using System;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.ComponentModel;

using System.IO;

using System.Linq;

using System.Windows;

namespace DuplicateFileFinder

{

public class FileDuplicateInfo

{

public string Name { get; set; }

public string Path { get; set; }

public long Size { get; set; }

public DateTime LastModified { get; set; }

}

public class MainViewModel : INotifyPropertyChanged

{

private string \_selectedDirectory;

private FileDuplicateInfo \_selectedFile;

public string SelectedDirectory

{

get => \_selectedDirectory;

set { \_selectedDirectory = value; OnPropertyChanged(nameof(SelectedDirectory)); }

}

public ObservableCollection<FileDuplicateInfo> DuplicateFiles { get; set; } = new();

public FileDuplicateInfo SelectedFile

{

get => \_selectedFile;

set { \_selectedFile = value; OnPropertyChanged(nameof(SelectedFile)); }

}

// Фильтры поиска

public bool SearchByName { get; set; } = true;

public bool SearchBySize { get; set; } = false;

public bool SearchByDate { get; set; } = false;

public void SelectFolder()

{

var dialog = new System.Windows.Forms.FolderBrowserDialog();

if (dialog.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

SelectedDirectory = dialog.SelectedPath;

FindDuplicateFiles();

}

}

private void FindDuplicateFiles()

{

DuplicateFiles.Clear();

if (string.IsNullOrEmpty(SelectedDirectory) || !Directory.Exists(SelectedDirectory))

{

MessageBox.Show("Выберите папку для поиска!");

return;

}

var files = Directory.GetFiles(SelectedDirectory, "\*.\*", SearchOption.AllDirectories)

.Select(f => new FileInfo(f))

.ToList();

var groupedFiles = files.GroupBy(f => new

{

Name = SearchByName ? f.Name : null,

Size = SearchBySize ? f.Length : (long?)null,

Date = SearchByDate ? f.LastWriteTime.Date : (DateTime?)null

});

foreach (var group in groupedFiles)

{

if (group.Count() > 1)

{

foreach (var file in group)

{

DuplicateFiles.Add(new FileDuplicateInfo

{

Name = file.Name,

Path = file.FullName,

Size = file.Length,

LastModified = file.LastWriteTime

});

}

}

}

if (DuplicateFiles.Count == 0)

{

MessageBox.Show("Дубликаты не найдены.");

}

}

public void OpenFile()

{

if (SelectedFile != null)

{

System.Diagnostics.Process.Start("explorer.exe", $"/select,\"{SelectedFile.Path}\"");

}

}

public void DeleteFile()

{

if (SelectedFile != null && MessageBox.Show("Удалить этот файл?", "Подтверждение", MessageBoxButton.YesNo) == MessageBoxResult.Yes)

{

try

{

File.Delete(SelectedFile.Path);

DuplicateFiles.Remove(SelectedFile);

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка удаления: {ex.Message}");

}

}

}

public event PropertyChangedEventHandler PropertyChanged;

protected void OnPropertyChanged(string name) => PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(name));

}

}

<Window x:Class="DuplicateFileFinder.MainWindow"

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

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

Title="Поиск дубликатов файлов" Height="500" Width="800">

<Window.DataContext>

<local:MainViewModel/>

</Window.DataContext>

<Grid>

<DockPanel>

<!-- Верхнее меню -->

<Menu DockPanel.Dock="Top">

<MenuItem Header="Файл">

<MenuItem Header="Выбрать папку" Click="SelectFolder\_Click"/>

<MenuItem Header="Выход" Click="Exit\_Click"/>

</MenuItem>

</Menu>

<!-- Панель фильтров -->

<StackPanel DockPanel.Dock="Top" Orientation="Horizontal" Margin="10">

<TextBlock Text="Фильтры: " VerticalAlignment="Center"/>

<CheckBox Content="По имени" IsChecked="{Binding SearchByName}"/>

<CheckBox Content="По размеру" IsChecked="{Binding SearchBySize}"/>

<CheckBox Content="По дате изменения" IsChecked="{Binding SearchByDate}"/>

<Button Content="Обновить" Click="SelectFolder\_Click" Margin="5"/>

</StackPanel>

<!-- Таблица с файлами -->

<DataGrid ItemsSource="{Binding DuplicateFiles}" SelectedItem="{Binding SelectedFile}"

AutoGenerateColumns="False" Margin="10">

<DataGrid.Columns>

<DataGridTextColumn Header="Имя" Binding="{Binding Name}" Width="200"/>

<DataGridTextColumn Header="Путь" Binding="{Binding Path}" Width="300"/>

<DataGridTextColumn Header="Размер (KB)" Binding="{Binding Size}" Width="100"/>

<DataGridTextColumn Header="Дата изменения" Binding="{Binding LastModified}" Width="150"/>

</DataGrid.Columns>

</DataGrid>

<!-- Кнопки действий -->

<StackPanel DockPanel.Dock="Bottom" Orientation="Horizontal" HorizontalAlignment="Center" Margin="10">

<Button Content="Открыть" Click="OpenFile\_Click" Width="100"/>

<Button Content="Удалить" Click="DeleteFile\_Click" Width="100" Margin="10,0,0,0"/>

</StackPanel>

</DockPanel>

</Grid>

</Window>

using System.Windows;

namespace DuplicateFileFinder

{

public partial class MainWindow : Window

{

private MainViewModel ViewModel => DataContext as MainViewModel;

public MainWindow()

{

InitializeComponent();

}

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

{

ViewModel?.SelectFolder();

}

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

{

ViewModel?.OpenFile();

}

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

{

ViewModel?.DeleteFile();

}

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

{

Application.Current.Shutdown();

}

}

}