# Лабораторная работа № 1

Тема: Создание приложения «Микропроводник».

Цель работы: получение практических навыков при работе с пространством имен System.IO.

Задание 1. Разработать приложение «Микропроводник», примерный вид которого представлен на Рисунке 1.

Рисунок 1 – Приложение Микропроводник

На форме список всех дисков загружается в компонент comboBox1. Список всех каталогов для данного диска загружается в listBox1. Список файлов, находящихся в выбранном каталоге, отображается listBox2.

Задание 2. Используя дополнительные компоненты,

* для выделенного диска необходимо выводить сведения: объем диска, свободное пространство;
* для выделенного каталога: полное название каталога, время создания каталога, корневой каталог.

Задание 3. При выделении файла в списке должно запускаться соответствующее приложение.

Задание 4. Сохранить в отдельный текстовый файл имена файлов, которые открывались за последние 10 секунд работы приложения.

Примечание. При выполнении задания необходимо работать с типом DateTime.

DateTime.Now – возвращает текущее время;

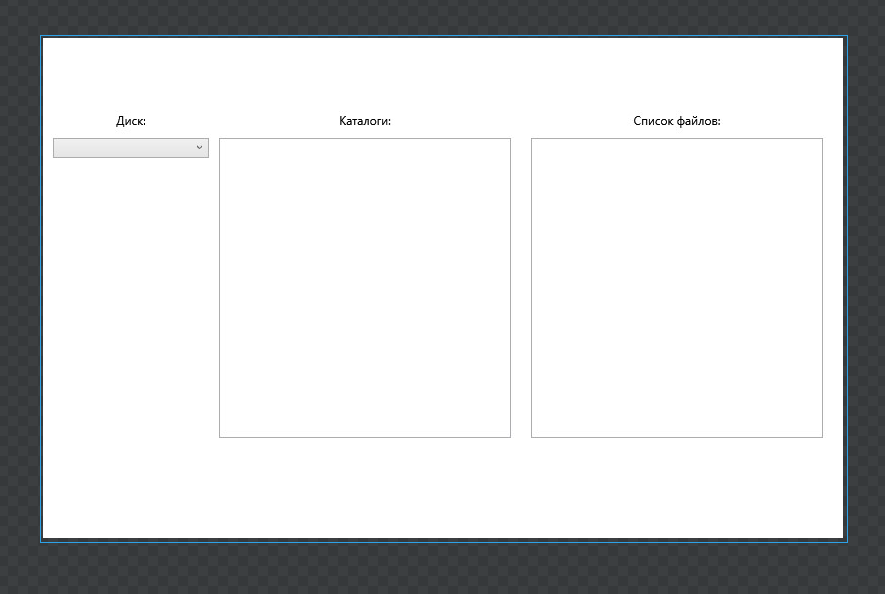
Convert.ToDateTime – преобразование строки в тип DateTime.

Самостоятельно необходимо разобраться как работать с секундами.

Решение:

Разметка:

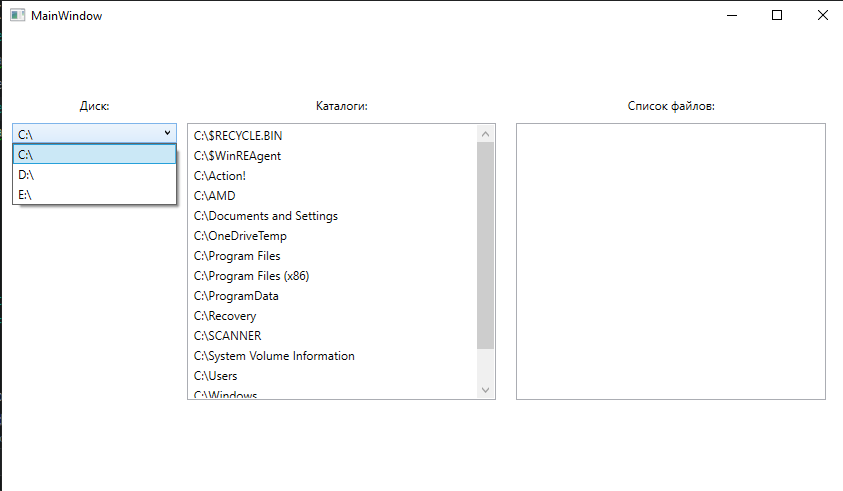
<Window x:Class="WorkWithFiles.MainWindow"  
 xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  
 xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  
 xmlns:d="http://schemas.microsoft.com/expression/blend/2008"  
 xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"  
 mc:Ignorable="d"  
 Title="MainWindow" Height="500" Width="800"  
 MinHeight="500" MinWidth="700">  
 <Grid Margin="10, 0">  
 <Grid.ColumnDefinitions>  
 <ColumnDefinition Width="\*" />  
 <ColumnDefinition Width="2\*" />  
 <ColumnDefinition Width="2\*" />  
 </Grid.ColumnDefinitions>  
 <Grid.RowDefinitions>  
 <RowDefinition Height="\*" />  
 <RowDefinition Height="3\*" />  
 <RowDefinition Height="\*" />  
 </Grid.RowDefinitions>  
 <TextBlock TextAlignment="Center" VerticalAlignment="Bottom" Margin="10">Диск:</TextBlock>  
 <StackPanel Grid.Row="1" Grid.Column="0" Orientation="Vertical">  
 <ComboBox SelectionChanged="HardDrives\_OnSelected" Name="HardDrivesComboBox" Height="20"  
 VerticalAlignment="Top" />  
 <TextBlock Name="HardDriveInformation" TextWrapping="Wrap" />  
 </StackPanel>  
 <TextBlock Grid.Row="0" Grid.Column="1" TextAlignment="Center" VerticalAlignment="Bottom" Margin="10">Каталоги:</TextBlock>  
 <TextBlock Margin="10" Grid.Row="2" Grid.Column="1" Name="FolderInformation" DockPanel.Dock="Bottom"  
 TextWrapping="Wrap" />  
 <ListBox Grid.Row="1" Grid.Column="1" VerticalAlignment="Stretch"  
 SelectionChanged="FoldersListBox\_OnSelectionChanged" Name="FoldersListBox"  
 Margin="10, 0" />  
  
 <TextBlock Grid.Row="0" Grid.Column="2" TextAlignment="Center" VerticalAlignment="Bottom" Margin="10">Список файлов:</TextBlock>  
 <ListBox Grid.Column="2" Name="FilesListBox" Grid.Row="1" SelectionMode="Single"  
 SelectionChanged="FilesListBox\_OnSelectionChanged"  
 Margin="10, 0" />  
 </Grid>  
</Window>



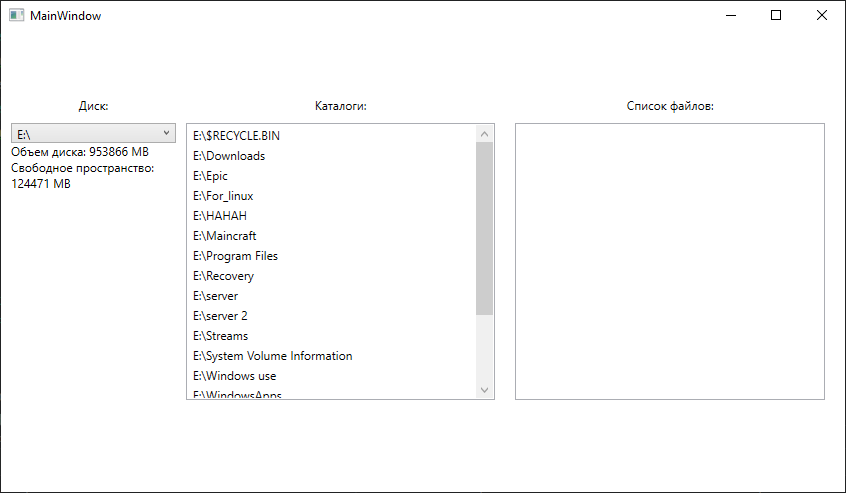
Код приложения:

using System;  
using System.Collections.Generic;  
using System.Diagnostics;  
using System.IO;  
using System.Linq;  
using System.Windows;  
using System.Windows.Controls;  
  
namespace WorkWithFiles;  
  
/// <summary>  
/// Interaction logic for MainWindow.xaml  
/// </summary>  
public partial class MainWindow : Window  
{  
 public MainWindow()  
 {  
 // Создаём и очищаем файл  
 var streamWriter = new StreamWriter("files.txt");  
 streamWriter.Write("");  
 streamWriter.Close();  
 InitializeComponent();  
 // Заполняем поле дисков  
 foreach (var drive in DriveInfo.GetDrives())  
 HardDrivesComboBox.Items.Add(drive.Name);  
 HardDrivesComboBox.SelectedIndex = 0;  
 }  
  
 private void HardDrives\_OnSelected(object sender, RoutedEventArgs e)  
 {  
 FoldersListBox.Items.Clear();  
 FilesListBox.Items.Clear();  
 var hardDrive = HardDrivesComboBox.SelectedItem.ToString()!;  
 foreach (var directory in Directory.GetDirectories(hardDrive))  
 FoldersListBox.Items.Add(directory);  
 // Получаем информацию о диске  
 var driveInfo = DriveInfo.GetDrives().Where(info => info.Name == hardDrive).ToList()[0];  
 HardDriveInformation.Text = @$"Объем диска: {driveInfo.TotalSize / (1024 \* 1024)} MB  
Свободное пространство: {driveInfo.TotalFreeSpace / (1024 \* 1024)} MB";  
 }  
  
  
 private void FoldersListBox\_OnSelectionChanged(object sender, SelectionChangedEventArgs e)  
 {  
 FilesListBox.Items.Clear();  
 FolderInformation.Text = "";  
 try  
 {  
 var path = FoldersListBox.SelectedItem.ToString()!;  
 var directoryInfo = new DirectoryInfo(path);  
 FolderInformation.Text = @$"Полное название каталога: {directoryInfo.Name}  
Время создания каталога: {directoryInfo.CreationTime}  
Корневой каталог: {directoryInfo.Parent?.Name ?? string.Empty}";  
 foreach (var file in Directory.GetFiles(path))  
 FilesListBox.Items.Add(file);  
 }  
 catch (NullReferenceException)  
 {  
 /\* При очистке поле Selection сбрасывается и нас выкидывает сюда, так как стоит проверка на null \*/  
 }  
 catch (UnauthorizedAccessException)  
 {  
 MessageBox.Show("Доступ запрещён к этой папке.");  
 }  
 }  
  
 /\* На открытие любого файла или при закрытии приложения  
 производим проверку все ли записанные файлы, открывались 10 секунд назад  
 \*/  
 private void fileOpened(string? path = null)  
 {  
 HashSet<(string, DateTime)> notDelete = new();  
 if (path != null)  
 notDelete.Add((path, DateTime.Now));  
 var streamReader = new StreamReader("files.txt");  
 while (!streamReader.EndOfStream)  
 {  
 var line = streamReader.ReadLine()!;  
 if (line.Trim() != "")  
 {  
 var runnedTime = DateTime.Parse(line.Split(',')[1]);  
 if (DateTime.Now - runnedTime < TimeSpan.FromSeconds(10))  
 notDelete.Add((line.Split(',')[0], runnedTime));  
 }  
 }  
  
 streamReader.Close();  
 var streamWriter = new StreamWriter("files.txt");  
 foreach (var pair in notDelete)  
 streamWriter.WriteLine($"{pair.Item1},{pair.Item2}");  
 streamWriter.Close();  
 }  
  
 protected override void OnClosed(EventArgs e)  
 {  
 fileOpened();  
 base.OnClosed(e);  
 }  
  
 private void FilesListBox\_OnSelectionChanged(object sender, SelectionChangedEventArgs e)  
 {  
 try  
 {  
 var path = FilesListBox.SelectedItem.ToString()!;  
 new Process  
 {  
 StartInfo = new ProcessStartInfo(path)  
 {  
 UseShellExecute = true  
 }  
 }.Start();  
 fileOpened(path);  
 FilesListBox.SelectedItem = null;  
 }  
 catch (NullReferenceException)  
 {  
 }  
 }  
}

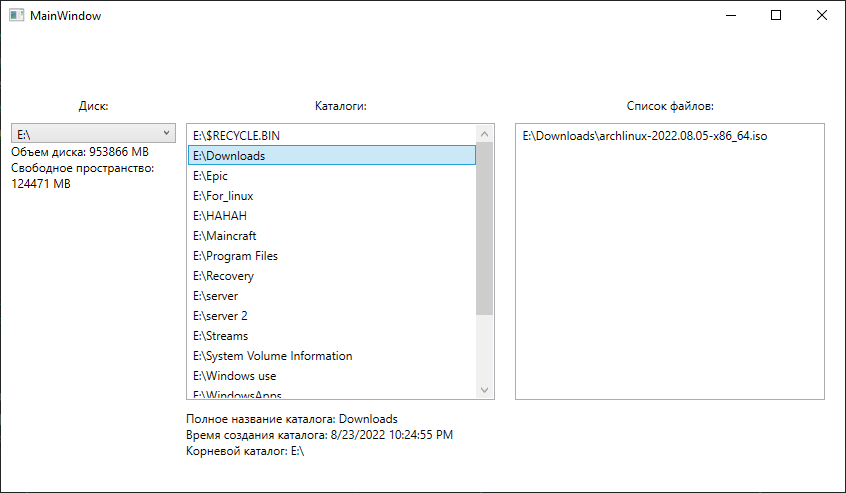
Демонстрация работы приложения:



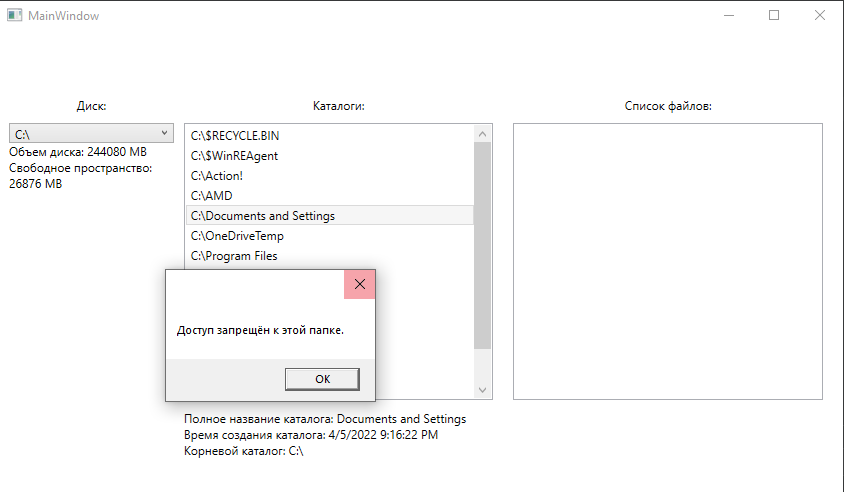
Выбран диск:



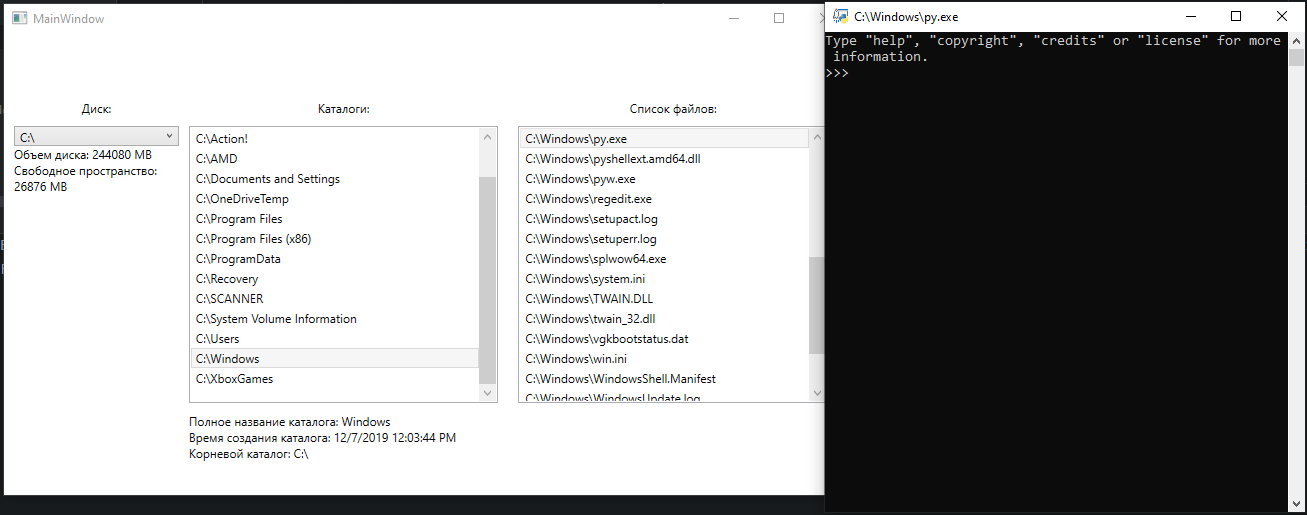
Выбрана папка:



Обработка ошибки с доступом:



Открытие файла(py.exe)



Запись открытия данного файла:



Итог работы:

Получил практические навыки при работе с пространством имен System.IO, а также создал приложение "Микропроводник".