

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

КУРСОВАЯ РАБОТА

по теме:

Галерея изображений

по курсу Мобильная разработка

Студент:

Самойлов Д. Р., 221-329

Москва, 2025

ОГЛАВЛЕНИЕ

ОГЛАВЛЕНИЕ.....	2
ВВЕДЕНИЕ.....	3
1. Требования к программе.....	3
1.1. Функциональные требования.....	3
1.2. Нефункциональные требования.....	3
ЛИСТИНГ ПРОЕКТА.....	4
Структура проекта.....	4
Program.cs.....	4
FileInfoProvider.cs.....	4
FolderNavigator.cs.....	4
ImageItem.cs.....	4
FullScreenWindow.xaml.....	4
FullScreenWindow.xaml.cs.....	4
MainWindow.xaml.....	4
MainWindow.xaml.cs.....	4

ВВЕДЕНИЕ

Целью данной курсовой работы является разработка кроссплатформенного мобильного приложения «Галерея изображений». Приложение предоставляет пользователю удобный интерфейс для просмотра изображений в полноэкранном и оконном режимах. Основное внимание уделено разработке архитектуры приложения, обеспечению модульности и удобства использования.

1. Требования к программе

1.1. Функциональные требования

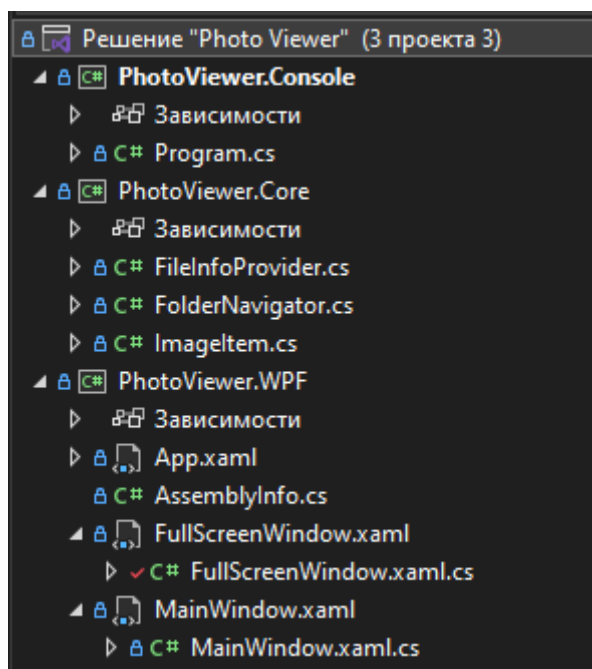
- Загрузка изображений из папки
- Отображение изображений в виде галереи
- Открытие изображения в полноэкранном режиме
- Навигация между изображениями
- Заккрытие полноэкранного режима

1.2. Нефункциональные требования

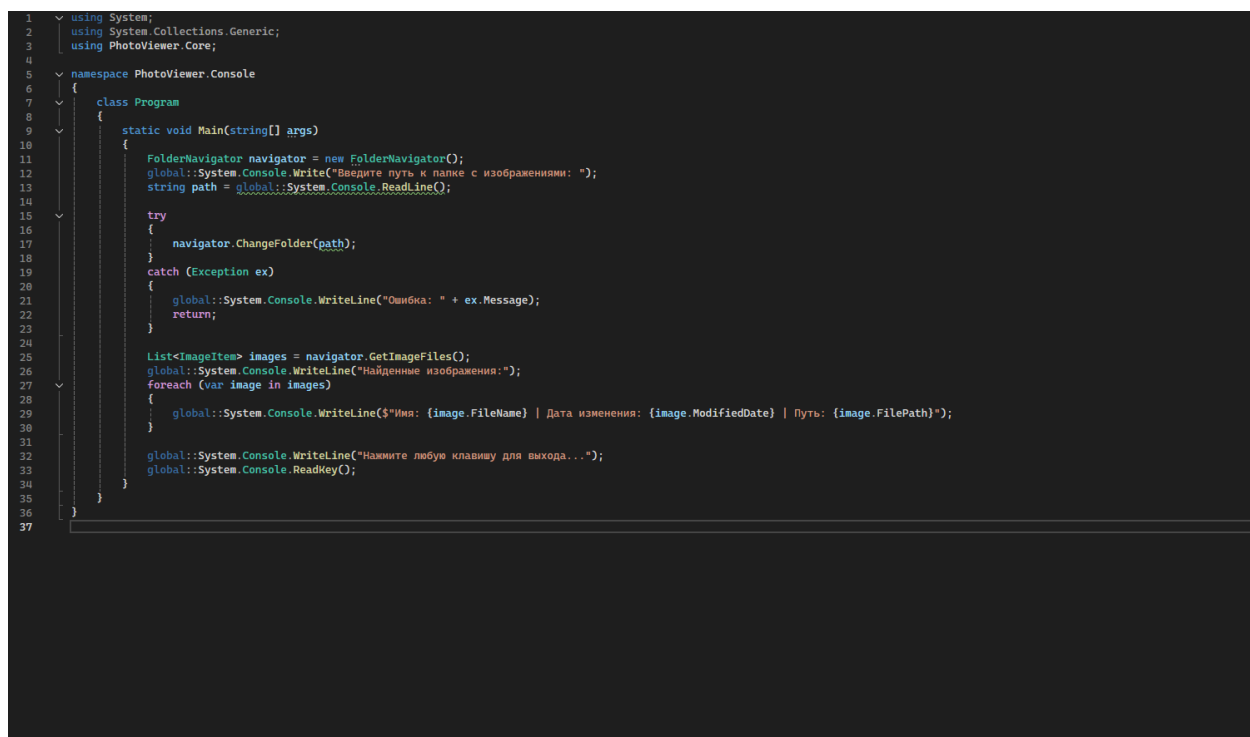
- Программа должна быть реализована с использованием .NET WPF
- Программа должна иметь Console и GUI Реализации

ЛИСТИНГ ПРОЕКТА

Структура проекта



Program.cs



FileInfoProvider.cs

```
1 using System;
2 using System.IO;
3
4 namespace PhotoViewer.Core
5 {
6     public static class FileInfoProvider
7     {
8         // Метод для получения размера файла
9         public static long GetFileSize(string filePath)
10         {
11             if (File.Exists(filePath))
12             {
13                 FileInfo info = new FileInfo(filePath);
14                 return info.Length;
15             }
16             throw new FileNotFoundException("Файл не найден", filePath);
17         }
18     }
19 }
20
```

FolderNavigator.cs

```
1 using System;
2 using System.Collections.Generic;
3 using System.IO;
4 using System.Linq;
5
6 namespace PhotoViewer.Core
7 {
8     public class FolderNavigator
9     {
10         public string CurrentPath { get; set; }
11
12         public delegate void FolderChangedEventHandler(string newPath);
13         public event FolderChangedEventHandler FolderChanged;
14
15         public List<ImageItem> GetImageFiles()
16         {
17             if (string.IsNullOrEmpty(CurrentPath) || !Directory.Exists(CurrentPath))
18                 return new List<ImageItem>();
19
20             var supportedExtensions = new[] { ".jpg", ".jpeg", ".png", ".bmp", ".gif" };
21
22             var files = Directory.GetFiles(CurrentPath)
23                 .Where(file => supportedExtensions.Contains(Path.GetExtension(file).ToLower()))
24                 .Select(file => new ImageItem
25                 {
26                     FilePath = file,
27                     FileName = Path.GetFileName(file),
28                     ModifiedDate = File.GetLastWriteTime(file),
29                     FileSize = FileInfoProvider.GetFileSize(file)
30                 })
31                 .ToList();
32
33             return files;
34         }
35
36         public void ChangeFolder(string newPath)
37         {
38             if (Directory.Exists(newPath))
39             {
40                 CurrentPath = newPath;
41                 FolderChanged?.Invoke(newPath);
42             }
43             else
44             {
45                 throw new DirectoryNotFoundException($"Папка '{newPath}' не найдена.");
46             }
47         }
48     }
49 }
50
```

ImageItem.cs

```
1 using System;
2
3 namespace PhotoViewer.Core
4 {
5     public class ImageItem
6     {
7         public string FilePath { get; set; }
8         public string FileName { get; set; }
9         public DateTime ModifiedDate { get; set; }
10        public long FileSize { get; set; }
11        public string DisplayFileSize => $"{FileSize / 1024} KB";
12    }
13 }
14
```

FullScreenWindow.xaml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <Window x:Class="PhotoViewer.WPF.FullScreenWindow"
3       xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
4       xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
5       WindowStyle="None"
6       WindowState="Maximized"
7       Background="Black"
8       KeyDown="Window_KeyDown">
9     <Grid>
10        <Image x:Name="FullScreenImage" Stretch="Uniform"/>
11        <TextBlock x:Name="FileInfoTextBlock"
12                  HorizontalAlignment="Right"
13                  VerticalAlignment="Bottom"
14                  Margin="10"
15                  Foreground="White"
16                  Background="Black"
17                  Opacity="0.7"
18                  Padding="5"
19                  FontSize="14"/>
20    </Grid>
21 </Window>
```

FullScreenWindow.xaml.cs

```
1  using System;
2  using System.Collections.Generic;
3  using System.Windows;
4  using System.Windows.Input;
5  using System.Windows.Media.Imaging;
6  using PhotoViewer.Core;
7
8  namespace PhotoViewer.WPF
9  {
10     public partial class FullScreenWindow : Window
11     {
12         private List<ImageItem> _images;
13         private int _currentIndex;
14
15         public FullScreenWindow(List<ImageItem> images, int currentIndex)
16         {
17             InitializeComponent();
18             _images = images;
19             _currentIndex = currentIndex;
20             LoadImage();
21             this.Loaded += (s, e) => this.Focus();
22         }
23
24         private void LoadImage()
25         {
26             var imageItem = _images[_currentIndex];
27             try
28             {
29                 BitmapImage bitmap = new BitmapImage();
30                 bitmap.BeginInit();
31                 bitmap.UriSource = new Uri(imageItem.FilePath);
32                 bitmap.CacheOption = BitmapCacheOption.OnLoad;
33                 bitmap.EndInit();
34                 FullScreenImage.Source = bitmap;
35
36                 long fileSize = FileInfoProvider.GetFileSize(imageItem.FilePath);
37                 FileInfoTextBlock.Text = $"{imageItem.FileName} | {fileSize / 1024} KB | {imageItem.ModifiedDate}";
38             }
39             catch (Exception ex)
40             {
41                 MessageBox.Show("Ошибка при загрузке изображения: " + ex.Message);
42                 this.Close();
43             }
44         }
45
46         private void Window_KeyDown(object sender, KeyEventArgs e)
47         {
48             if (e.Key == Key.Escape)
49             {
50                 this.Close();
51             }
52             else if (e.Key == Key.Right)
```

```
53         {
54             if (_currentIndex < _images.Count - 1)
55             {
56                 _currentIndex++;
57                 LoadImage();
58             }
59         }
60         else if (e.Key == Key.Left)
61         {
62             if (_currentIndex > 0)
63             {
64                 _currentIndex--;
65                 LoadImage();
66             }
67         }
68     }
69 }
70
71
```

MainWindow.xaml

```
1 <Window x:Class="PhotoViewer.WPF.MainWindow"
2       xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3       xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4       Title="Photo Viewer" Height="600" Width="800">
5     <Grid Margin="10">
6       <Grid.RowDefinitions>
7         <RowDefinition Height="Auto"/>
8         <RowDefinition Height="*/>
9         <RowDefinition Height="Auto"/>
10      </Grid.RowDefinitions>
11      <!-- Выбор папки -->
12      <StackPanel Orientation="Horizontal" Grid.Row="0" Margin="0,0,0,10">
13        <TextBox x:Name="FolderPathTextBox" Width="500" Margin="0,0,10,0"/>
14        <Button Content="Открыть" Click="OpenFolder_Click" Width="100"/>
15      </StackPanel>
16      <!-- Список изображений с автоматической прокруткой -->
17      <ListView x:Name="ImagesListView" Grid.Row="1"
18        SelectionChanged="ImagesListView_SelectionChanged"
19        MouseDoubleClick="ImagesListView_MouseDoubleClick">
20        <ListView.View>
21          <GridView>
22            <GridViewColumn Header="Имя файла" DisplayMemberBinding="{Binding FileName}" Width="250"/>
23            <GridViewColumn Header="Размер" DisplayMemberBinding="{Binding DisplayFileSize}" Width="100"/>
24            <GridViewColumn Header="Дата изменения" DisplayMemberBinding="{Binding ModifiedDate}" Width="150"/>
25          </GridView>
26        </ListView.View>
27      </ListView>
28      <!-- Превью изображения с корректным масштабированием -->
29      <Image x:Name="PreviewImage" Grid.Row="2" Height="200" Margin="0,10,0,0" Stretch="Uniform"/>
30    </Grid>
31  </Window>
```

MainWindow.xaml.cs

```
53 if (ImagesListView.SelectedItem is ImageItem selectedItem)
54 {
55     try
56     {
57         BitmapImage bitmap = new BitmapImage();
58         bitmap.BeginInit();
59         bitmap.UriSource = new Uri(selectedItem.FilePath);
60         bitmap.CacheOption = BitmapCacheOption.OnLoad;
61         bitmap.EndInit();
62         PreviewImage.Source = bitmap;
63     }
64     catch (Exception ex)
65     {
66         MessageBox.Show("Ошибка при загрузке предпросмотра: " + ex.Message);
67         PreviewImage.Source = null;
68     }
69 }
70 else
71 {
72     PreviewImage.Source = null;
73 }
74
75 private void ImagesListView_MouseDoubleClick(object sender, MouseButtonEventArgs e)
76 {
77     if (ImagesListView.SelectedItem is ImageItem selectedItem)
78     {
79         var images = ImagesListView.ItemsSource as List<ImageItem>;
80         int currentIndex = images.IndexOf(selectedItem);
81         FullScreenWindow fullScreenWindow = new FullScreenWindow(images, currentIndex);
82         fullScreenWindow.Show();
83     }
84 }
85
86
87
88 }
```



```

53         if (ImagesListView.SelectedItem is ImageItem selectedItem)
54         {
55             try
56             {
57                 BitmapImage bitmap = new BitmapImage();
58                 bitmap.BeginInit();
59                 bitmap.UriSource = new Uri(selectedItem.FilePath);
60                 bitmap.CacheOption = BitmapCacheOption.OnLoad;
61                 bitmap.EndInit();
62                 PreviewImage.Source = bitmap;
63             }
64             catch (Exception ex)
65             {
66                 MessageBox.Show("Ошибка при загрузке предпросмотра: " + ex.Message);
67                 PreviewImage.Source = null;
68             }
69         }
70     else
71     {
72         PreviewImage.Source = null;
73     }
74 }
75
76 private void ImagesListView_MouseDoubleClick(object sender, MouseButtonEventArgs e)
77 {
78     if (ImagesListView.SelectedItem is ImageItem selectedItem)
79     {
80         var images = ImagesListView.ItemsSource as List<ImageItem>;
81         int currentIndex = images.IndexOf(selectedItem);
82         FullScreenWindow fullScreenWindow = new FullScreenWindow(images, currentIndex);
83         fullScreenWindow.Show();
84     }
85 }
86
87 }
88

```