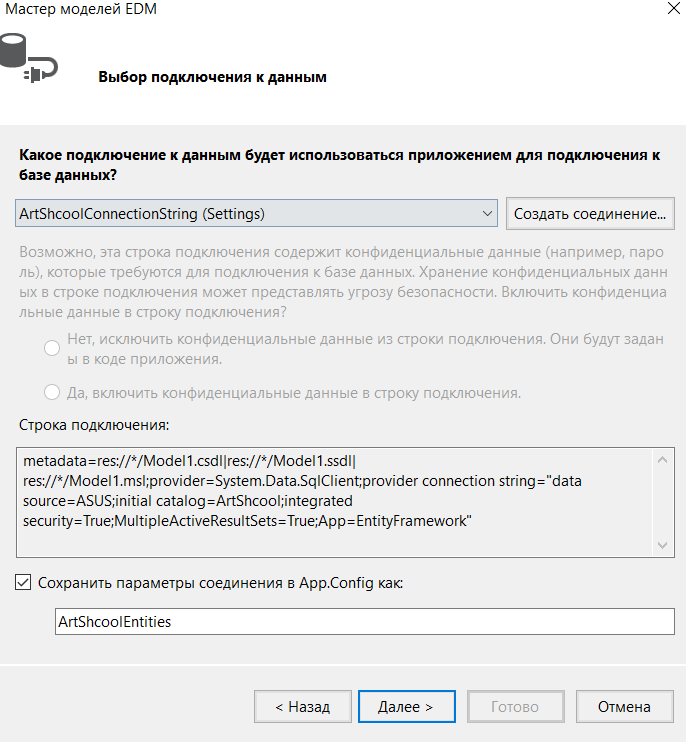


Рисунок 2.9 - добавление базы данных

1. В новом окне пишем имя нашего сервера и выбираем нашу базу данных, ставим галочку на таблице и жмем далее

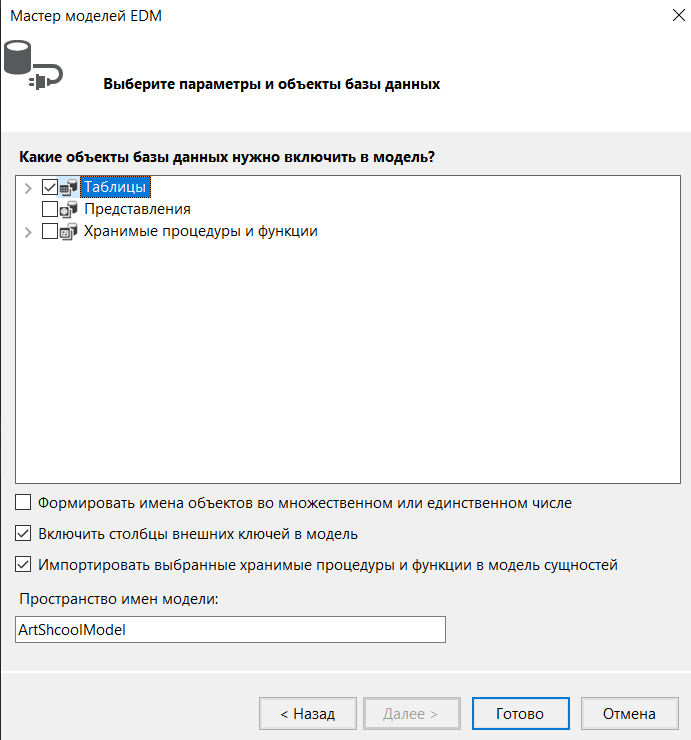


Рисунок 2.10 - добавление базы данных

**2.2 Разработка руководства пользователя**

Данное руководство предназначено для пользователей программного продукта ИС «Учёт оплаты за обучение».

Документ «Руководство Пользователя» предназначен для конечных пользователей продукта, администраторов системы и другого персонала.

Для авторизации введите логин, пароль в открывшемся окне при входе в приложение.

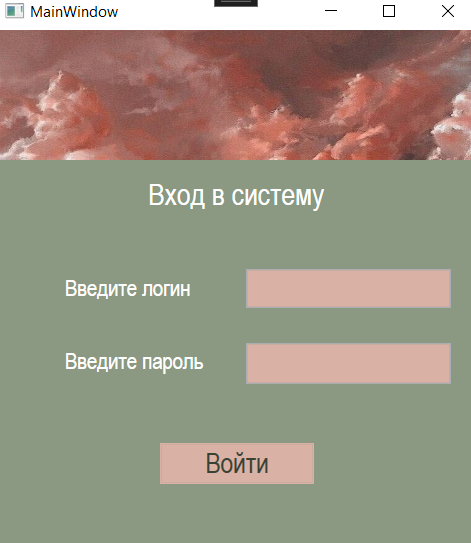


Рисунок 2.1 – Окно «Авторизация»

Диалоговое окно «Пользователь Авторизовался» обозначит успешный вход в приложение. При нажатии кнопки «Ок» в зависимости от того какой пользователь вошёл в систему (Администратор, бухгалтер или Преподаватель) будет открыто соответствующие окно пользователя.

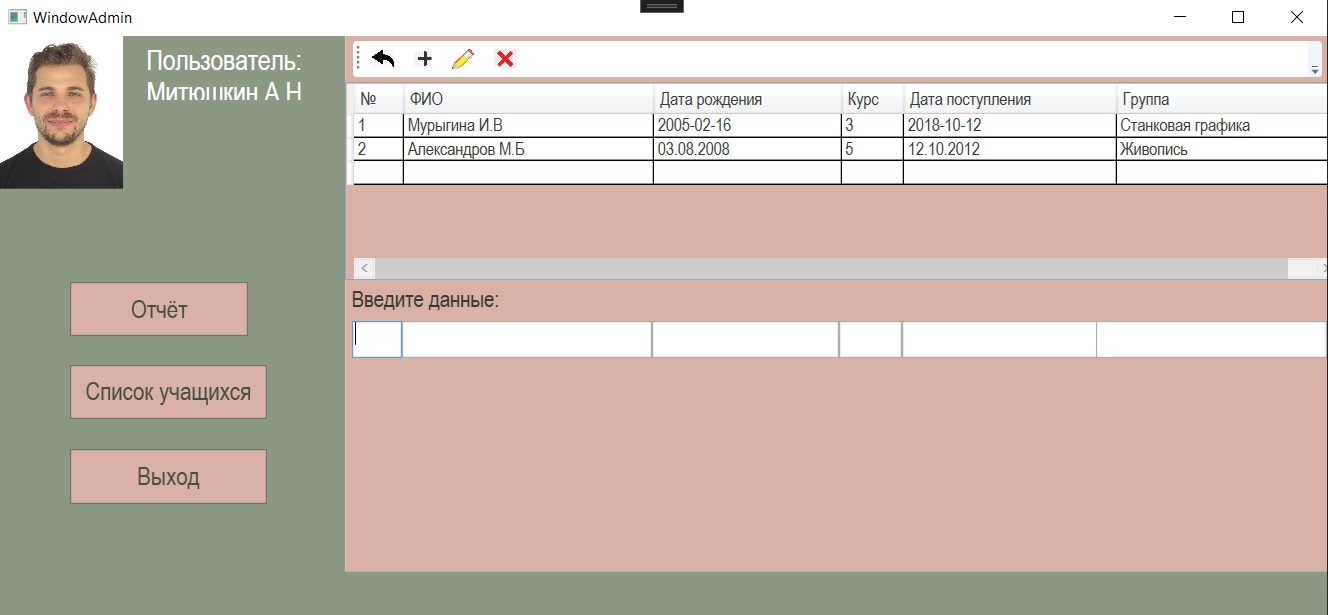


Рисунок 2.2 – Окно пользователя – «Администратор»

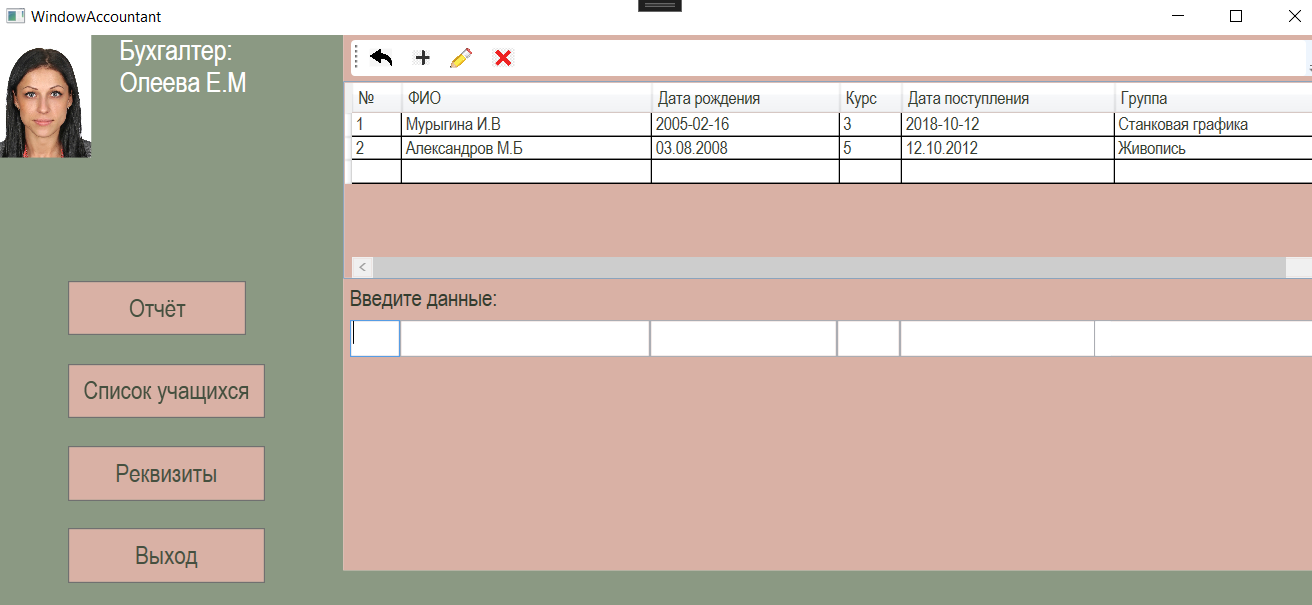


Рисунок 2.3 – Окно пользователя – «Бухгалтер»

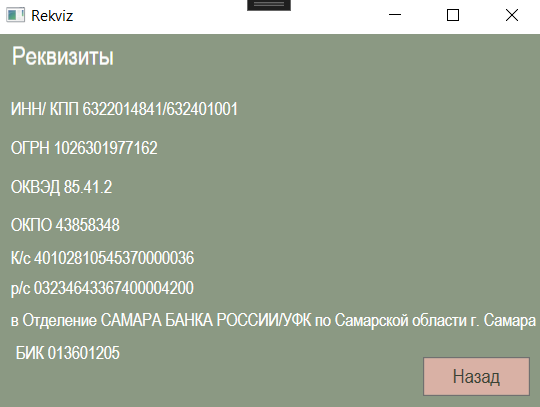


Рисунок 2.4 – Окно «Реквизиты»

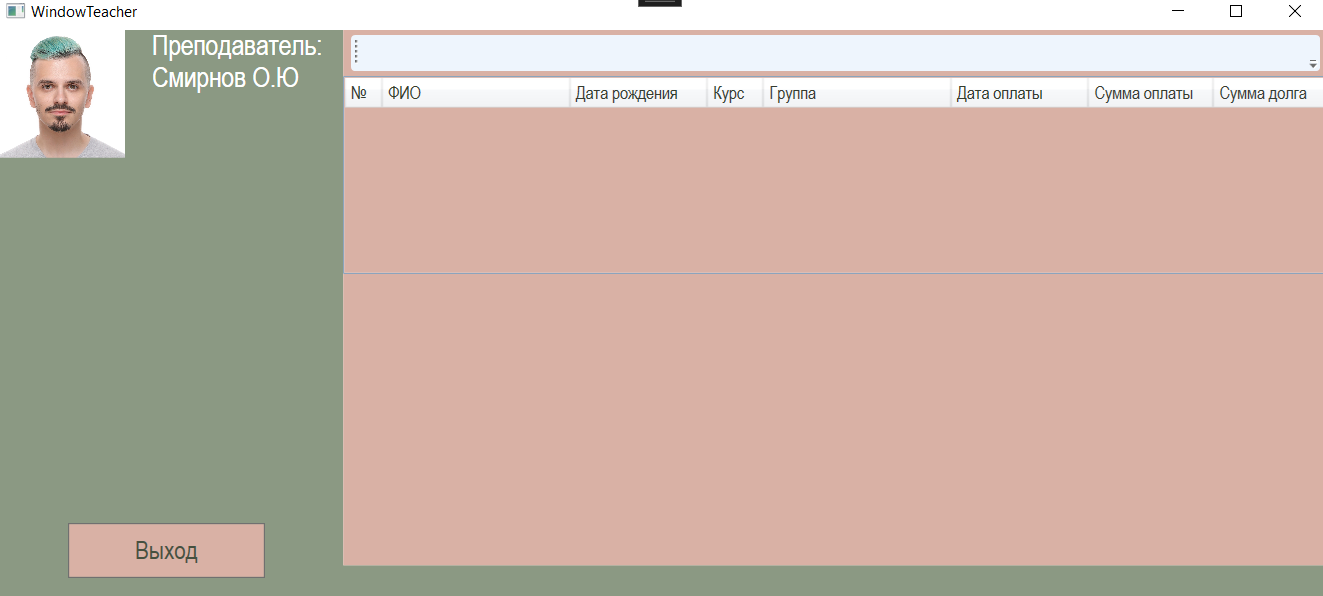


Рисунок 2.5 – Окно пользователя – «Преподаватель»

У пользователей реализованы разные возможности:

Пользователь Администратор может добавлять, изменять и удалять данные из таблиц «Список учащихся» и «Отчёт».

Пользователь Бухгалтер может добавлять, изменять и удалять данные из таблиц «Список учащихся» и «Отчёт», а также имеет доступ к реквизитам.

Пользователь Преподаватель имеет доступ только к просмотру таблицы «Отчёт» без возможности как-либо с ней взаимодействовать.

У всех пользователей есть возможность выйти на окно Авторизации с помощью кнопки «Выйти»

Для установки клиента компьютер конечного пользователя должен удовлетворять следующие требования.

Компьютер должен иметь следующую конфигурацию:

* Intel CPU: все процессоры Intel Core i3/i5/i7/i9-10xxx вплоть до 10 поколения
* оперативную память 4 гигабайт (ГБ) (для 32-разрядной системы) или 8 ГБ (для 64-разрядной системы);
* жесткий диск (при установке используется не менее 120 Мбайт на жестком диске);
* мышь;
* SVGA-совместимый монитор;
* USB порт;
* операционную систему MicrosoftWindows 7-10;
* на компьютере должна быть установлена система «Visual studio», имеющая соответствующую лицензию. Конфигурация предназначена для работы при разрешении экрана не ниже 1024 х 768 и при размере экранного шрифта «Обычный размер».

Для работы с программой необходимо следующее программное обеспечение:

1. SQL Server Management **Studio** (SSMS) — это интегрированная среда для управления любой инфраструктурой SQL, от SQL Server до баз данных. SSMS предоставляет средства для настройки, наблюдения и администрирования экземпляров SQL Server и баз данных

2. Microsoft SQL Server — система управления реляционными базами данных, разработанная корпорацией Microsoft

3. Visual Studio — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений

# ЗАКЛЮЧЕНИЕ

ИС представляет собой распределенное приложение, и состоит из приложения .NET Framework и БД, доступ к которой осуществляется посредством СУБД Microsoft SQL Server.

В курсовой работе проанализирована предметная область и определены требования к ИС и средствам его разработки. Осуществлено проектирование и реализация БД с информацией о клиентах, покупок и корзины, и взаимодействующего с ней приложения .NET Framework.

В курсовой работе было реализован программный продукт ИС «Учёт оплаты за обучение». Основными информационными технологиями, использовавшимися в курсовой работе, являются:

* реляционные БД (Microsoft SQL Server) и технология объектно-реляционного проецирования to SQL;
* программирование на современных языках высокого уровня (Visual C#);
* UML-моделирование;

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

# Руководство. создание первого приложения WPF в Visual Studio 2019

# Кормышев В.М. Исследование систем управления при проектировании информационных систем: учебное пособие.

# Кормышев В.М. Управление знаниями в информационных системах.

1. Информационные системы и технологии Тельнова Ю.Ф.. - М.: Юнити,
2. Информационные технологии и вычислительные системы. Программное обеспечение. Операционные системы.
3. Руководство. SQL Server Management Studio.
4. Руководство. Помощник по настройке ядра СУБД.
5. Microsoft SQL Server 2012. Основы T-SQL.

# ПРИЛОЖЕНИЕ А

# ЛИСТИНГ ПРОГРАММЫ

**Окно Администратора**

<Window x:Class="Registor.WindowAdmin"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d"

Title="WindowAdmin" Height="501" Width="1077">

<Grid Background="#FF8B9983">

<Image Margin="0,0,963,341" Source="/man.jpg" Stretch="UniformToFill"/>

<Frame x:Name="Frame1" Content="" Margin="277,0,0,35" Background="White" />

<Button x:Name="buttonOtch" Content="Отчёт" Margin="57,197,863,0" Click="buttonOtch\_Click" Background="#FFD9B1A5" FontFamily="Arial Narrow" FontSize="20" Foreground="#FF45503F" OpacityMask="Black" Height="43" VerticalAlignment="Top"/>

<Button x:Name="buttonPya" Content="Выход" HorizontalAlignment="Left" Margin="57,330,0,0" Width="157" Click="buttonPya\_Click" Background="#FFD9B1A5" FontFamily="Arial Narrow" FontSize="20" Foreground="#FF45503F" Height="44" VerticalAlignment="Top"/>

<TextBlock HorizontalAlignment="Left" Margin="118,7,0,412" TextWrapping="Wrap" Text="Пользователь: Митюшкин А.Н" Width="134" FontFamily="Arial Narrow" Foreground="White" FontSize="22"/>

<Button x:Name="buttonOtch\_Copy" Content="Список учащихся" HorizontalAlignment="Left" Margin="57,263,0,0" Width="157" Click="buttonOtch\_Copy\_Click" Background="#FFD9B1A5" FontFamily="Arial Narrow" FontSize="20" Foreground="#FF45503F" OpacityMask="Black" Height="43" VerticalAlignment="Top"/>

</Grid>

</Window>

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

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;

namespace Registor

{

/// <summary>

/// Логика взаимодействия для WindowAdmin.xaml

/// </summary>

public partial class WindowAdmin : Window

{

public WindowAdmin()

{

InitializeComponent();

OpenPage(pages.PageMenuAdmin);

}

public DataTable Select(string selectSQL) // функция подключения к базе данных и обработка запросов

{

DataTable dataTable = new DataTable("dataBase"); // создаём таблицу в приложении

// подключаемся к базе данных

SqlConnection sqlConnection = new SqlConnection("server=ASUS;Trusted\_Connection=Yes;DataBase=ArtShcool;");

sqlConnection.Open(); // открываем базу данных

SqlCommand sqlCommand = sqlConnection.CreateCommand(); // создаём команду

sqlCommand.CommandText = selectSQL; // присваиваем команде текст

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(sqlCommand); // создаём обработчик

sqlDataAdapter.Fill(dataTable); // возращаем таблицу с результатом

return dataTable;

}

public enum pages

{

PageMenuAdmin

}

public void OpenPage(pages pages)

{

if (pages == pages.PageMenuAdmin)

{

Frame1.Navigate(new PageMenuAdmin(this));

}

}

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

{

Frame1.Content = new PageOtchetAdmin();

}

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

{

MessageBox.Show("Выход из системы");

MainWindow main = new MainWindow();

main.Show();

this.Close();

}

private void buttonOtch\_Copy\_Click(object sender, RoutedEventArgs e)

{

OpenPage(pages.PageMenuAdmin);

}

}

}

<Page x:Class="Registor.PageMenuAdmin"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Registor"

xmlns:command="clr-namespace:Registor"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="PageMenuAdmin" x:Name="crud" Loaded="crud\_Loaded">

<StackPanel Background="#FFD9B1A5">

<ToolBar Name="ToolBar" Margin="3" Height="31" Background="White">

<Button x:Name="Undo" Margin="5,2,5,2" ToolTip="Выйти" Click="Undo\_Click">

<Image Source="/kisspng-arrow-computer-icons-symbol-encapsulated-postscrip-5afe6e50b2a392.1156691115266238247317.jpg" Height="18" Width="18" RenderTransformOrigin="0.5,0.5" Stretch="Uniform" >

</Image>

</Button>

<Button Name="Add" Margin="5,2,5,2" ToolTip="Добавить учащегося" Width="20" Height="20" Click="Add\_Click">

<Image Source="/png-clipart-plus-and-minus-signs-computer-icons-plus-sign-cross-addition.png" Width="18" Height="18"></Image>

</Button>

<Button Name="Edit" Margin="5,2,5,2" Click ="Edit\_Click">

<Image Source="/karandash.jpg" Width="18" Height="18" >

</Image>

</Button>

<Button Name="Delet" Margin="5,2,5,2" Click="Delet\_Click">

<Image Source="/png-clipart-red-x-mark-illustration-x-mark-cross-computer-icons-cartoon-geometry-miscellaneous-angle.png" Width="18" Height="18"></Image>

</Button>

</ToolBar>

<DockPanel Background="White">

<DataGrid x:Name="mydataGrid" AutoGenerateColumns="False" ColumnWidth="\*" FontFamily="Arial Narrow" FontSize="14" Width="794" Height="158" SelectionChanged="mydataGrid\_SelectionChanged" Background="#FFD9B1A5" Foreground="#FF333C2E">

<DataGrid.Columns>

<DataGridTextColumn Header="№" Binding="{Binding id}" Width="40" Foreground="#FF333C2E"/>

<DataGridTextColumn Header="ФИО" Binding="{Binding FIO}" Width="200" Foreground="#FF333C2E"/>

<DataGridTextColumn Header="Дата рождения" Binding="{Binding DateBirth}" Width="150" Foreground="#FF333C2E"/>

<DataGridTextColumn Header="Курс" Binding="{Binding Kurs}" Width="50" Foreground="#FF333C2E"/>

<DataGridTextColumn Header="Дата поступления" Binding="{Binding DatePostup}" Width="170" Foreground="#FF333C2E"/>

<DataGridTextColumn Header="Группа" Binding="{Binding Group}" Width="200" Foreground="#FF333C2E"/>

</DataGrid.Columns>

</DataGrid>

</DockPanel>

<Label Content="Введите данные:" Width="129" HorizontalAlignment="Left" FontFamily="Arial Narrow" FontSize="18" Foreground="#FF333C2E"/>

<Grid Height="30" Margin="0,2,0,0">

<TextBox x:Name="tbId" Width="40" HorizontalAlignment="Left" Margin="5,0,0,0" FontSize="16" FontFamily="Arial Narrow"></TextBox>

<TextBox x:Name="tbFIO" Width="200" HorizontalAlignment="Left" Margin="45,0,0,0" FontSize="16" FontFamily="Arial Narrow"></TextBox>

<TextBox x:Name="tbBirth" Margin="245,0,0,0" Width="150" HorizontalAlignment="Left" FontSize="16" FontFamily="Arial Narrow"></TextBox>

<TextBox x:Name="tbKurs" Width="50" Margin="395,0,0,0" HorizontalAlignment="Left" FontSize="16" FontFamily="Arial Narrow"></TextBox>

<TextBox x:Name="tbDatePostup" Width="170" HorizontalAlignment="Left" Margin="445,0,0,0" FontSize="16" FontFamily="Arial Narrow"></TextBox>

<TextBox x:Name="tbGroup" Width="185" HorizontalAlignment="Right" FontSize="16" FontFamily="Arial Narrow"></TextBox>

</Grid>

</StackPanel>

</Page>

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

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;

namespace Registor

{

/// <summary>

/// Логика взаимодействия для PageMenuAdmin.xaml

/// </summary>

public partial class PageMenuAdmin : Page

{

public WindowAdmin windowAdmin;

public PageMenuAdmin(WindowAdmin \_windowAdmin)

{

InitializeComponent();

windowAdmin = \_windowAdmin;

}

SqlConnection sqlConnection = new SqlConnection("server=ASUS;Trusted\_Connection=Yes;DataBase=ArtShcool;");

SqlCommand cmd;

SqlDataAdapter da;

DataTable db;

private void clear()

{

tbId.Text = " ";

tbFIO.Text = " ";

tbBirth.Text = " ";

tbKurs.Text = " ";

tbDatePostup.Text = " ";

tbGroup.Text = " ";

}

private void show()

{

da = new SqlDataAdapter("Select \* From [dbo].[Students]", sqlConnection);

db = new DataTable();

da.Fill(db);

mydataGrid.ItemsSource = db.DefaultView;

tbId.Focus();

}

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

{

show();

}

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

{

cmd = new SqlCommand("INSERT INTO [dbo].[Students] (id, FIO, DateBirth, Kurs, DatePostup, [Group]) Values (@id, @FIO, @DateBirth, @Kurs, @DatePostup, @Group)", sqlConnection);

cmd.Parameters.AddWithValue("@id", tbId.Text);

cmd.Parameters.AddWithValue("@FIO", tbFIO.Text);

cmd.Parameters.AddWithValue("@DateBirth", tbBirth.Text);

cmd.Parameters.AddWithValue("@Kurs", tbKurs.Text);

cmd.Parameters.AddWithValue("@DatePostup", tbDatePostup.Text);

cmd.Parameters.AddWithValue("@Group", tbGroup.Text);

sqlConnection.Open();

cmd.ExecuteNonQuery();

sqlConnection.Close();

MessageBox.Show("Данные успешно добавлены");

show();

clear();

}

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

{

cmd = new SqlCommand("Update [dbo].[Students] Set FIO=@FIO, DateBirth=@DateBirth, Kurs=@Kurs, DatePostup=@DatePostup, [Group]=@Group Where id=@id", sqlConnection);

cmd.Parameters.AddWithValue("@id", tbId.Text);

cmd.Parameters.AddWithValue("@FIO", tbFIO.Text);

cmd.Parameters.AddWithValue("@DateBirth", tbBirth.Text);

cmd.Parameters.AddWithValue("@Kurs", tbKurs.Text);

cmd.Parameters.AddWithValue("@DatePostup", tbDatePostup.Text);

cmd.Parameters.AddWithValue("@Group", tbGroup.Text);

sqlConnection.Open();

cmd.ExecuteNonQuery();

sqlConnection.Close();

MessageBox.Show("Данные успешно изменены");

show();

clear()

}

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

{

cmd = new SqlCommand("Delete From [dbo].[Students] Where id=@id", sqlConnection);

cmd.Parameters.AddWithValue("@id", tbId.Text);

sqlConnection.Open();

cmd.ExecuteNonQuery();

sqlConnection.Close();

MessageBox.Show("Данные успешно удалены");

show();

clear();

}

private void mydataGrid\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

DataGrid dataGrid = (DataGrid)sender;

DataRowView dataRowView = dataGrid.SelectedItem as DataRowView;

if (dataRowView != null)

{

tbId.Text = dataRowView["id"].ToString();

tbFIO.Text = dataRowView["FIO"].ToString();

tbBirth.Text = dataRowView["DateBirth"].ToString();

tbKurs.Text = dataRowView["Kurs"].ToString();

tbDatePostup.Text = dataRowView["DatePostup"].ToString();

tbGroup.Text = dataRowView["Group"].ToString();

}

}

}

}

<Page x:Class="Registor.PageOtchetAdmin"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Registor"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="PageOtchetAdmin">

<StackPanel Background="#FFD9B1A5">

<ToolBar Name="ToolBar" Margin="3" Height="31">

<Button x:Name="Undo" Margin="5,2,5,2" ToolTip="Выйти" >

<Image Source="/kisspng-arrow-computer-icons-symbol-encapsulated-postscrip-5afe6e50b2a392.1156691115266238247317.jpg" Height="18" Width="18" RenderTransformOrigin="0.5,0.5" Stretch="Uniform" >

</Image>

</Button>

<Button Name="Add" Margin="5,2,5,2" ToolTip="Добавить учащегося" Width="20" Height="20" Click="Add\_Click">

<Image Source="/png-clipart-plus-and-minus-signs-computer-icons-plus-sign-cross-addition.png" Width="18" Height="18"></Image>

</Button>

<Button Name="Edit" Margin="5,2,5,2" Click="Edit\_Click">

<Image Source="/karandash.jpg" Width="18" Height="18" >

</Image>

</Button>

<Button Name="Delet" Margin="5,2,5,2" Click="Delet\_Click">

<Image Source="/png-clipart-red-x-mark-illustration-x-mark-cross-computer-icons-cartoon-geometry-miscellaneous-angle.png" Width="18" Height="18"></Image>

</Button>

</ToolBar>

<DockPanel Background="White">

<DataGrid x:Name="mydataGrid" AutoGenerateColumns="False" ColumnWidth="\*" FontFamily="Arial Narrow" FontSize="14" Width="794" Height="158" SelectionChanged="mydataGrid\_SelectionChanged" Background="#FFD9B1A5" Foreground="#FF343C2F">

<DataGrid.Columns>

<DataGridTextColumn Header="№" Binding="{Binding id}" Width="30" FontFamily="Arial Narrow"/>

<DataGridTextColumn Header="ФИО" Binding="{Binding FIO}" Width="150"/>

<DataGridTextColumn Header="Дата рождения" Binding="{Binding DateBirth}" Width="110"/>

<DataGridTextColumn Header="Курс" Binding="{Binding Well}" Width="45"/>

<DataGridTextColumn Header="Группа" Binding="{Binding Group}" Width="150"/>

<DataGridTextColumn Header="Дата оплаты" Binding="{Binding PaymentAmount}" Width="110"/>

<DataGridTextColumn Header="Сумма оплаты" Binding="{Binding DatePayment}" Width="100"/>

<DataGridTextColumn Header="Сумма долга" Binding="{Binding DedtAmount}" Width="100"/>

</DataGrid.Columns>

</DataGrid>

</DockPanel>

<Label Content="Введите данные:" Width="137" HorizontalAlignment="Left" FontFamily="Arial Narrow" FontSize="18" Foreground="#FF343C2F"/>

<Grid Height="30" Margin="0,2,0,0">

<TextBox x:Name="tbId" Width="35" HorizontalAlignment="Left"/>

<TextBox x:Name="tbFIO" Width="150" HorizontalAlignment="Left" Margin="35,0,0,0"></TextBox>

<TextBox x:Name="tbBirth" Margin="185,0,0,0" Width="110" HorizontalAlignment="Left"></TextBox>

<TextBox x:Name="tbDatePostup" Width="45" HorizontalAlignment="Left" Margin="295,0,0,0"></TextBox>

<TextBox x:Name="tbGroup" Width="150" HorizontalAlignment="Left" Margin="340,0,0,0"></TextBox>

<TextBox x:Name="tb1" Width="110" HorizontalAlignment="Left" Margin="490,0,0,0"></TextBox>

<TextBox x:Name="tb2" Width="100" HorizontalAlignment="Left" Margin="600,0,0,0"></TextBox>

<TextBox x:Name="tb3" Width="100" HorizontalAlignment="Left" Margin="700,0,0,0"></TextBox>

</Grid>

</StackPanel>

</Page>

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

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;

namespace Registor

{

/// <summary>

/// Логика взаимодействия для PageOtchetAdmin.xaml

/// </summary>

public partial class PageOtchetAdmin : Page

{public MainWindow mainWindow;

public PageOtchetAdmin()

{

InitializeComponent();

}

SqlConnection sqlConnection = new SqlConnection("server=ASUS;Trusted\_Connection=Yes;DataBase=ArtShcool;");

SqlCommand cmd;

SqlDataAdapter da;

DataTable db;

private void clear()

{

tbId.Text = " ";

tbFIO.Text = " ";

tbBirth.Text = " ";

tbDatePostup.Text = " ";

tbGroup.Text = " ";

tb1.Text = " ";

tb2.Text = " ";

tb3.Text = " ";

}

private void show()

{

da = new SqlDataAdapter("Select \* From [dbo].[Otchet]", sqlConnection);

db = new DataTable();

da.Fill(db);

mydataGrid.ItemsSource = db.DefaultView;

tbId.Focus();

}

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

{

show();

}

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

{

cmd = new SqlCommand("INSERT INTO [dbo].[Otchet] (id, FIO, DateBirth, Well, [Group], PaymentAmount, DatePayment, DedtAmount) Values (@id, @FIO, @DateBirth, @Well, @Group, @PaymentAmount, @DatePayment, @DedtAmount)", sqlConnection);

cmd.Parameters.AddWithValue("@id", tbId.Text);

cmd.Parameters.AddWithValue("@FIO", tbFIO.Text);

cmd.Parameters.AddWithValue("@DateBirth", tbBirth.Text);

cmd.Parameters.AddWithValue("@Well", tbDatePostup.Text);

cmd.Parameters.AddWithValue("@Group", tbGroup.Text);

cmd.Parameters.AddWithValue("@PaymentAmount", tb1.Text);

cmd.Parameters.AddWithValue("@DatePayment", tb2.Text);

cmd.Parameters.AddWithValue("@DedtAmount", tb3.Text);

sqlConnection.Open();

cmd.ExecuteNonQuery();

sqlConnection.Close();

MessageBox.Show("Данные успешно добавлены");

show();

clear();

}

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

{

cmd = new SqlCommand("Delete From [dbo].[Otchet] Where id=@id", sqlConnection);

cmd.Parameters.AddWithValue("@id", tbId.Text);

sqlConnection.Open();

cmd.ExecuteNonQuery();

sqlConnection.Close();

MessageBox.Show("Данные успешно удалены");

show();

clear();

}

private void mydataGrid\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

DataGrid dataGrid = (DataGrid)sender;

DataRowView dataRowView = dataGrid.SelectedItem as DataRowView;

if (dataRowView != null)

{

tbId.Text = dataRowView["id"].ToString();

tbFIO.Text = dataRowView["FIO"].ToString();

tbBirth.Text = dataRowView["DateBirth"].ToString();

tbDatePostup.Text = dataRowView["Well"].ToString();

tbGroup.Text = dataRowView["Group"].ToString();

tb1.Text = dataRowView["PaymentAmount"].ToString();

tb2.Text = dataRowView["DatePayment"].ToString();

tb3.Text = dataRowView["DedtAmount"].ToString();

}

}

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

{

cmd = new SqlCommand("Update [dbo].[Otchet] Set FIO=@FIO, DateBirth=@DateBirth, Well=@Well, [Group]=@Group, PaymentAmount=@PaymentAmount, DatePayment=@DatePayment, DedtAmount=@DedtAmount Where id=@id", sqlConnection);

cmd.Parameters.AddWithValue("@id", tbId.Text);

cmd.Parameters.AddWithValue("@FIO", tbFIO.Text);

cmd.Parameters.AddWithValue("@DateBirth", tbBirth.Text);

cmd.Parameters.AddWithValue("@Well", tbDatePostup.Text);

cmd.Parameters.AddWithValue("@Group", tbGroup.Text);

cmd.Parameters.AddWithValue("@PaymentAmount", tb1.Text);

cmd.Parameters.AddWithValue("@DatePayment", tb2.Text);

cmd.Parameters.AddWithValue("@DedtAmount", tb3.Text);

sqlConnection.Open();

cmd.ExecuteNonQuery();

sqlConnection.Close();

MessageBox.Show("Данные успешно изменены");

show();

clear();

}

}

}