

## ПРАКТИЧЕСКАЯ РАБОТА № 2

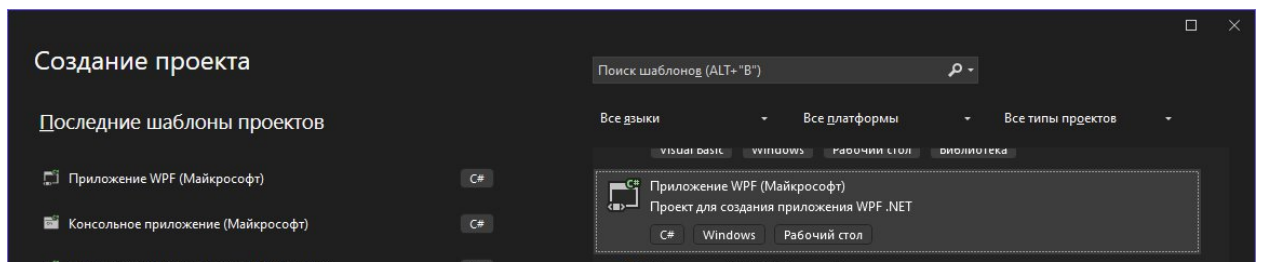
**Тема:** Создание нового проекта с подключением графического фреймворка WPF.

**Цель:** создать каркас приложения с использованием графического фреймворка Windows Presentation Foundation и подключить к нему базу данных.

**Перечень оснащения и оборудования, источников:** ПК, раздаточный материал, среда разработки Microsoft Visual Studio.

**Задание:** создать новый проект с шаблоном WPF и подключить к нему базу данных.

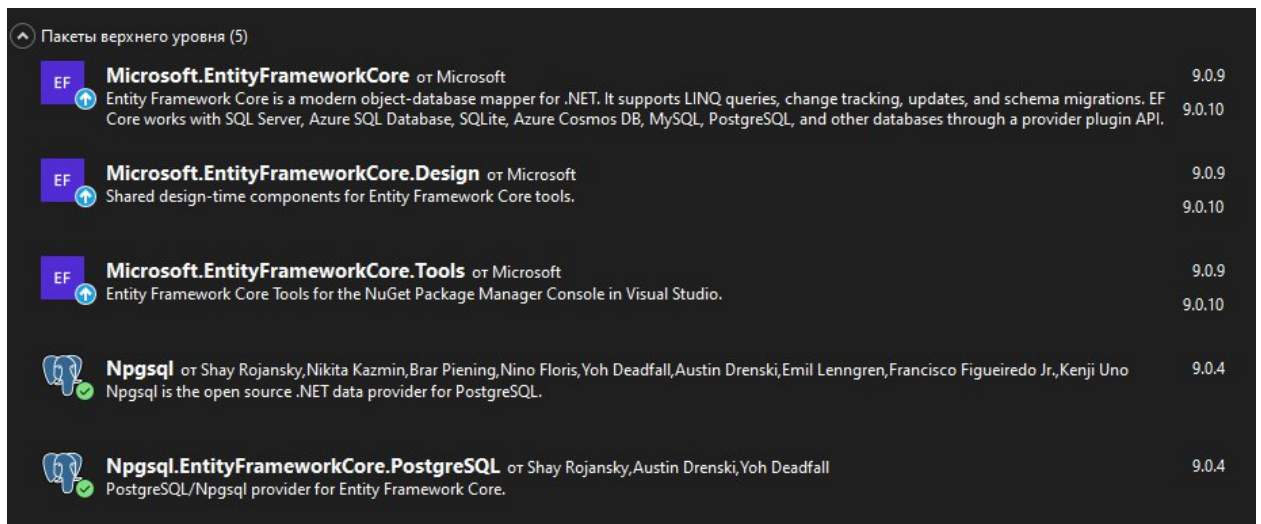
№ 1. Создайте новый проект в Microsoft Visual Studio с использованием графического фреймворка WPF. Для этого в перечне шаблонов найдите вариант «Приложение WPF (Майкрософт)». Остальные шаги создания проекта совпадают с порядком действий, которые Вы проделывали при создании проектов с использованием других шаблонов.



№ 2. Visual Studio автоматически создаст первое окно приложения под названием MainWindow. Перейдите в менеджер пакетов NuGet (или в консоль диспетчера пакетов) и установите необходимые для дальнейшей разработки библиотеки и модули: Npgsql и EntityFramework. Базовый набор библиотек следующий:

1. Microsoft.EntityFrameworkCore
2. Microsoft.EntityFrameworkCore.Design
3. Microsoft.EntityFrameworkCore.Tools
4. Npgsql
5. Npgsql.EntityFrameworkCore.PostgreSQL

Установите дополнительные модули по своему усмотрению.



№ 3. Перенесите все файлы из директории data в ресурсы приложения (заранее создайте папку Resources в проекте). В этой директории хранятся обложки альбомов и некоторые дополнительные файлы, необходимые для оформления приложения.

№ 4. Ознакомьтесь с руководством по стилю и найдите требование по шрифту. На всех окнах должен использоваться единый шрифт и это можно задать в ресурсах приложения (вместо того, чтобы вручную везде его менять).

Откройте файл App.xaml и найдите тег <Application.Resources>. В этом теге можно определять дополнительные ресурсы для форм и задавать глобальные стили оформления.

Создайте новый тег <Style> внутри <Application.Resources>. Внутри этого тега можно писать собственные стили, задавая стандартные значения для свойств элементов на экране. Стил нужно определить для окна, для этого дополните тег атрибутом TargetType и укажите для него значение Window (в кавычках).

Должна получиться следующая структура:

```
<Application x:Class="MusicApp.App"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="clr-namespace:MusicApp"
    StartupUri="MainWindow.xaml">
  <Application.Resources>
    <Style TargetType="Window">
      ...
    </Style>
  </Application.Resources>
</Application>
```

Внутри тега <Style> нужно задать стандартное значение для свойства FontFamily, которое определяет шрифт текста на элементе. Создайте новый тег <Setter> и передайте ему следующие атрибуты:

1. Property="FontFamily" – свойство, которому задаётся значение

2. Value="Times New Roman" – значение, представляющее собой название шрифта.

Должна получиться следующая разметка свойств приложения:

```
<Application x:Class="MusicApp.App"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="clr-namespace:MusicApp"
    StartupUri="MainWindow.xaml">
    <Application.Resources>
        <Style TargetType="Window">
            <Setter Property="FontFamily" Value="Times New Roman"></Setter>
        </Style>
    </Application.Resources>
</Application>
```

Проверьте работу стиля, создав текстовый элемент в MainWindow.

**Самостоятельно:** задайте размер шрифта по умолчанию и иконку приложения.

№ 5. Подключите базу данных к приложению через EntityFramework и для проверки выведите информацию о количестве композиций, альбомов и плейлистов в базе данных. Информацию выведите с помощью текстовых полей (Label или TextBlock) в произвольном виде.

## Руководство по стилю

Все экранные формы пользовательского интерфейса должны иметь заголовок. Кроме того, на главной форме должен быть установлен логотип (представлен в ресурсах). Логотип не искажать: не менять изображение, пропорции изображения, цвет.

Для приложения должна быть установлена иконка (дана в ресурсах), если это реализуемо в рамках платформы.

Интерфейс должен быть адаптивным: корректно отображаться при разных разрешениях экрана.

Использовать шрифт **Times New Roman**, размер не менее 12pt. Допускается использование шрифта **Segoe UI** для цифровых значений и таблиц.

Для интерактивных элементов (кнопок, полей для ввода, карточек и других элементов, которые могут быть задействованы пользователем) задайте скругление не менее 8px.

Цветовая палитра приложения:

Элемент интерфейса	Цвет (HEX-значение)	Пример
Основной фон	#FFFFFF	
Основной акцент	#7A00CC	
Вторичный акцент	#9C4DFF	
Дополнительный фон	#EDE2FF	
Цвет текста (основной)	#1C1C1C	
Цвет текста (вторичный)	#4B4B4B	
Цвет ошибок	#CC3366	
Цвет успеха	#5DCC99	

### *Навигация между окнами*

В верхней части окна должна быть реализована панель навигации вида: *Альбомы | Артисты | Плейлисты | Пользователи | Выход.*

Активный пункт меню подсвечивается соответствующим цветом. Элементы навигации могут быть недоступны (должны быть скрыты) для пользователей, у которых нет достаточного уровня доступа.

### *Макет списка треков в плейлистах*

Карточки треков содержат следующие элементы (расположение в соответствии с порядком их указания в списке ниже):

1. Обложку альбома (миниатюра 128\*128 px).

2. Название трека и исполнителей через запятую (название чуть больше перечня исполнителей).

3. Название альбома.

4. Рейтинг композиции.

5. Длительность трека в формате ЧЧ:ММ:СС (если трек меньше 60 минут, то ММ:СС).

Список треков для альбомов оформляется также, но без указания миниатюры обложки и названия альбома.