Государственное бюджетное профессиональное образовательное учреждение

Самарской области

«Тольяттинский социально-экономический колледж»

**КУРСОВАЯ РАБОТА**

**РАЗРАБОТКА И ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «УЧЁТ УЧАСТНИКОВ ОЛИМПИАДЫ»**

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

**МДК 05.02 Разработка кода информационных систем**

**09.02.07 Информационные системы и программирование**

|  |  |  |  |
| --- | --- | --- | --- |
| Студент |  | **/** | Ю.А. Москалёв |
|  | *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 системмодуля, выполняемой в рамках МДК.05.02 Разработка кода информационных систем**

студента группы ИСП-31

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Москалёва Юрия Александровича\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Фамилия Имя отчество студента*

Тема курсовой работы: «Разработка информационной системы **«**Учёт контингента учащихся учебного заведения»

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 (название)

1.1 Разработка физической структуры БД

1.2 Разработка запросов ИС

2 Разработка интерфейса информационной системы (название)

3 Разработка мобильного приложения

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

По теме Разработка и проектирование ИС «Учёт участников олимпиады»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  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 2021 | 5% |  |
| 2 | Утверждение темы, согласование плана. Введение, библиография | Январь 2021 | 10% |  |
| 3 | Изучение и анализ информационных материалов по теме | Февраль 2021 | 15% |  |
| 4 | Обоснование актуальности выбранной темы применительно к профессиональной деятельности (введение) | Февраль 2021 | 20% |  |
| 5 | Изложение материала основной части по теме курсовой работы | Февраль 2021 | 20% |  |
| 6 | Подведение итогов проведенного анализа, формулировка выводов УИР применительно к профессиональной деятельности (заключение) | Март 2021 | 20% |  |
| 7 | Оформление работы и сдача на проверку | Март 2021 | 10% |  |
| 8 | Защита работы |  |  |  |

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

**Содержание**

[ВВЕДЕНИЕ 5](#_Toc94035240)

[1 РАЗРАБОТКА БАЗЫ ДАННЫХ ИНФОРМАЦИОННОЙ СИСТЕМЫ УЧЁТ КОНТИНГЕНТА УЧАЩИХСЯ УЧЕБНОГО ЗАВЕДЕНИЯ 5](#_Toc94035241)

[1.1 Разработка физической структуры БД 5](#_Toc94035242)

[1.2 Разработка запросов ИС 5](#_Toc94035243)

[2 РАЗРАБОТКА ИНТЕРФЕЙСА ИНФОРМАЦИОННОЙ СИСТЕМЫ УЧЁТ КОНТИНГЕНТА УЧАЩИХСЯ УЧЕБНОГО ЗАВЕДЕНИЯ 6](#_Toc94035244)

[3 РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ 7](#_Toc94035245)

[ЗАКЛЮЧЕНИЕ 8](#_Toc94035246)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 9](#_Toc94035247)

[ПРИЛОЖЕНИЕ А.](#_Toc94035248) [ЛИСТИНГ ПРОГРАММЫ 10](#_Toc94035249)

# ВВЕДЕНИЕ

Актуальность внедрения автоматизированной информационной системы заключается в инновационном подходе хранения документов в электронной базе данных, является более выгодным вариантом, нежели хранение документов в бумажном формате.

Целью создания информационной системы является: минимизация времени учёта участников олимпиады и её информационным отображением, но самым главным - повышение эффективности взаимодействия и улучшение качества работы.

ИС предназначена для эффективного и удобного учёта участников олимпиады «ТСЭКSkills»; подаче желающими заявки на участие в олимпиаде через ЭВМ; быстрого доступа к информации.

# Для достижения поставленной цели были определены следующие задачи:

# проанализировать предметную область;

# обосновать актуальность разработки информационной системы;

# разработать функциональную модель;

# описать средства разработки информационной системы;

* + - спроектировать информационную систему, построив различные виды диаграмм.

При изучении предметной области, выяснилось, что для реализации информационной системы «Учёт участников олимпиады» необходимо обратиться к разделам, таким как создание базы данных и создания приложения для взаимодействия с этой базой данных.

Для разработки информационной системы будут применяться методы: анализ деятельности предприятия с выявлением его функции, а также функциональное моделирование системы.

# 1 РАЗРАБОТКА БАЗЫ ДАННЫХ ИНФОРМАЦИОННОЙ СИСТЕМЫ (УЧЁТ УЧАСТНИКОВ ОЛИМПИАДЫ)

## 1.1 Разработка физической структуры БД

Чтобы создать новую базу данных, необходимо запустить SQL Server Management Studio и подключится к серверу, нажать на назначение сервера в окне Object Explorer и в появившемся меню выберем пункт New Database.

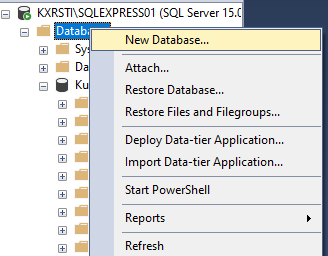
**

Рисунок 1.1 – Создание базы данных

В обозревателе объектов необходимо открыть контейнер «Базы данных», затем нужную базу данных и кликом правой кнопкой мыши по пункту «Table» навести мышку на вкладку «New» и выбрать «Table…».

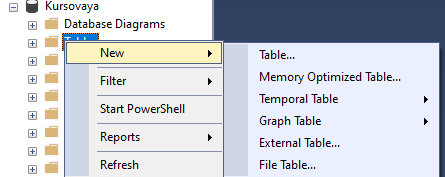
**

Рисунок 1.2 – Создание таблицы

После чего необходимо дать имена столбцов, выбрать типы данных и указать необходимые свойства.

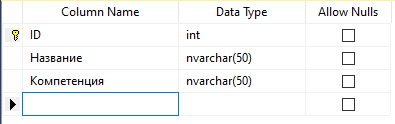


Рисунок 1.3 – Название таблиц и выбор типов данных

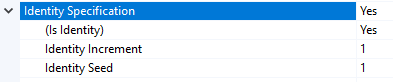


Рисунок 1.4 – Указание столбцу ID свойства «Identity Spetification»

Таблицы, созданные в рамках текущей базы данных:

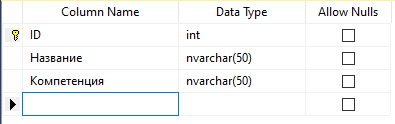


Рисунок 1.5 – Информация об олимпиадах

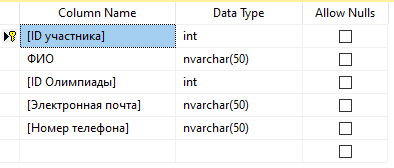


Рисунок 1.6 – Информация об участниках

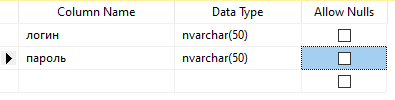


Рисунок 1.7 – Таблица авторизации

## 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: **C**reate (создание), **R**ead (чтение), **U**pdate (обновление) и **D**elete (удаление).

Скрипт для создания таблицы:

CREATE TABLE [dbo].[Информация об олимпиадах](

[ID] [int] IDENTITY(1,1) NOT NULL,

[Название] [nvarchar](50) NULL,

[Компетенция] [nvarchar](50) NULL,

CONSTRAINT [PK\_Информация об олимпиадах] PRIMARY KEY CLUSTERED

Скрипт для обновления таблицы:

UPDATE [dbo].[Информация об олимпиадах]

SET [Название] = <Название, nvarchar(50),>

,[Компетенция] = <Компетенция, nvarchar(50),>

WHERE <Search Conditions,,>

Скрипт для выбора таблицы:

SELECT [ID]

,[Название]

,[Компетенция]

# FROM [dbo].[Информация об олимпиадах]

Скрипт для удаления таблицы

DELETE FROM [dbo].[Информация об олимпиадах]

WHERE <Search Conditions,,>

# 2 РАЗРАБОТКА ИНТЕРФЕЙСА ИНФОРМАЦИОННОЙ СИСТЕМЫ «УЧЁТ УЧАСТНИКОВ ОЛИМПИАДЫ»

**2.1 Разработка интерфейса**

Первым шагом является создание проекта WPF в Visual C#

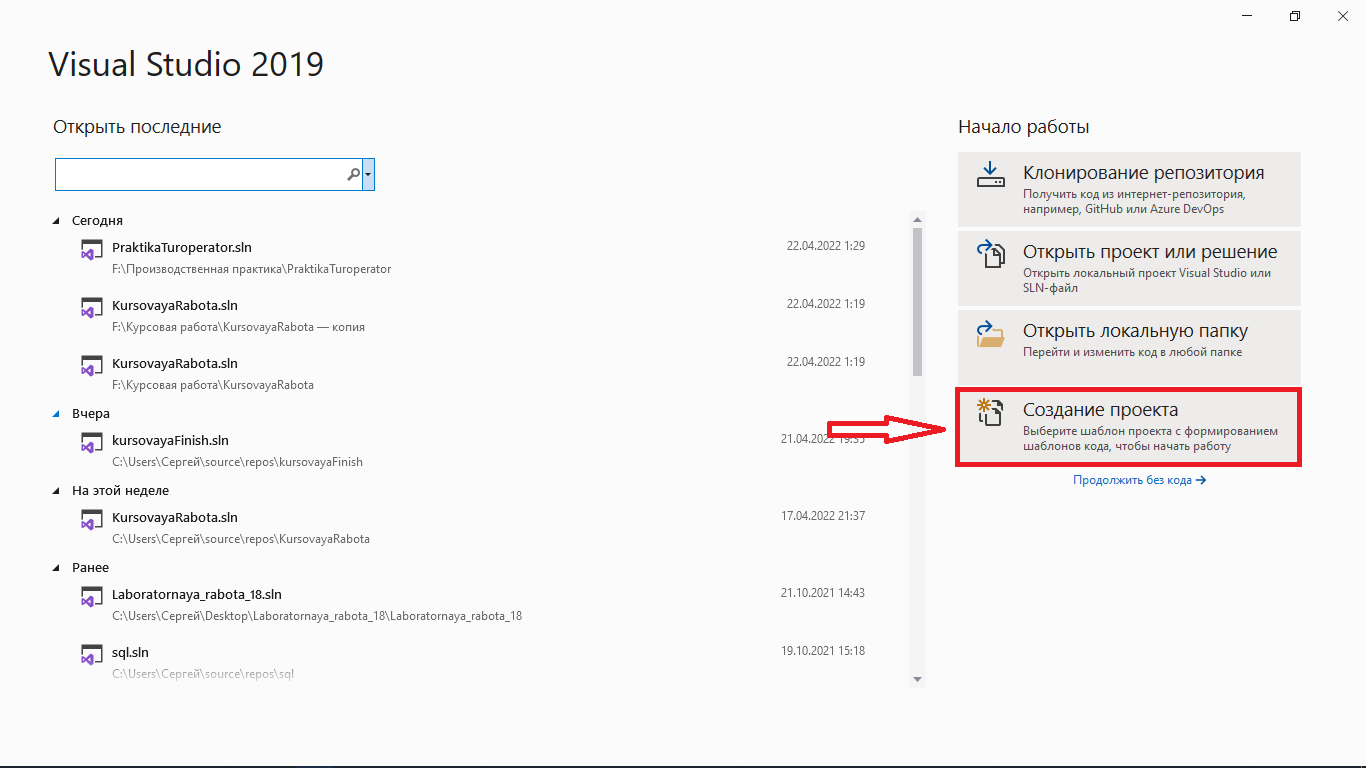


Рисунок 2.1 – Создание проекта

Откроется диалоговое окно «Создание проекта», самым простым способом нужного нам шаблона будет поиск. Необходимо в строке сверху написать «WPF», выбрать шаблон «Приложение WPF (.NET Framework)» и нажать кнопку далее.

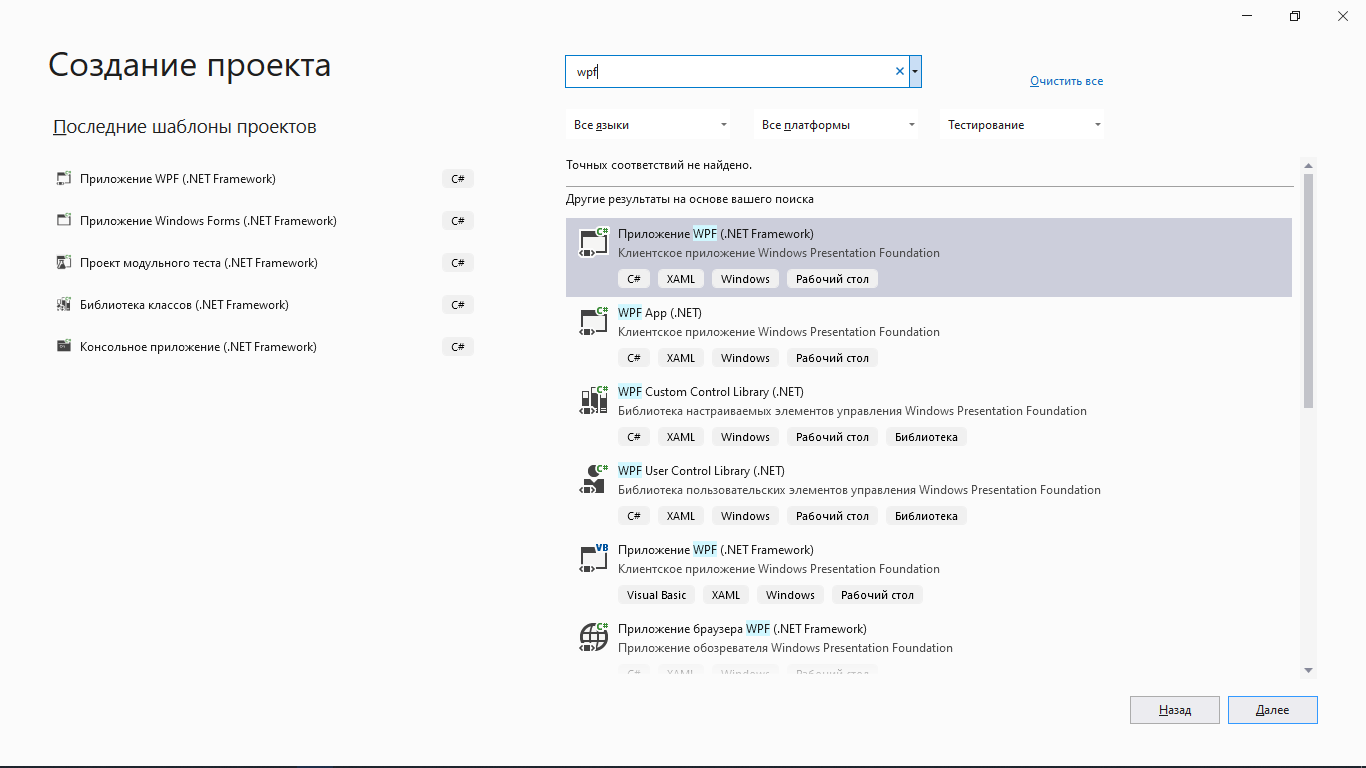


Рисунок 2.2 – Выбор нужного шаблона

Откроется диалоговое окно «Настроить новый проект», где необходимо указать имя проекта, расположение, имя решения и платформу

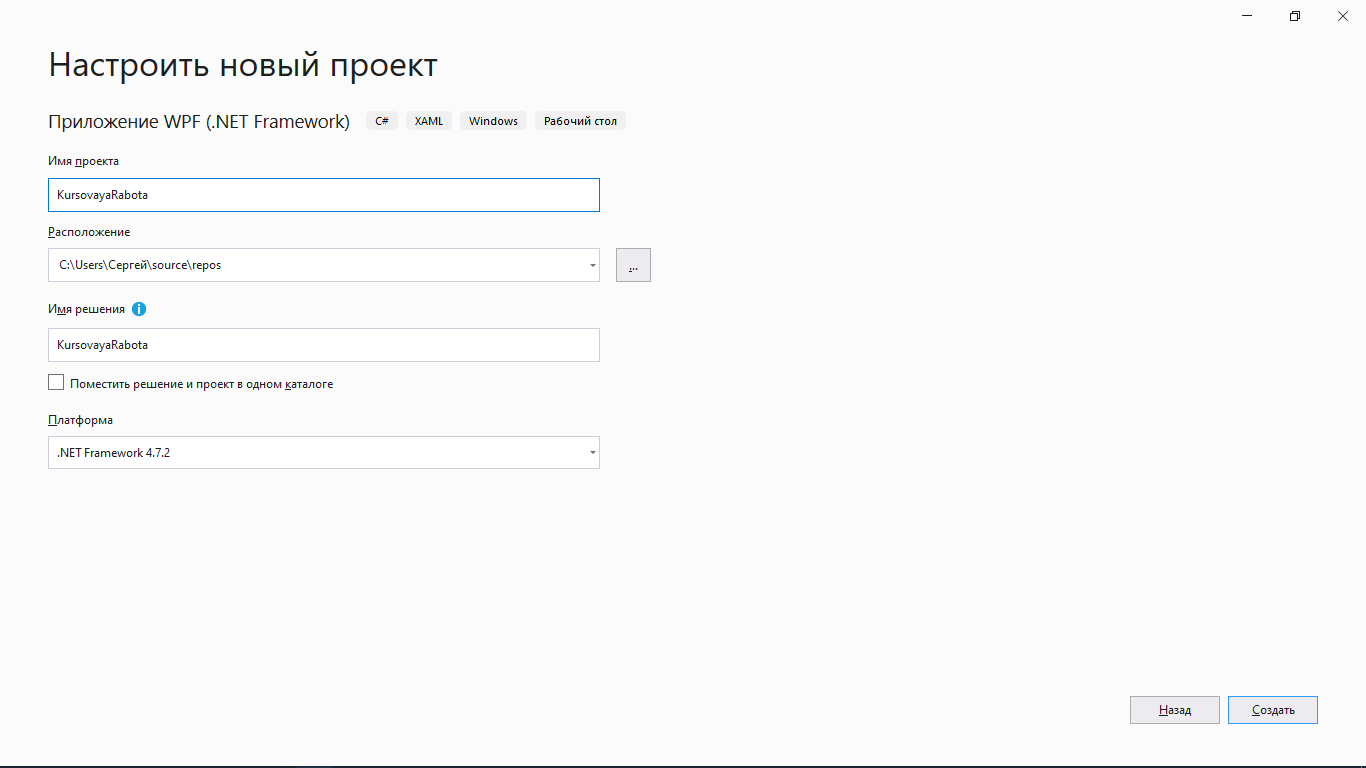


Рисунок 2.3 – Настройка проекта

Visual Studio создаст проект и откроет конструктор для окна приложения по умолчанию с именем MainWindow.xaml.

Для визуально красивого интерфейса, можно подключить пакет NuGet «MaterialDesignThemes».

Код App.Xaml с подключенным пакетом NuGet должен выглядеть следующим образом:

<Application x:Class="KursovayaRabota.App"

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

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

xmlns:local="clr-namespace:KursovayaRabota"

StartupUri="MainWindow.xaml">

<Application.Resources>

<ResourceDictionary>

<ResourceDictionary.MergedDictionaries>

<ResourceDictionary Source="pack://application:,,,/MaterialDesignThemes.Wpf;component/Themes/MaterialDesignTheme.Light.xaml" />

<ResourceDictionary Source="pack://application:,,,/MaterialDesignThemes.Wpf;component/Themes/MaterialDesignTheme.Defaults.xaml" />

<ResourceDictionary Source="pack://application:,,,/MaterialDesignColors;component/Themes/Recommended/Primary/MaterialDesignColor.Green.xaml" />

<ResourceDictionary Source="pack://application:,,,/MaterialDesignColors;component/Themes/Recommended/Accent/MaterialDesignColor.Lime.xaml" />

</ResourceDictionary.MergedDictionaries>

</ResourceDictionary>

</Application.Resources>

</Application>

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.

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.

Код MainWindow.xaml должен выглядеть следующим образом:

<Window x:Class="KursovayaRabota.MainWindow"

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"

xmlns:local="clr-namespace:KursovayaRabota"

mc:Ignorable="d"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

TextElement.Foreground="{DynamicResource MaterialDesignBody}"

TextElement.FontWeight="Regular"

TextElement.FontSize="13"

TextOptions.TextFormattingMode="Ideal"

TextOptions.TextRenderingMode="Auto"

Background="{DynamicResource MaterialDesignPaper}"

FontFamily="{DynamicResource MaterialDesignFont}"

Title="Авторизация" Height="450" Width="800" MinHeight="450" MinWidth="800">

<Grid>

<Grid.Background>

<LinearGradientBrush EndPoint="0.5,1" MappingMode="RelativeToBoundingBox" StartPoint="0.5,0">

<GradientStop Color="#FFFF4BA7"/>

<GradientStop Color="#FF56CCFF" Offset="1"/>

</LinearGradientBrush>

</Grid.Background>

<Border MinWidth="400" Margin="15" Background="White" VerticalAlignment="Center" Padding="30" MaxWidth="550" CornerRadius="20">

<Border.Effect>

<DropShadowEffect BlurRadius="10" Color="LightGray" ShadowDepth="0" />

</Border.Effect>

<StackPanel Margin="0,0,0,0">

<Image Width="50" HorizontalAlignment="Center" Margin="0 0 0 10" Source="tHWDWmULLcA-architectural-scale-4\_00x.png" />

<TextBox x:Name="textBoxLogin" materialDesign:HintAssist.Hint="Логин" Style="{StaticResource MaterialDesignFloatingHintTextBox}" CaretBrush="#FFB86BE4" SelectionBrush="#FFB86BE4" FontFamily="Comic Sans MS"/>

<PasswordBox x:Name="passBox" materialDesign:HintAssist.Hint="Пароль" Style="{StaticResource MaterialDesignFloatingHintPasswordBox}" CaretBrush="#FFB86BE4" SelectionBrush="#FFB86BE4" FontFamily="Comic Sans MS"/>

<Button x:Name="buttonEnter" Content="Войти" Margin="0 20 0 0" Background="#FF01B905" BorderBrush="#FF01B905" Foreground="White" FontFamily="Comic Sans MS" Click="buttonEnter\_Click"/>

<Button x:Name="buttonReg" Content="войти как гость" Margin="0,10,0,0" HorizontalAlignment="Center" Style="{StaticResource MaterialDesignFlatButton}" Foreground="#FF01B905" FontFamily="Comic Sans MS" BorderBrush="#FF01B905"/>

</StackPanel>

</Border>

</Grid>

</Window>

## 2.5 Разработка руководства пользователя

ИС «Учёт участников олимпиады» представляет собой программный комплекс для эффективного учёта и лёгкого поиска информации об олимпиадах «ТСЭКSkills», и его участников.

Данное руководство предназначено для пользователей компьютерной программы, установленного на устройствах, работающих под управлением операционной системы windows.

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

Для авторизации как Администратора необходимо произвести вход с помощью логина и пароля в открывшемся окне при входе в приложение. Для входа как гостя (участника) логин и пароль не требуется.

После входа под администратором, откроется окно «Меню администратора», где можно увидеть все функции администратора. После входа под гостем (участником), откроется окно с информацией об олимпиадах с возможностью подать заявку на участие.

При нажатии на кнопку «Информация об олимпиадах» в «Меню администратора», откроется окно с возможностью просмотра, добавления, изменения и удаления информации об олимпиадах.

При нажатии на кнопку «Информация об участниках» в «Меню администратора», откроется окно с возможностью просмотра и удаления информации об участниках.

## Окна приложения:

|  |  |
| --- | --- |
| Окно | Функционал |
| Авторизация | Главная страница приложения |
| Меню администратора | Содержит переходы на главные функции администратора |
| Информация об олимпиадах | Содержит информацию об олимпиадах |
| Информация об участниках | Содержит информацию об участниках |
| Окно гостя | Содержит информацию об олимпиадах, функцию подачи заявки на участие |

Для работы системы требуются IBM совместимые персональные компьютеры.

Минимальная конфигурация сервера:

* тип процессора – AMD Ryzen 5 3500 with Redon Vega Mobile Gfx и выше или совместимый с ним;
* объем оперативного запоминающего устройства 4Гб и более;
* жесткий диск 80 Гб;
* модем, для выхода в Интернет;
* монитор, клавиатура, мышь.
* Требования, предъявляемые к конфигурации клиентских станций:
* процессор, с тактовой частотой не менее 400 MHz,
* 256 Mb оперативной памяти;
* Монитор – SVGA;
* Клавиатура - 101/102 клавиши;
* Манипулятор типа «мышь».

# На рисунке 2.4 представлено окно авторизации

Рисунок 2.4 – окно авторизации

Если мы введём неправильные данные, то нам выдаст ошибку, показанную на рисунке 2.5

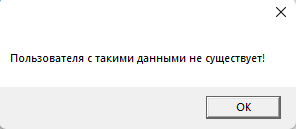


Рисунок 2.5 – окно ошибки

После успешной авторизации под администратором мы попадаем в «Меню администратора», на котором изображено две кнопки «Информация об олимпиадах» и «Информация об участниках», показанные на рисунке 2.6

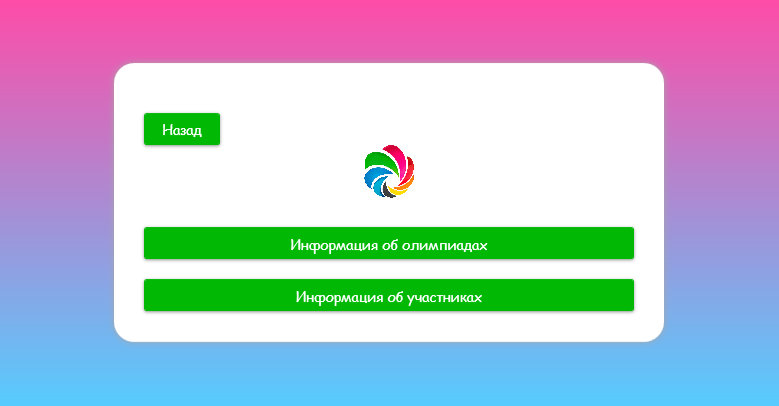


Рисунок 2.6 – меню администратора

Если мы кликнем на кнопку «Информация об олимпиадах», то мы попадем в окно, где можно будет посмотреть, изменять, добавлять, удалять информацию об олимпиадах. Окно показано на рисунке 2.7

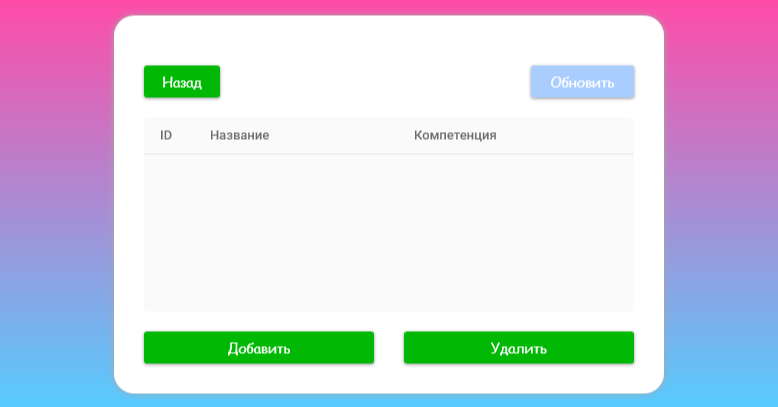


Рисунок 2.7 – информация об олимпиадах

Если мы кликнем на кнопку «Информация об участниках», то мы попадем в окно, где можно будет посмотреть и удалять информацию об участниках. Окно показано на рисунке 2.8

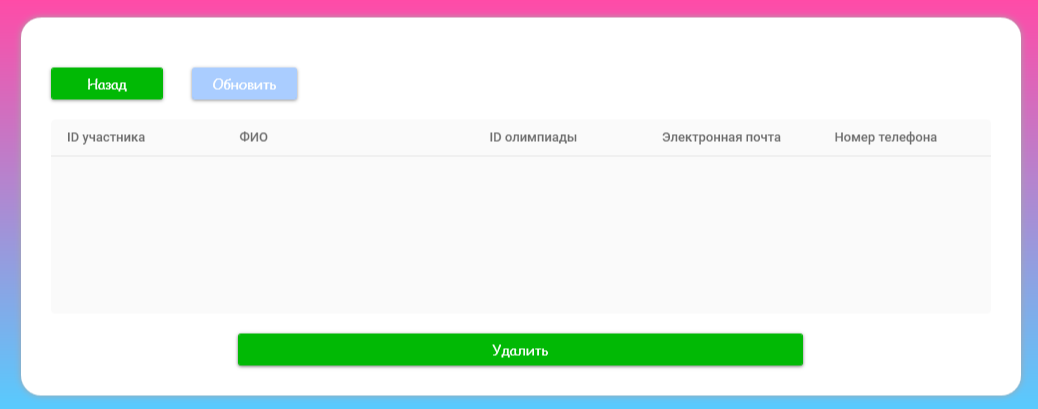


Рисунок 2.8 – информация об участниках

После входа под гостем мы попадаем в «Окно гостя», где можно ознакомиться с информацией об олимпиадах и подать заявку на участие. Окно показано на рисунке 2.9

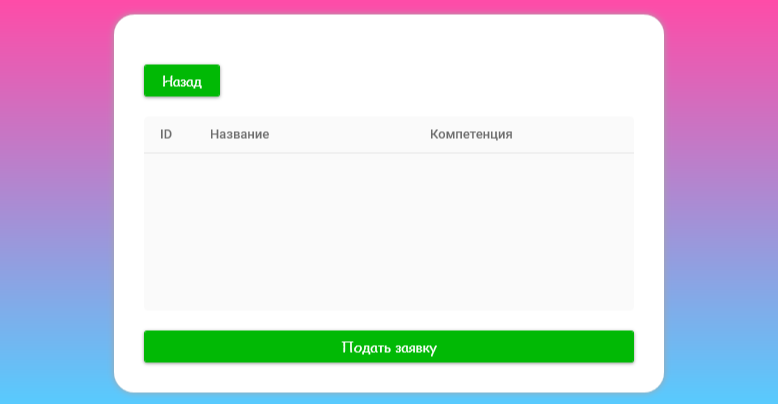


Рисунок 2.9 – окно гостя

# ЗАКЛЮЧЕНИЕ

В ходе выполнения второй части курсовой работы, была разработана программа, получены и развиты навыки создания базы данных.

Разработанная информационная система «Учёт участников олимпиады» предназначена для эффективного учёта и быстрого поиска информации.

Система содержит основную информацию об олимпиадах и их участниках.

ИС обеспечивает возможность удаления, добавления и редактирования записей.

Удобный, интуитивно понятный интерфейс автоматизированной системы, с одной стороны, позволяет легко ориентироваться в программе, требуя от пользователя лишь небольшого количества специальных навыков работы с электронно-вычислительными машинами, с другой стороны, предоставляет пользователю оперативную информацию обо всех интересующих его данных.

Данный программный продукт предназначен для организаторов олимпиад, чтобы упростить им учёт участников, работу с поиском информации и позволит повысить эффективность выполнения работ.**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. ГОСТ 34.601 – 90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.
2. ГОСТ 34.602 - 2020. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.
3. ГОСТ 19.201 - 78 ЕСПД. Техническое задание. Требования к содержанию и оформлению.
4. ГОСТ 19.202 - 78 ЕСПД. Спецификация. Требования к содержанию и оформлению.
5. ГОСТ Р ИСО/МЭК 12207. Процессы жизненного цикла программных средств.
6. Золотов С.Ю. Проектирование информационных систем: Учебно-методическое пособие. Методические рекомендации для выполнения курсового проекта, лабораторных работ и практических занятий по дисциплине «Проектирование информационных систем» - Томск: ТУСУР, 2013. - 34 с.
7. Шнайдер, Роберт Microsoft SQL Server 6.5. Проектирование высокопроизводительных баз данных; М.: Лори, 2010. - 361 c
8. Петкович, Душан Microsoft SQL Server 2012. Руководство для начинающих / Душан Петкович. - М.: БХВ-Петербург, 2012. - 460 c.
9. Тейлор, Аллен SQL для чайников / Аллен Тейлор. - М.: Вильямс, 2014. - 416 c.
10. Браст, Э.Дж. Разработка приложений на основе Microsoft SQL Server 2008 / Э.Дж. Браст. - М.: Русская Редакция, 2010. - 751 c.
11. Хетагуров, Я. А. Проектирование автоматизированных систем обработки информации и управления (АСОИУ). Учебник / Я.А. Хетагуров. - М.: Бином. Лаборатория знаний, 2015. - 240 c.
12. Кристофер, Д. Маннинг Введение в информационный поиск / Кристофер Д. Маннинг, ПрабхакарРагхаван ,ХайнрихШютце. - М.: Вильямс, 2014. - 528 c.
13. Бишоп Дж. С# в кратком изложении; Бином. Лаборатория знаний - М., 2015. - 234 c.
14. ГриффитсИэн Программирование на C# 5.0; Эксмо - М., 2014. - 580 c.
15. Гуриков С. Р. Введение в программирование на языке Visual C#; ИЛ - Москва, 2013. - 448 c.
16. Шилдт Герберт C# 4.0. Полное руководство; Вильямс - М., 2015. - 291 c.
17. Эндрю Троелсен Язык программирования C# 5.0 и платформа .NET 4.5; Диалектика / Вильямс - М., 2015. - 126 c.

ПРИЛОЖЕНИЕ А

# ЛИСТИНГ ПРОГРАММЫ

**Авторизация**

xaml:

<Window x:Name="mainWindow" x:Class="KursovayaRabota.MainWindow"

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"

xmlns:local="clr-namespace:KursovayaRabota"

mc:Ignorable="d"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

TextElement.Foreground="{DynamicResource MaterialDesignBody}"

TextElement.FontWeight="Regular"

TextElement.FontSize="13"

TextOptions.TextFormattingMode="Ideal"

TextOptions.TextRenderingMode="Auto"

Background="{DynamicResource MaterialDesignPaper}"

FontFamily="{DynamicResource MaterialDesignFont}"

Title="Авторизация" Height="450" Width="800" MinHeight="450" MinWidth="800">

<Grid>

<Grid.Background>

<LinearGradientBrush EndPoint="0.5,1" MappingMode="RelativeToBoundingBox" StartPoint="0.5,0">

<GradientStop Color="#FFFF4BA7"/>

<GradientStop Color="#FF56CCFF" Offset="1"/>

</LinearGradientBrush>

</Grid.Background>

<Border MinWidth="400" Margin="15" Background="White" VerticalAlignment="Center" Padding="30" MaxWidth="550" CornerRadius="20">

<Border.Effect>

<DropShadowEffect BlurRadius="10" Color="LightGray" ShadowDepth="0" />

</Border.Effect>

<StackPanel Margin="0,0,0,0">

<Image Width="50" HorizontalAlignment="Center" Margin="0 0 0 10" Source="tHWDWmULLcA-architectural-scale-4\_00x.png" />

<TextBox x:Name="textBoxLogin" materialDesign:HintAssist.Hint="Логин" Style="{StaticResource MaterialDesignFloatingHintTextBox}" CaretBrush="#FFB86BE4" SelectionBrush="#FFB86BE4" FontFamily="Comic Sans MS"/>

<PasswordBox x:Name="passBox" materialDesign:HintAssist.Hint="Пароль" Style="{StaticResource MaterialDesignFloatingHintPasswordBox}" CaretBrush="#FFB86BE4" SelectionBrush="#FFB86BE4" FontFamily="Comic Sans MS"/>

<Button x:Name="buttonEnter" Content="Войти" Margin="0 20 0 0" Background="#FF01B905" BorderBrush="#FF01B905" Foreground="White" FontFamily="Comic Sans MS" Click="buttonEnter\_Click"/>

<Button x:Name="buttonReg" Content="войти как гость" Margin="0,10,0,0" HorizontalAlignment="Center" Style="{StaticResource MaterialDesignFlatButton}" Foreground="#FF01B905" FontFamily="Comic Sans MS" BorderBrush="#FF01B905" Click="buttonReg\_Click"/>

</StackPanel>

</Border>

</Grid>

</Window>

xaml.cs:

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;

namespace KursovayaRabota

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

Admin admin = new Admin();

}

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

{

if (textBoxLogin.Text.Length > 0)

{

if(passBox.Password.Length > 0)

{

DataTable dt\_user = avtorizacia.Select("SELECT \* FROM [dbo].[авторизация] WHERE [login] = " +

"'" + textBoxLogin.Text + "' AND [password] = '" + passBox.Password + "'");

if (dt\_user.Rows.Count > 0) // если такая запись существует

{

admin.Show();

this.Close();

}

else

{

MessageBox.Show("Пользователя с такими данными не существует!");

}

}

}

else

{

MessageBox.Show("Пользователя с такими данными не существует!");

}

}

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

{

goZayavka menu = new goZayavka();

menu.Show();

this.Close();

}

}

}

**Меню администратора**

xaml:

<Window x:Class="KursovayaRabota.Admin"

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"

xmlns:local="clr-namespace:KursovayaRabota"

mc:Ignorable="d"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

TextElement.Foreground="{DynamicResource MaterialDesignBody}"

TextElement.FontWeight="Regular"

TextElement.FontSize="13"

TextOptions.TextFormattingMode="Ideal"

TextOptions.TextRenderingMode="Auto"

Background="{DynamicResource MaterialDesignPaper}"

FontFamily="{DynamicResource MaterialDesignFont}"

Title="Меню администратора" Height="450" Width="800" MinHeight="450" MinWidth="800">

<Grid>

<Grid.Background>

<LinearGradientBrush EndPoint="0.5,1" MappingMode="RelativeToBoundingBox" StartPoint="0.5,0">

<GradientStop Color="#FFFF4BA7"/>

<GradientStop Color="#FF56CCFF" Offset="1"/>

</LinearGradientBrush>

</Grid.Background>

<Border MinWidth="400" Margin="15" Background="White" VerticalAlignment="Center" Padding="30" MaxWidth="550" CornerRadius="20">

<Border.Effect>

<DropShadowEffect BlurRadius="10" Color="LightGray" ShadowDepth="0" />

</Border.Effect>

<StackPanel Margin="0,0,0,0">

<Button x:Name="buttonAdminBack" Content="Назад" Margin="0,20,414,0" Background="#FF01B905" BorderBrush="#FF01B905" Foreground="White" FontFamily="Comic Sans MS" Click="buttonAdminBack\_Click"/>

<Image Width="50" HorizontalAlignment="Center" Margin="0 0 0 10" Source="tHWDWmULLcA-architectural-scale-4\_00x.png" />

<Button x:Name="buttonOlimpInfo" Content="Информация об олимпиадах" Margin="0 20 0 0" Background="#FF01B905" BorderBrush="#FF01B905" Foreground="White" FontFamily="Comic Sans MS" Click="buttonOlimpInfo\_Click"/>

<Button x:Name="buttonUchastnikInfo" Content="Информация об участниках" Margin="0 20 0 0" Background="#FF01B905" BorderBrush="#FF01B905" Foreground="White" FontFamily="Comic Sans MS" Click="buttonUchastnikInfo\_Click"/>

</StackPanel>

</Border>

</Grid>

</Window>

xaml.cs:

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;

namespace KursovayaRabota

{

/// <summary>

/// Логика взаимодействия для Admin.xaml

/// </summary>

public partial class Admin : Window

{

public Admin()

{

InitializeComponent();

}

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

{

AdminOlimp adminolimp = new AdminOlimp();

adminolimp.Show();

this.Close();

}

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

{

MainWindow mainwindow = new MainWindow();

mainwindow.Show();

this.Close();

}

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

{

UchastnikiInfo result = new UchastnikiInfo();

result.Show();

this.Close();

}

}

}

**Информация об олимпиадах**

xaml:

<Window x:Class="KursovayaRabota.AdminOlimp"

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"

xmlns:local="clr-namespace:KursovayaRabota"

mc:Ignorable="d"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

TextElement.Foreground="{DynamicResource MaterialDesignBody}"

TextElement.FontWeight="Regular"

TextElement.FontSize="13"

TextOptions.TextFormattingMode="Ideal"

TextOptions.TextRenderingMode="Auto"

Background="{DynamicResource MaterialDesignPaper}"

FontFamily="{DynamicResource MaterialDesignFont}"

Title="Информация об олимпиадах" Height="450" Width="800" MinHeight="450" MinWidth="800">

<Grid>

<Grid.Background>

<LinearGradientBrush EndPoint="0.5,1" MappingMode="RelativeToBoundingBox" StartPoint="0.5,0">

<GradientStop Color="#FFFF4BA7"/>

<GradientStop Color="#FF56CCFF" Offset="1"/>

</LinearGradientBrush>

</Grid.Background>

<Border MinWidth="400" Margin="15" Background="White" VerticalAlignment="Center" Padding="30" MaxWidth="550" CornerRadius="20">

<Border.Effect>

<DropShadowEffect BlurRadius="10" Color="LightGray" ShadowDepth="0" />

</Border.Effect>

<StackPanel Margin="0,0,0,0">

<Grid>

<Button x:Name="buttonAdminOlimpBack" Content="Назад" Margin="0,20,414,20" Background="#FF01B905" BorderBrush="#FF01B905" Foreground="White" FontFamily="Comic Sans MS" Click="buttonAdminBack\_Click"/>

<Button x:Name="buttonAdminOlimpReload" Content="Обновить" Margin="387,20,0,20" Background="#FFAACDFF" BorderBrush="#FFAACDFF" Foreground="White" FontFamily="Comic Sans MS" Click="buttonAdminOlimpReload\_Click"/>

</Grid>

<DataGrid x:Name="dataGridOlimpiad" Height="194" AutoGenerateColumns="False" Loaded="dataGridOlimpiad\_Loaded" IsReadOnly="True">

<DataGrid.Columns>

<DataGridTextColumn Header="ID" Binding="{Binding ID}" Width="50"/>

<DataGridTextColumn Header="Название" Binding="{Binding Название}" Width="\*"/>

<DataGridTextColumn Header="Компетенция" Binding="{Binding Компетенция}" Width="\*"/>

<DataGridTemplateColumn Width="auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Image x:Name="imageOlimpEdit" Source="img\_237015.png" Width="15" Height="15" MouseLeftButtonDown="imageOlimpEdit\_MouseLeftButtonDown"></Image>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<Grid>

<Button x:Name="buttonOlimpAdd" Content="Добавить" Margin="0,20,260,0" Background="#FF01B905" BorderBrush="#FF01B905" Foreground="White" FontFamily="Comic Sans MS" Click="buttonOlimpAdd\_Click" Height="Auto"/>

<Button x:Name="buttonOlimpDelete" Content="Удалить" Margin="260,20,0,0" Background="#FF01B905" BorderBrush="#FF01B905" Foreground="White" FontFamily="Comic Sans MS" Click="buttonOlimpDelete\_Click"/>

</Grid>

</StackPanel>

</Border>

</Grid>

</Window>

xaml.cs:

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.SqlClient;

using System.Data;

namespace KursovayaRabota

{

/// <summary>

/// Логика взаимодействия для AdminOlimp.xaml

/// </summary>

public partial class AdminOlimp : Window

{

public AdminOlimp()

{

InitializeComponent();

dataGridOlimpiad.ItemsSource = KursovayaEntities1.GetContext().Информация\_об\_олимпиадах.ToList();

}

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

{

Admin admin = new Admin();

admin.Show();

this.Close();

}

public void dataGridOlimpiad\_Loaded(object sender, RoutedEventArgs e)

{

}

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

{

}

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

{

addOlimpiad add = new addOlimpiad(null);

add.Show();

}

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

{

// KursovayaEntities1.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload);

dataGridOlimpiad.ItemsSource = KursovayaEntities1.GetContext().Информация\_об\_олимпиадах.ToList();

}

private void imageOlimpEdit\_MouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

addOlimpiad add = new addOlimpiad((sender as Image).DataContext as Информация\_об\_олимпиадах);

add.Show();

}

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

{

var Информация\_об\_олимпиадахForRemoving = dataGridOlimpiad.SelectedItems.Cast<Информация\_об\_олимпиадах>().ToList();

if (MessageBox.Show($"Вы точно хотите удалить следующие {Информация\_об\_олимпиадахForRemoving.Count()} элементов?","Внимание",

MessageBoxButton.YesNo, MessageBoxImage.Question)==MessageBoxResult.Yes)

{

try

{

KursovayaEntities1.GetContext().Информация\_об\_олимпиадах.RemoveRange(Информация\_об\_олимпиадахForRemoving);

KursovayaEntities1.GetContext().SaveChanges();

MessageBox.Show("Данные удалены!");

dataGridOlimpiad.ItemsSource = KursovayaEntities1.GetContext().Информация\_об\_олимпиадах.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

}

}

**Информация об участниках**

xaml:

<Window x:Class="KursovayaRabota.UchastnikiInfo"

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"

xmlns:local="clr-namespace:KursovayaRabota"

mc:Ignorable="d"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

TextElement.Foreground="{DynamicResource MaterialDesignBody}"

TextElement.FontWeight="Regular"

TextElement.FontSize="13"

TextOptions.TextFormattingMode="Ideal"

TextOptions.TextRenderingMode="Auto"

Background="{DynamicResource MaterialDesignPaper}"

FontFamily="{DynamicResource MaterialDesignFont}"

Title="Информация об участниках" Height="450" Width="1060" MinHeight="450" MinWidth="800">

<Grid>

<Grid.Background>

<LinearGradientBrush EndPoint="0.5,1" MappingMode="RelativeToBoundingBox" StartPoint="0.5,0">

<GradientStop Color="#FFFF4BA7"/>

<GradientStop Color="#FF56CCFF" Offset="1"/>

</LinearGradientBrush>

</Grid.Background>

<Border MinWidth="400" Margin="15" Background="White" VerticalAlignment="Center" Padding="30" MaxWidth="1000" CornerRadius="20">

<Border.Effect>

<DropShadowEffect BlurRadius="10" Color="LightGray" ShadowDepth="0" />

</Border.Effect>

<StackPanel Margin="0,0,0,0">

<Grid>

<Button x:Name="buttonAdminOlimpBack" Content="Назад" Margin="0,20,828,20" Background="#FF01B905" BorderBrush="#FF01B905" Foreground="White" FontFamily="Comic Sans MS" Click="buttonAdminBack\_Click"/>

<Button x:Name="buttonAdminOlimpReload" Content="Обновить" Margin="141,20,694,20" Background="#FFAACDFF" BorderBrush="#FFAACDFF" Foreground="White" FontFamily="Comic Sans MS" Click="buttonAdminOlimpReload\_Click"/>

</Grid>

<DataGrid x:Name="dataGridUchastniki" Height="194" AutoGenerateColumns="False" IsReadOnly="True">

<DataGrid.Columns>

<DataGridTextColumn Header="ФИО" Binding="{Binding ФИО}" Width="250"/>

<DataGridTextColumn Header="ID олимпиады" Binding="{Binding ID\_Олимпиады}" Width="\*"/>

<DataGridTextColumn Header="Электронная почта" Binding="{Binding Электронная\_почта}" Width="\*"/>

<DataGridTextColumn Header="Номер телефона" Binding="{Binding Номер\_телефона}" Width="\*"/>

</DataGrid.Columns>

</DataGrid>

<Grid>

<Button x:Name="buttonUchastnikiDelete" Content="Удалить" Margin="187,20,188,0" Background="#FF01B905" BorderBrush="#FF01B905" Foreground="White" FontFamily="Comic Sans MS" Click="buttonOlimpDelete\_Click" VerticalAlignment="Center"/>

</Grid>

</StackPanel>

</Border>

</Grid>

</Window>

xaml.cs:

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;

namespace KursovayaRabota

{

/// <summary>

/// Логика взаимодействия для UchastnikiInfo.xaml

/// </summary>

public partial class UchastnikiInfo : Window

{

public UchastnikiInfo()

{

InitializeComponent();

dataGridUchastniki.ItemsSource = KursovayaEntities1.GetContext().Информация\_об\_участниках.ToList();

}

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

{

var Информация\_об\_участникахForRemoving = dataGridUchastniki.SelectedItems.Cast<Информация\_об\_участниках>().ToList();

if (MessageBox.Show($"Вы точно хотите удалить следующие {Информация\_об\_участникахForRemoving.Count()} элементов?", "Внимание",

MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

KursovayaEntities1.GetContext().Информация\_об\_участниках.RemoveRange(Информация\_об\_участникахForRemoving);

KursovayaEntities1.GetContext().SaveChanges();

MessageBox.Show("Данные удалены!");

dataGridUchastniki.ItemsSource = KursovayaEntities1.GetContext().Информация\_об\_участниках.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

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

{

Admin admin = new Admin();

admin.Show();

this.Close();

}

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

{

dataGridUchastniki.ItemsSource = KursovayaEntities1.GetContext().Информация\_об\_участниках.ToList();

}

}

}

**Окно гостя**

xaml:

<Window x:Class="KursovayaRabota.goZayavka"

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"

xmlns:local="clr-namespace:KursovayaRabota"

mc:Ignorable="d"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

TextElement.Foreground="{DynamicResource MaterialDesignBody}"

TextElement.FontWeight="Regular"

TextElement.FontSize="13"

TextOptions.TextFormattingMode="Ideal"

TextOptions.TextRenderingMode="Auto"

Background="{DynamicResource MaterialDesignPaper}"

FontFamily="{DynamicResource MaterialDesignFont}"

Title="Окно гостя" Height="450" Width="800" MinHeight="450" MinWidth="800">

<Grid>

<Grid.Background>

<LinearGradientBrush EndPoint="0.5,1" MappingMode="RelativeToBoundingBox" StartPoint="0.5,0">

<GradientStop Color="#FFFF4BA7"/>

<GradientStop Color="#FF56CCFF" Offset="1"/>

</LinearGradientBrush>

</Grid.Background>

<Border MinWidth="400" Margin="15" Background="White" VerticalAlignment="Center" Padding="30" MaxWidth="550" CornerRadius="20">

<Border.Effect>

<DropShadowEffect BlurRadius="10" Color="LightGray" ShadowDepth="0" />

</Border.Effect>

<StackPanel Margin="0,0,0,0">

<Grid>

<Button x:Name="buttonAdminOlimpBack" Content="Назад" Margin="0,20,414,20" Background="#FF01B905" BorderBrush="#FF01B905" Foreground="White" FontFamily="Comic Sans MS" Click="buttonAdminOlimpBack\_Click"/>

</Grid>

<DataGrid x:Name="dataGridOlimpiad" Height="194" AutoGenerateColumns="False" Loaded="dataGridOlimpiad\_Loaded" IsReadOnly="True">

<DataGrid.Columns>

<DataGridTextColumn Header="ID" Binding="{Binding ID}" Width="50"/>

<DataGridTextColumn Header="Название" Binding="{Binding Название}" Width="\*"/>

<DataGridTextColumn Header="Компетенция" Binding="{Binding Компетенция}" Width="\*"/>

</DataGrid.Columns>

</DataGrid>

<Grid>

<Button x:Name="buttongozayavka" Content="Подать заявку" Margin="0,20,0,0" Background="#FF01B905" BorderBrush="#FF01B905" Foreground="White" FontFamily="Comic Sans MS" VerticalAlignment="Top" Click="buttongozayavka\_Click"/>

</Grid>

</StackPanel>

</Border>

</Grid>

</Window>

xaml.cs:

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;

namespace KursovayaRabota

{

/// <summary>

/// Логика взаимодействия для goZayavka.xaml

/// </summary>

public partial class goZayavka : Window

{

public goZayavka()

{

InitializeComponent();

}

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

{

dataGridOlimpiad.ItemsSource = KursovayaEntities1.GetContext().Информация\_об\_олимпиадах.ToList();

}

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

{

UchastnikMenu menu = new UchastnikMenu();

menu.Show();

this.Close();

}

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

{

addZayavka add = new addZayavka();

add.Show();

}

}

}