

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського  
«Харківський авіаційний інститут»

Факультет радіоелектроніки, комп'ютерних систем та інфокомунікацій

Кафедра комп'ютерних систем, мереж і кібербезпеки

## Лабораторна робота

з Системного програмування  
(назва дисципліни)

на тему: «Работа с файловой системой через классы Windows.Storage»

Виконала: студентка 3-го курсу групи №525ст2  
напряму підготовки (спеціальності)  
123-«Комп'ютерна інженерія»

\_\_\_\_\_  
(шифр і назва напряму підготовки (спеціальності))

Коваленко Я.О.

\_\_\_\_\_  
(прізвище й ініціали студента)

Прийняв: асистент каф.503

Мозговий М.В.

\_\_\_\_\_  
(посада, науковий ступінь, прізвище й ініціали)

Національна шкала: \_\_\_\_\_

Кількість балів: \_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_

Цель работы:

1. Создание Universal Windows Application для Windows 10 на базе UWP.
2. Работа с файловой системой через классы Windows.Storage.

Постановка задачи:

Необходимо создать оконное приложение, которое будет реализовывать функции файлового менеджера:

- Отображать перечень файлов и папок в заданном каталоге.
- Переходить в подкаталог по клику на папку.
- Отображать для списка файлов размер (приведенный к Байтам, Кбайтам, Мбайтам), дату создания, имя файла.
- Предоставлять возможность смены каталога через FolderPicker

Вариант 3	Выполнить поиск файла по части имени в текущей папке и подпапках
-----------	--

Ход работы:

Код программы:

```
<Page
  x:Class="lab7.MainPage"
  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"
  Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">

  <Grid>
    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="Auto"/>
      <ColumnDefinition Width="*/>
    </Grid.ColumnDefinitions>

    <StackPanel Grid.Column="0" Orientation="Vertical">
      <TextBlock Margin="5" Text="Kovalenko Yana 525st2" FontSize="14"/>
      <Button Margin="5" HorizontalAlignment="Stretch" Padding="5" Content="Choose"
Click="ChooseStartFolderButton_Click"/>
      <Button Margin="5" HorizontalAlignment="Stretch" Padding="5" Content="Create"
Click="CreateFolderButton_Click"/>
      <TextBox Margin="5" Padding="5" x:Name="folderName" />
      <Button Margin="5" HorizontalAlignment="Stretch" Padding="5" Content="Delete"
Click="DeleteFolderButton_Click" Background="#33141457"
FocusVisualPrimaryBrush="#FF5966A4" Foreground="#FF020202"/>
    </StackPanel>
```

```

        <ListBox      Margin="5,3,10,6"      Grid.Column="1"      x:Name="listBox"
RenderTransformOrigin="0.5,0.5" UseLayoutRounding="False" d:LayoutRounding="Auto" >
        <ListBox.RenderTransform>
            <CompositeTransform SkewY="-0.393" TranslateY="-1.493"/>
        </ListBox.RenderTransform>
    </ListBox>
</Grid>
</Page>

```

```

using System;
using System.Collections.Generic;
using Windows.Storage;
using Windows.Storage.FileProperties;
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Input;

```

```

namespace lab7
{

```

```

    public sealed partial class MainPage : Page
    {
        private StorageFolder currentFolder;
        public MainPage()
        {
            this.InitializeComponent();
        }

```

```

        private async void ChooseStartFolderButton_Click(object sender, RoutedEventArgs e)
        {
            Windows.Storage.Pickers.FolderPicker folderPicker = new
Windows.Storage.Pickers.FolderPicker();
            folderPicker.ViewMode = Windows.Storage.Pickers.PickerViewMode.Thumbnail;
            folderPicker.FileTypeFilter.Add(".txt");

            StorageFolder folder = await folderPicker.PickSingleFolderAsync();
            ScanDirectory(folder);
        }

```

```

        private async void ScanDirectory(StorageFolder folder)
        {
            if (folder != null)
            {
                listBox.Items.Clear();
                currentFolder = folder;
                StorageFolder parentFolder = await folder.GetParentAsync();
                ListBoxItem listBoxItem = new ListBoxItem();
                listBoxItem.Content = "Back";
                listBoxItem.DoubleTapped += Navigate;
                listBoxItem.Tag = parentFolder;
                listBox.Items.Add(listBoxItem);
            }
        }

```

```

        IReadOnlyList<IStorageItem> itemsList = await folder.GetItemsAsync();
        foreach (var item in itemsList)
        {
            listBoxItem = new ListBoxItem();
            listBoxItem.Tag = item;
            if (item is StorageFolder)
            {
                listBoxItem.Content = "[Folder]" + item.Name + "\tCreated: " +
item.DateCreated.DateTime;
            }
            else
            {
                BasicProperties properties = await item.GetBasicPropertiesAsync();
                listBoxItem.Content = "[Folder]" + item.Name + "\tCreated: " +
item.DateCreated.DateTime + "\tSize: " + GetSize(properties.Size);
            }
            listBox.Items.Add(listBoxItem);
            listBoxItem.DoubleTapped += Navigate;
        }
    }
}

```

```

private string GetSize(ulong byteSize)

```

```

{
    ulong finalSize = byteSize;
    string endString = " B";
    if (finalSize > 1024)
    {
        endString = " kB";
        finalSize /= 1024;
    }

    if (finalSize > 1024)
    {
        endString = " MB";
        finalSize /= 1024;
    }

    if (finalSize > 1024)
    {
        endString = " GB";
        finalSize /= 1024;
    }
    return finalSize+endString;
}

```

```

private void Navigate(object sender, DoubleTappedRoutedEventArgs e)

```

```

{
    ListBoxItem listBoxItem = (ListBoxItem)sender;
    if (listBoxItem.Tag is StorageFolder)
    {
        ScanDirectory((StorageFolder)listBoxItem.Tag);
    }
}

```

```

    }
}

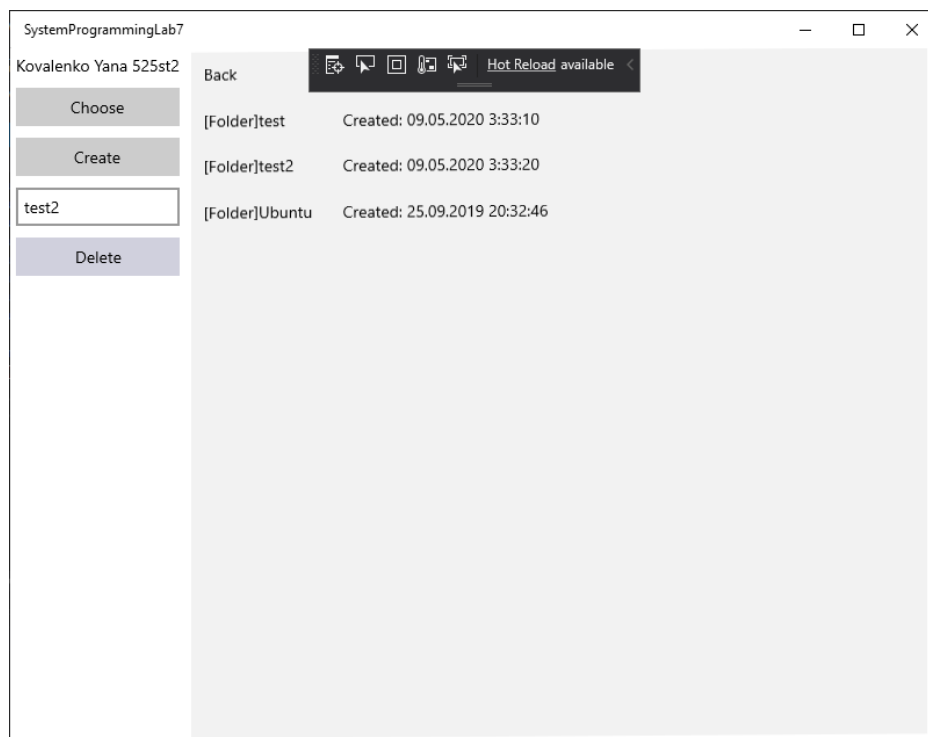
private async void DeleteFolderButton_Click(object sender, RoutedEventArgs e)
{
    if (listBox.SelectedItem == null)
        return;

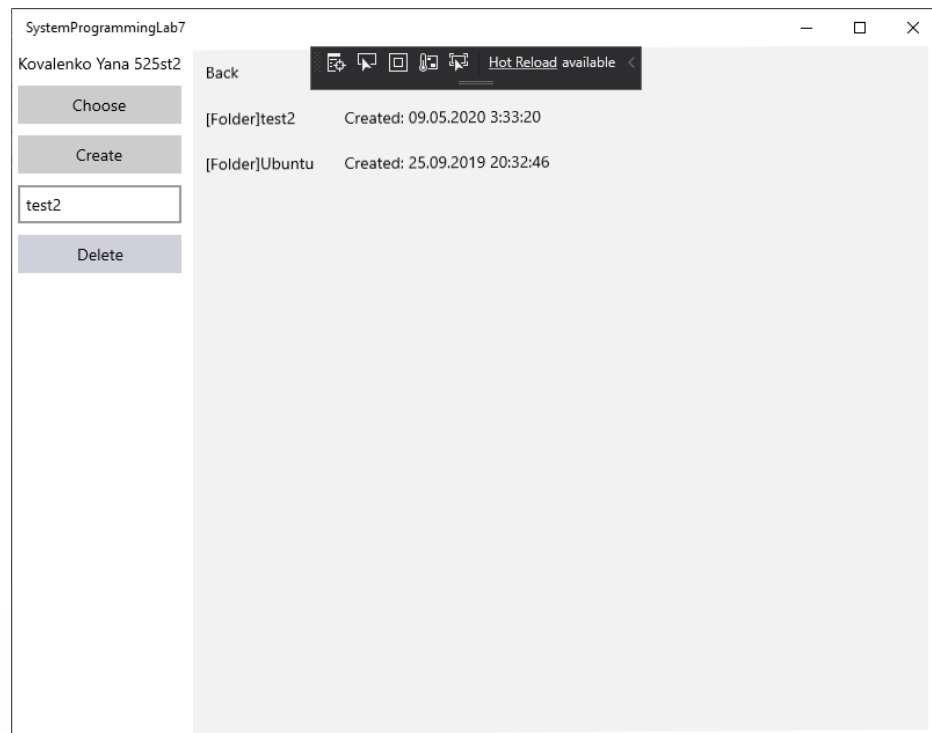
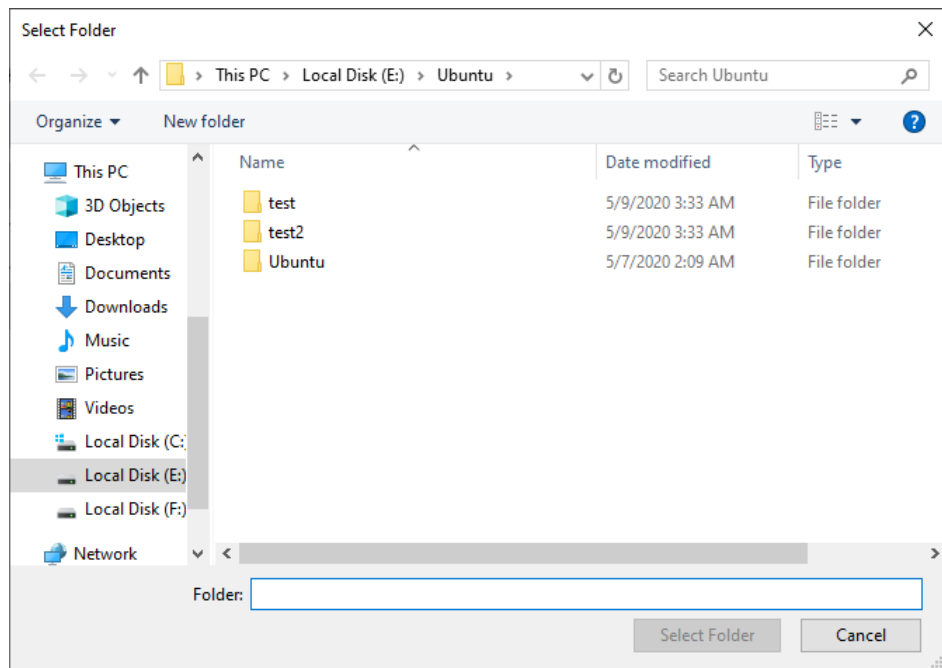
    ListBoxItem listBoxItem = (ListBoxItem)listBox.SelectedItem;
    if (listBoxItem.Tag is StorageFolder)
    {
        StorageFolder folder = (StorageFolder)listBoxItem.Tag;
        await folder.DeleteAsync();
    }
    ScanDirectory(currentFolder);
}

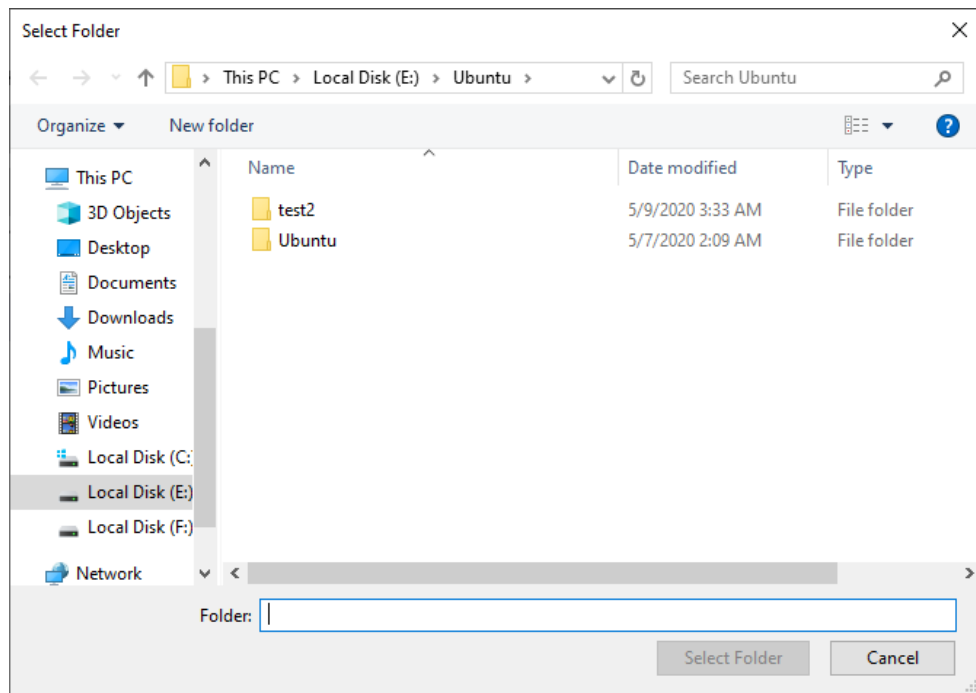
private async void CreateFolderButton_Click(object sender, RoutedEventArgs e)
{
    if (!string.IsNullOrEmpty(folderName.Text) &&
    !string.IsNullOrEmpty(currentFolder.Name))
    {
        await currentFolder.CreateFolderAsync(folderName.Text);
        ScanDirectory(currentFolder);
    }
}
}
}

```

Результат работы:







#### Выводы:

В результате выполнения данной лабораторной работы было создано Universal Windows Application для Windows 10 на базе UWP, а также реализована работа с файловой системой через классы Windows.Storage.