ГУАП

ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

ОТЧЁТ  
ЗАЩИЩЕН С ОЦЕНКОЙ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| преподаватель |  |  |  | И. А. Юрьева |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЁТ О ЛАБОРАТОРНЫХ РАБОТАХ |
|  |
| по дисциплине МДК 01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | С022 |  |  |  | М. А. Соколов |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

**СОДЕРЖАНИЕ**

[Лабораторная работа №1 3](#_Toc114559349)

# 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.

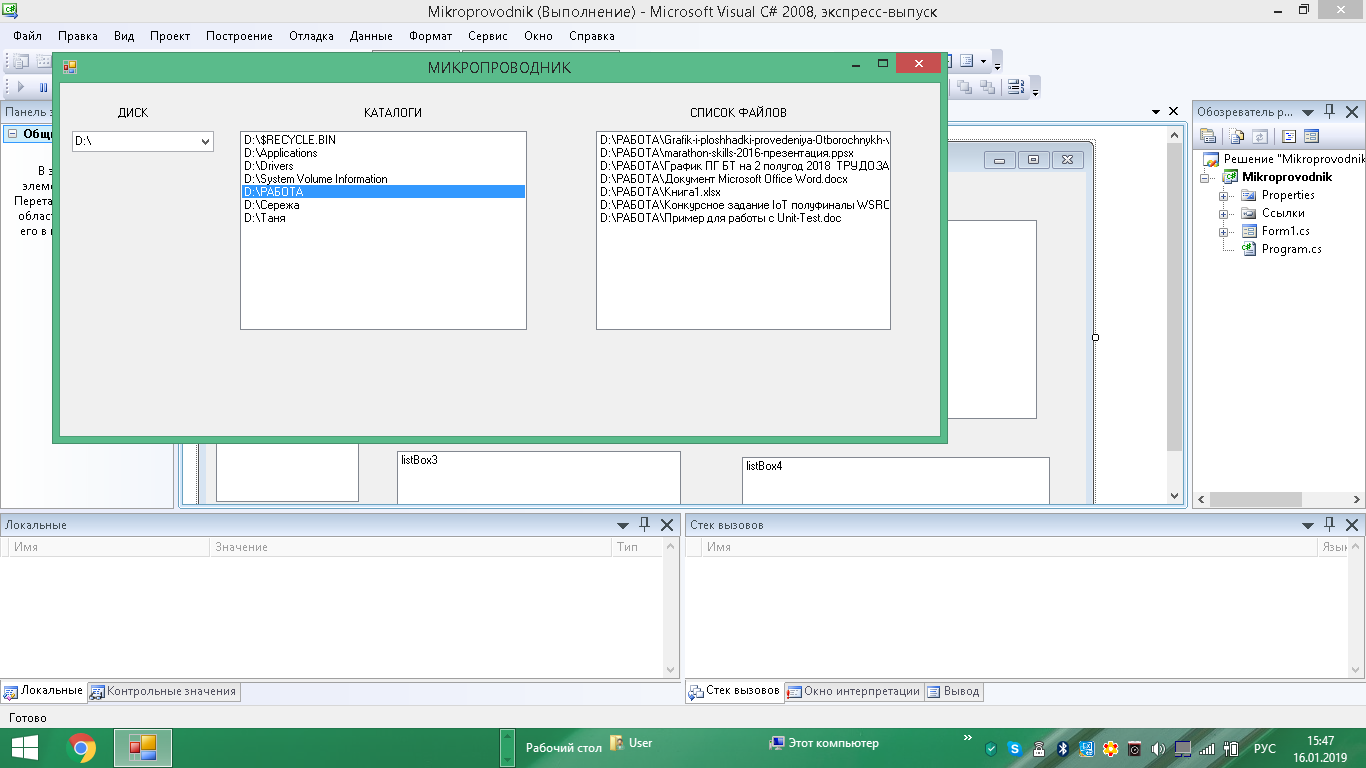


Рисунок №1 – Приложение Микропроводник

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.

**Скриншоты приложения:**

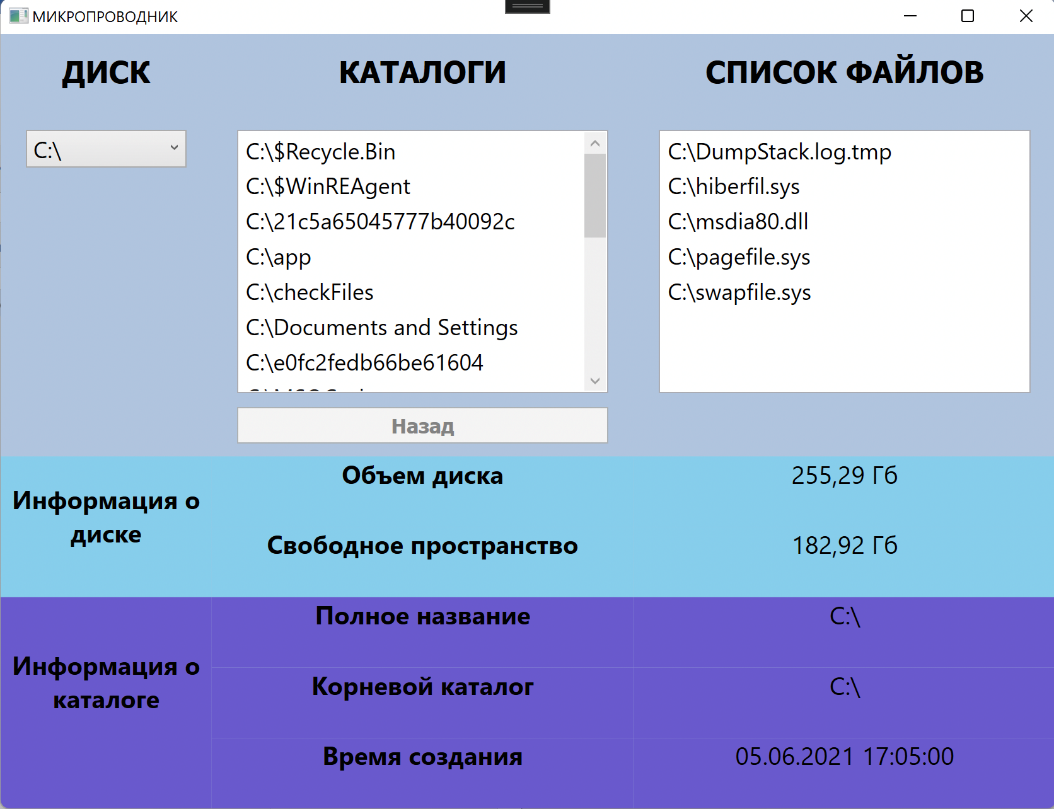
****

Рисунок 1– Начальный экран приложения.

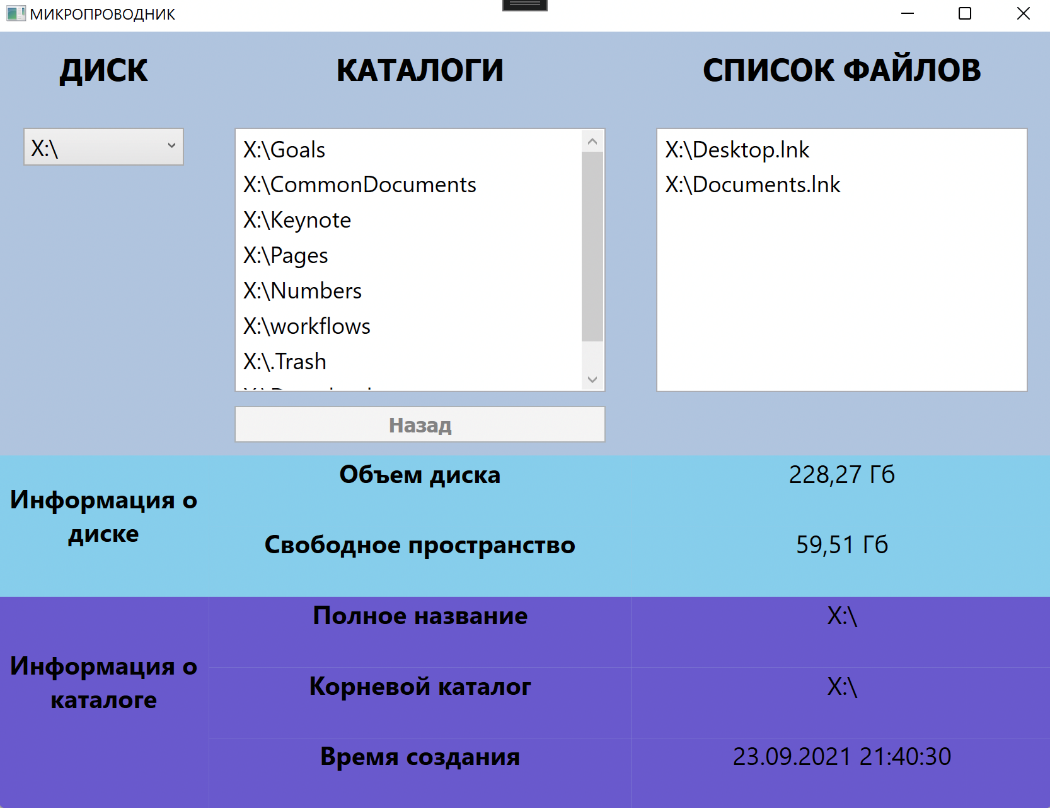


Рисунок 2– Выбор диска. Выводится информация о диске,

каталоги диска и файлы.

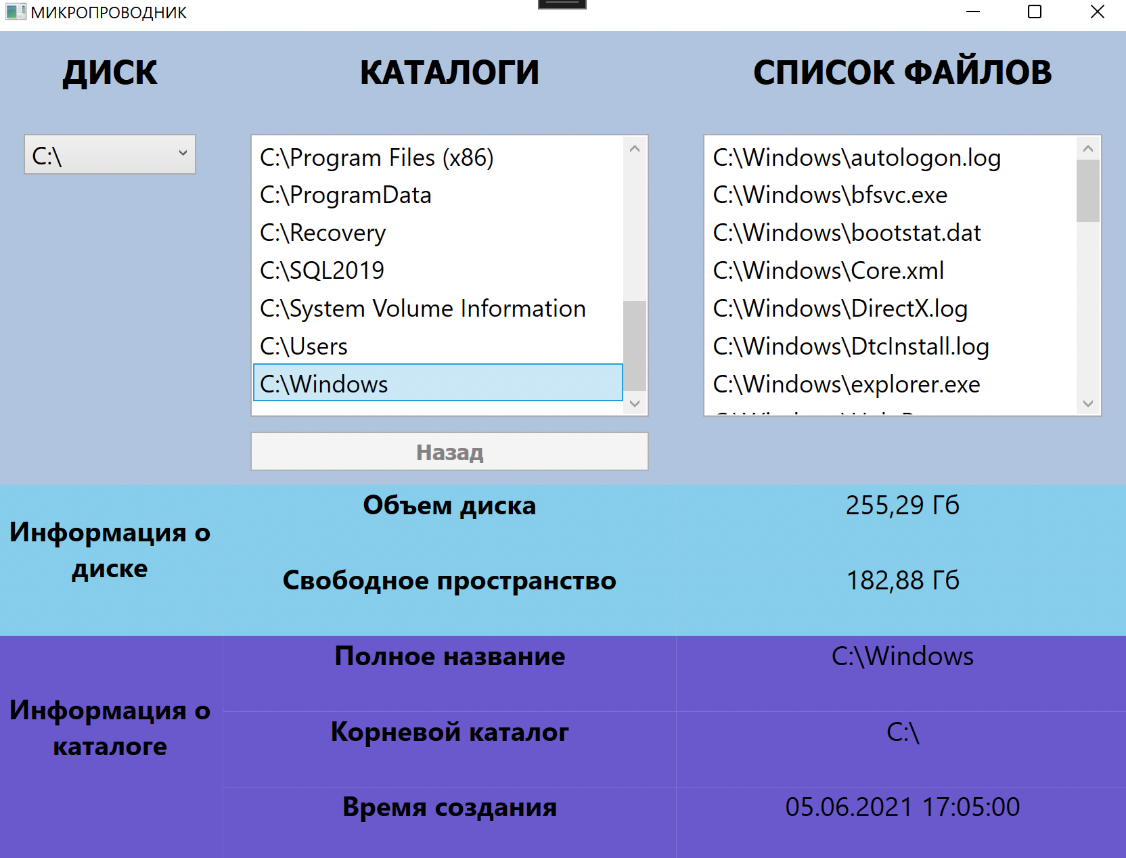


Рисунок 3– Выбор каталога. При одном клике на каталог выводится информация о нем, при двойном клике открывается список подкаталогов и файлы выбранного каталога.

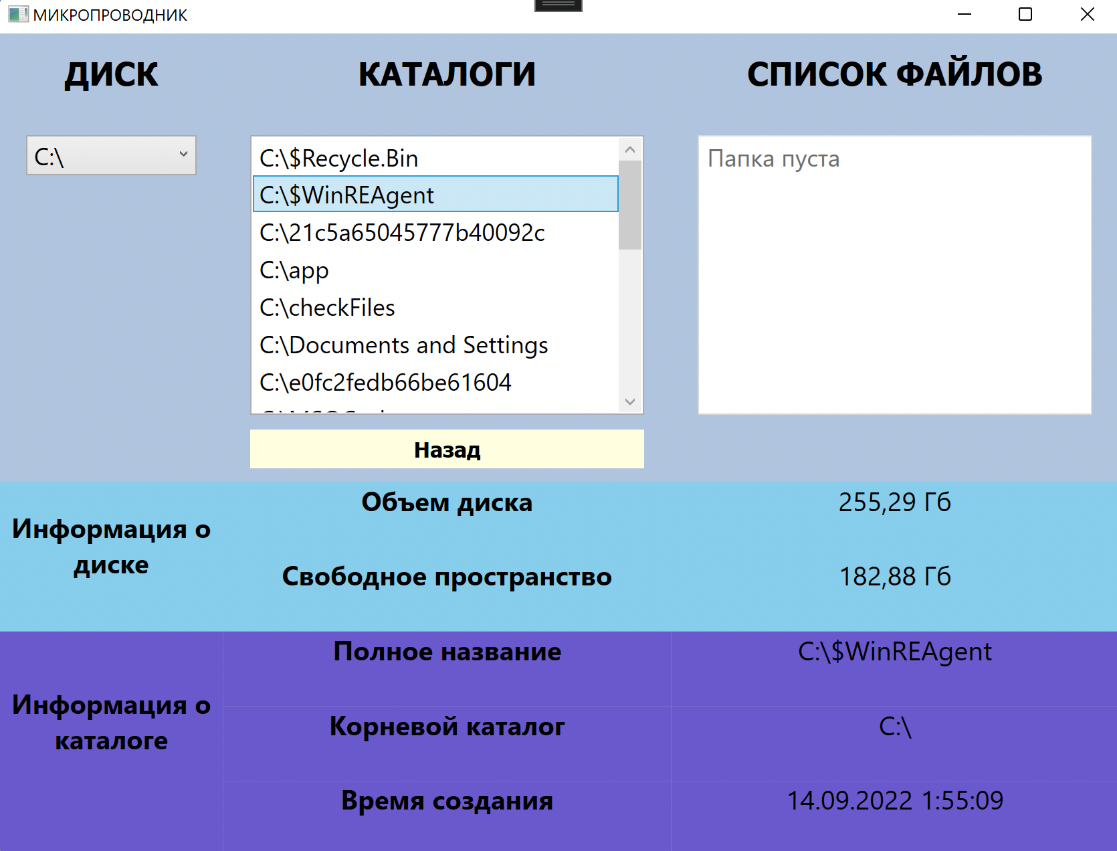


Рисунок 4– Обработка отсутствия файлов в каталоге.

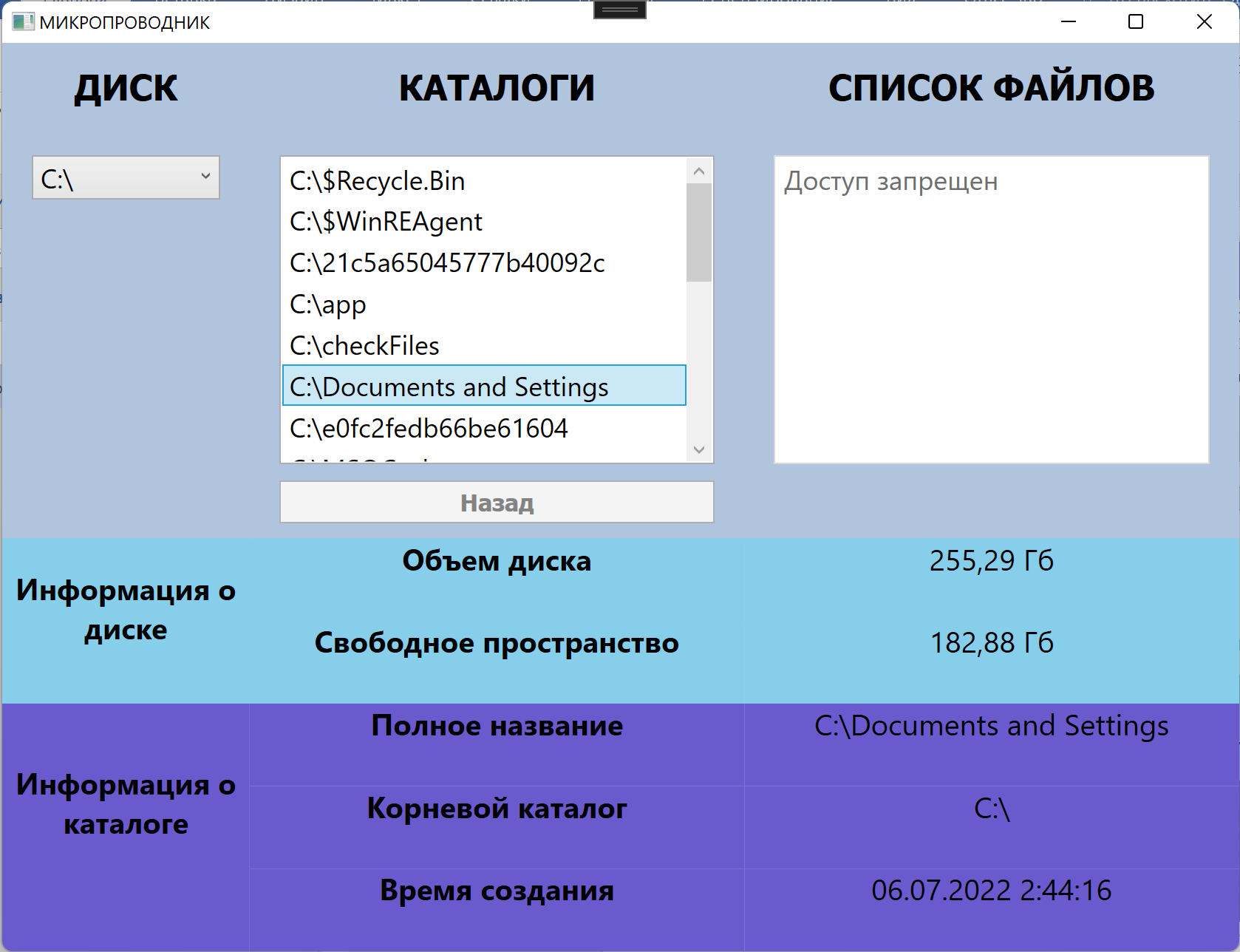


Рисунок 5– Обработка отсутствия прав доступа.

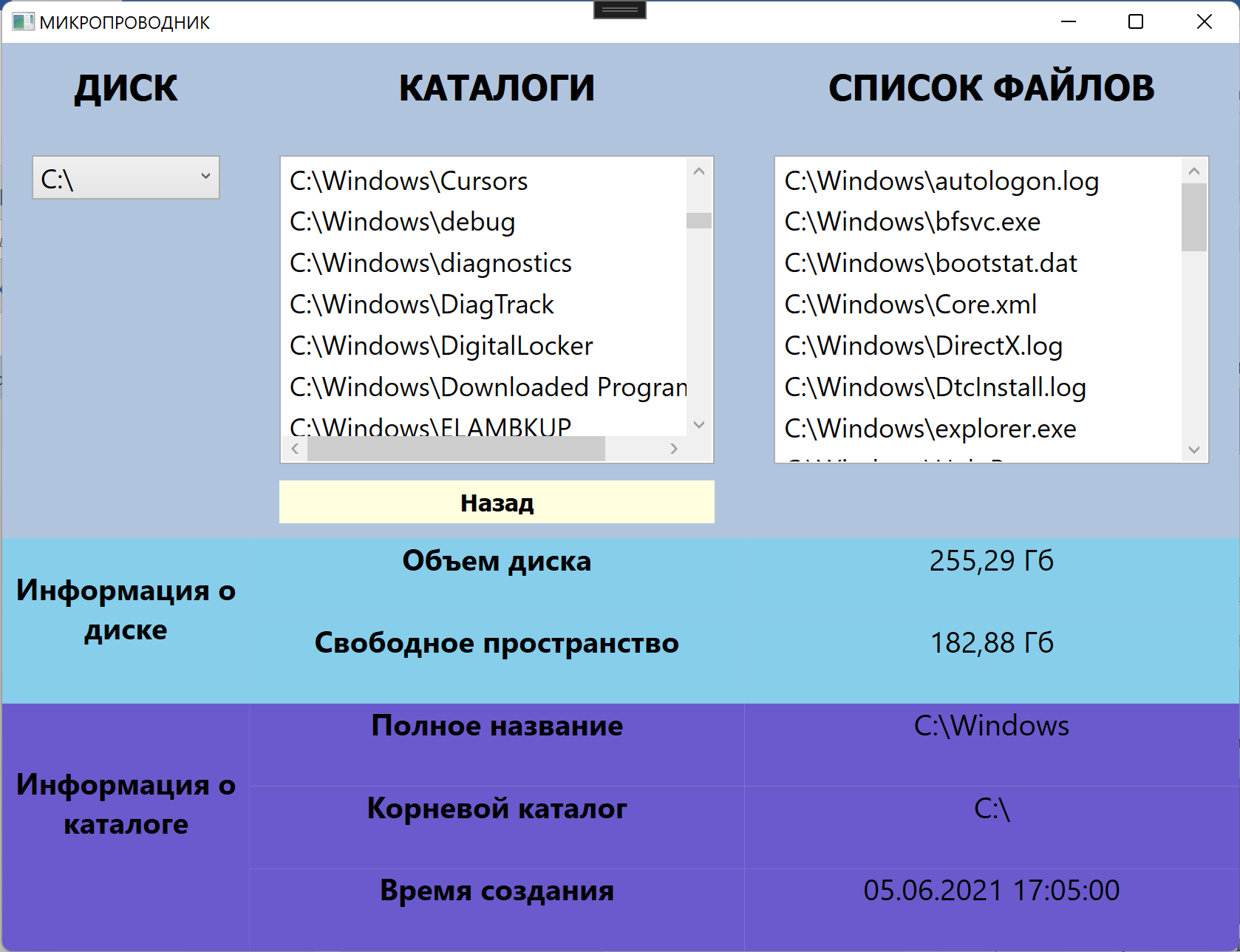


Рисунок 6– При нажатии на стрелку «Назад» пользователь поднимается по дереву файлов вверх.

**Листинг файла MainWindow.xaml:**

<Window x:Class="Lab1Task.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:Lab1Task"

mc:Ignorable="d"

Title="МИКРОПРОВОДНИК" Height="650" Width="850"

Closing="Window\_Closing">

<Grid Background="LightSteelBlue">

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition />

<ColumnDefinition Width="2\*"/>

<ColumnDefinition Width="2\*"/>

</Grid.ColumnDefinitions>

<StackPanel >

<TextBlock TextAlignment="Center"

FontSize="24"

Margin="0,15,0,0"

FontFamily="Tahoma"

FontWeight="Bold">

ДИСК

</TextBlock>

</StackPanel>

<StackPanel Grid.Row="0" Grid.Column="1">

<TextBlock TextAlignment="Center"

FontSize="24"

Margin="0,15,0,0"

FontFamily="Tahoma"

FontWeight="Bold">

КАТАЛОГИ

</TextBlock>

</StackPanel>

<StackPanel Grid.Row="0" Grid.Column="2">

<TextBlock TextAlignment="Center"

FontSize="24"

Margin="0,15,0,0"

FontFamily="Tahoma"

FontWeight="Bold">

СПИСОК ФАЙЛОВ

</TextBlock>

</StackPanel>

<StackPanel Grid.Row="1" Grid.Column="0"

>

<ComboBox Margin="20,20,20,10"

Height="30"

Name="comboBox1"

SelectionChanged="comboBox1\_SelectionChanged"

Grid.Row="0">

</ComboBox>

</StackPanel>

<StackPanel >

</StackPanel>

<ListBox Name="listBox1"

Background="White"

FontSize="18"

Grid.Row="1"

Grid.Column="1"

Margin="20,20,20,50"

SelectionChanged="listBox1\_SelectionChanged"

MouseDoubleClick="listBox1\_MouseDoubleClick">

</ListBox>

<Button Name="backButton"

Grid.Row="1"

Grid.Column="1"

Margin="20,240,20,10"

Content="Назад"

Background="LightYellow"

BorderBrush="LightYellow"

Click="backButton\_Click"

FontFamily="Tahoma"

FontWeight="Bold"

FontSize="16">

</Button>

<ListBox Name="listBox2"

Background="White"

FontSize="18"

Grid.Row="1"

Grid.Column="2"

Margin="20,20,20,50"

MouseDoubleClick="listBox2\_MouseDoubleClick"

>

</ListBox>

<StackPanel Grid.RowSpan="2" Grid.Row="2"

Background="SkyBlue">

<TextBlock FontSize="20"

TextWrapping="Wrap"

TextAlignment="Center"

FontWeight="Bold"

Margin="0,20,0,0">

Информация о диске

</TextBlock>

</StackPanel>

<StackPanel Grid.Row="4" Grid.RowSpan="3"

Background="SlateBlue">

<TextBlock FontSize="20"

TextWrapping="Wrap"

TextAlignment="Center"

FontWeight="Bold"

Margin="0,40,0,0">

Информация о каталоге

</TextBlock>

</StackPanel>

<StackPanel Grid.Row="2" Grid.Column="1"

Background="SkyBlue">

<TextBlock FontSize="20"

TextWrapping="Wrap"

TextAlignment="Center"

FontWeight="Bold"

Text="Объем диска">

</TextBlock>

</StackPanel>

<StackPanel Grid.Row="3" Grid.Column="1"

Background="SkyBlue">

<TextBlock FontSize="20"

TextWrapping="Wrap"

TextAlignment="Center"

FontWeight="Bold"

Text="Свободное пространство">

</TextBlock>

</StackPanel>

<StackPanel Grid.Row="2" Grid.Column="2"

Background="SkyBlue">

<TextBlock FontSize="20"

TextWrapping="Wrap"

TextAlignment="Center"

Name="driveVolume">

</TextBlock>

</StackPanel>

<StackPanel Grid.Row="3" Grid.Column="2"

Background="SkyBlue">

<TextBlock FontSize="20"

TextWrapping="Wrap"

TextAlignment="Center"

Name="driveFreeSpace">

</TextBlock>

</StackPanel>

<StackPanel Grid.Row="4" Grid.Column="1"

Background="SlateBlue">

<TextBlock FontSize="20"

TextWrapping="Wrap"

TextAlignment="Center"

Text="Полное название"

FontWeight="Bold">

</TextBlock>

</StackPanel>

<StackPanel Grid.Row="4" Grid.Column="2"

Background="SlateBlue">

<TextBlock FontSize="20"

TextWrapping="Wrap"

TextAlignment="Center"

Name="fullPathName"

>

</TextBlock>

</StackPanel>

<StackPanel Grid.Row="5" Grid.Column="1"

Background="SlateBlue">

<TextBlock FontSize="20"

TextWrapping="Wrap"

TextAlignment="Center"

Text="Корневой каталог"

FontWeight="Bold">

</TextBlock>

</StackPanel>

<StackPanel Grid.Row="5" Grid.Column="2"

Background="SlateBlue">

<TextBlock FontSize="20"

TextWrapping="Wrap"

TextAlignment="Center"

Name="rootDirectory"

>

</TextBlock>

</StackPanel>

<StackPanel Grid.Row="6" Grid.Column="1"

Background="SlateBlue">

<TextBlock FontSize="20"

TextWrapping="Wrap"

TextAlignment="Center"

Text="Время создания"

FontWeight="Bold">

</TextBlock>

</StackPanel>

<StackPanel Grid.Row="6" Grid.Column="2"

Background="SlateBlue">

<TextBlock FontSize="20"

TextWrapping="Wrap"

TextAlignment="Center"

Name="createTime"

>

</TextBlock>

</StackPanel>

</Grid>

</Window>

**Листинг файла MainWindow.xaml.cs:**

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.IO;

using System.Linq;

using System.Runtime.Remoting.Messaging;

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;

using static System.Net.WebRequestMethods;

using File = System.IO.File;

using Path = System.IO.Path;

namespace Lab1Task

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

Dictionary<string, DateTime> openedFiles = new Dictionary<string, DateTime>();

DriveInfo[] drives = DriveInfo.GetDrives();

string openedDir = "";

public MainWindow()

{

InitializeComponent();

foreach (DriveInfo drive in drives)

{

openedDir = Path.GetDirectoryName(drive.Name);

comboBox1.Items.Add(drive);

comboBox1.FontSize = 18;

comboBox1.SelectedIndex = 0;

}

}

private void comboBox1\_SelectionChanged(object sender, SelectionChangedEventArgs e )

{

listBox1.Items.Clear();

listBox2.Items.Clear();

backButton.IsEnabled = false;

driveFreeSpace.Text = string.Empty;

driveVolume.Text = string.Empty;

DriveInfo drive = new DriveInfo(comboBox1.SelectedItem.ToString());

openedDir = Path.GetDirectoryName(drive.Name);

DirectoryInfo dirInfo = new DirectoryInfo(drive.Name);

fullPathName.Text = dirInfo.FullName;

createTime.Text = dirInfo.CreationTime.ToString();

rootDirectory.Text = dirInfo.Root.ToString();

if (drive.IsReady)

{

driveFreeSpace.Text = Math.Round(drive.TotalFreeSpace / (Math.Pow(2, 30)), 2) + " Гб";

driveVolume.Text = Math.Round(drive.TotalSize / (Math.Pow(2, 30)), 2) + " Гб";

listBox2.IsEnabled = true;

listBox1.IsEnabled = true;

foreach (string file in GetFilesInDirecrory(drive.Name))

{

listBox2.Items.Add(file);

}

foreach (string directory in GetDirecroriesInRootDir(drive.Name))

{

listBox1.Items.Add(directory);

}

}

else

{

listBox1.Items.Add("Диск недоступен");

listBox1.IsEnabled = false;

}

}

private List<string> GetFilesInDirecrory(string path)

{

return Directory.GetFiles(path).ToList();

}

private List<string> GetDirecroriesInRootDir(string path)

{

return Directory.GetDirectories(path).ToList();

}

private void listBox1\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

listBox2.IsEnabled = true;

if (listBox1.SelectedItem != null)

{

string directory = listBox1.SelectedItem.ToString();

DirectoryInfo dirInfo = new DirectoryInfo(listBox1.SelectedItem.ToString());

fullPathName.Text = dirInfo.FullName;

createTime.Text = dirInfo.CreationTime.ToString();

rootDirectory.Text = dirInfo.Root.ToString();

try

{

listBox2.Items.Clear();

openedDir = Path.GetDirectoryName(directory);

if (GetFilesInDirecrory(directory).Count == 0)

{

listBox2.Items.Add("Папка пуста");

listBox2.IsEnabled = false;

}

else

{

foreach (string file in GetFilesInDirecrory(directory))

{

listBox2.Items.Add(file);

}

}

}

catch

{

listBox2.Items.Add("Доступ запрещен");

listBox2.IsEnabled = false;

}

}

}

private void Window\_Closing(object sender, System.ComponentModel.CancelEventArgs e)

{

DateTime windCloseTime=DateTime.Now;

string path= Directory.GetCurrentDirectory()+"/Текстовый файл/";

if (!Directory.Exists(path))

{

Directory.CreateDirectory(path);

}

if (!File.Exists(path + "filesNames.txt"))

{

File.Create(path + "filesNames.txt").Close();

}

Dictionary<string, DateTime> currentOpenedFiles =

openedFiles.Where(f => f.Value >= windCloseTime.

AddSeconds(-10)).ToDictionary(f => f.Key, f => f.Value);

File.WriteAllLines(

path + "filesNames.txt",

currentOpenedFiles.Select(kvp => string.Format($"{kvp.Key,20} \t {kvp.Value,10}")));

}

private void listBox2\_MouseDoubleClick(object sender, MouseButtonEventArgs e)

{

if (listBox2.SelectedItem != null)

{

Process.Start(listBox2.SelectedItem.ToString());

if (!openedFiles.ContainsKey(listBox2.SelectedItem.ToString()))

{

openedFiles.Add(listBox2.SelectedItem.ToString(), DateTime.Now);

}

else

{

openedFiles[listBox2.SelectedItem.ToString()] = DateTime.Now;

}

}

}

private void listBox1\_MouseDoubleClick(object sender, MouseButtonEventArgs e)

{

backButton.IsEnabled = true;

string directory = listBox1.SelectedItem.ToString();

DirectoryInfo dirInfo = new DirectoryInfo(directory);

fullPathName.Text = dirInfo.FullName;

createTime.Text = dirInfo.CreationTime.ToString();

rootDirectory.Text = dirInfo.Root.ToString();

try

{

listBox2.Items.Clear();

openedDir = directory;

if (GetDirecroriesInRootDir(directory).Count != 0)

{

listBox1.Items.Clear();

listBox2.Items.Clear();

foreach (string dir in GetDirecroriesInRootDir(directory))

{

listBox1.Items.Add(dir);

}

}

if (GetFilesInDirecrory(directory).Count == 0)

{

listBox2.Items.Add("Папка пуста");

listBox2.IsEnabled = false;

}

else

{

foreach (string file in GetFilesInDirecrory(directory))

{

listBox2.Items.Add(file);

}

}

}

catch

{

listBox2.Items.Add("Доступ запрещен");

listBox2.IsEnabled = false;

}

}

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

{

listBox1.Items.Clear();

listBox2.Items.Clear();

DirectoryInfo dirInfo = new DirectoryInfo(Directory.GetParent(openedDir).ToString());

fullPathName.Text = dirInfo.FullName;

createTime.Text = dirInfo.CreationTime.ToString();

rootDirectory.Text = dirInfo.Root.ToString();

listBox2.IsEnabled = true;

foreach (DriveInfo drive in drives)

{

if (drive.Name == Directory.GetParent(openedDir).ToString())

{

backButton.IsEnabled = false;

break;

}

}

List<string> dirs = GetDirecroriesInRootDir(Directory.GetParent(openedDir).ToString());

foreach (var dir in dirs)

{

listBox1.Items.Add(dir);

}

List<string> files = GetFilesInDirecrory(Directory.GetParent(openedDir).FullName);

if (files.Count == 0)

{

listBox2.Items.Add("Папка пуста");

listBox2.IsEnabled = false;

}

else

{

foreach (string file in files)

{

listBox2.Items.Add(file);

}

}

openedDir = Directory.GetParent(openedDir).ToString();

}

}

}