

**федеральное государственное автономное образовательное  
учреждение высшего образования**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
(ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ)  
(Факультет информационных технологий)**

*(Институт Принтмедиа и информационных технологий)  
Кафедра Информатики и информационных технологий*

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**ЛАБОРАТОРНАЯ РАБОТА № 4**

**Дисциплина:** Технология прикладного программирования

**Тема:** Элементы управления.

**Выполнила: студентка группы 221-375**

Орлова Екатерина Артемьевна

**Дата, подпись** 16.05.2023  
(Дата)

\_\_\_\_\_  
(Подпись)

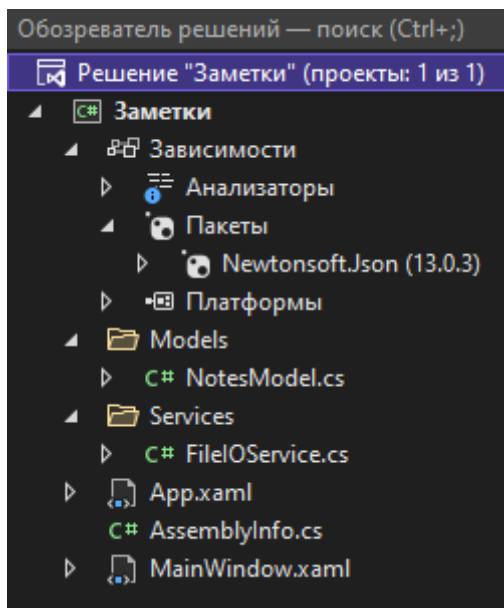
**Замечания:** \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**Москва 2022**

## **Задание**

Разработать программу "Заметки".

## Листинг кода



### *MainWindow.xaml*

```
<Window x:Class="Заметки.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:Заметки"
        mc:Ignorable="d"
        Title="Заметки" Height="450" Width="800" Background="#ffdbde" Loaded="Window_Loaded">
    <Grid>
        <DataGrid x:Name="notes" FontSize="16" Margin="10" BorderBrush="#fa8f89"
        Background="White" AutoGenerateColumns="False">
            <DataGrid.Columns>
                <DataGridTextColumn Header="Дата и время" Width="165" IsReadOnly="True"
                Binding="{Binding Path=CreateDate}"></DataGridTextColumn>
                <DataGridCheckBoxColumn Header="Актуальность" Width="110" IsReadOnly="False"
                Binding="{Binding Path=CheckNot,
                UpdateSourceTrigger=PropertyChanged}"></DataGridCheckBoxColumn>
                <DataGridTextColumn Header="Мои заметки" Width="*" IsReadOnly="False"
                Binding="{Binding Path=ListNot}"></DataGridTextColumn>
            </DataGrid.Columns>
        </DataGrid>
    </Grid>
</Window>
```

### *MainWindow.xaml.cs*

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Linq;
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 Заметки.Models;
using Заметки.Services;

namespace Заметки
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        private readonly string PATH = $"{Environment.CurrentDirectory}\\notData.json";
        private BindingList<NotesModel> _notData;
        private FileIOService _fileIOService;

        public MainWindow()
        {
            InitializeComponent();
        }

        private void Window_Loaded(object sender, RoutedEventArgs e)
        {
            _fileIOService = new FileIOService(PATH);

            try
            {
                _notData = _fileIOService.LoadData();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
                Close();
            }

            notes.ItemsSource = _notData;

            _notData.ListChanged += _notData_ListChanged;
        }

        private void _notData_ListChanged(object? sender, ListChangedEventArgs e)
        {
            if(e.ListChangedType == ListChangedType.ItemAdded
                || e.ListChangedType == ListChangedType.ItemDeleted
                || e.ListChangedType == ListChangedType.ItemChanged)
            {
                try
                {
                    _fileIOService.SaveData(sender);
                }
                catch (Exception ex)
                {
                    MessageBox.Show(ex.Message);
                    Close();
                }
            }
        }
    }
}

```

### *NotesModel.cs*

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Linq;

```

```

using System.Text;
using System.Threading.Tasks;

namespace Заметки.Models
{
    class NotesModel : INotifyPropertyChanged
    {
        public DateTime CreatDate { get; set; } = DateTime.Now; // добавление текущей даты и
времени

        private bool _CheckNot;
        public bool CheckNot
        {
            get { return _CheckNot; }
            set
            {
                if (_CheckNot == value)
                {
                    return;
                }
                _CheckNot = value;
                OnPropertyChanged("CheckNot");
            }
        }

        private string _ListNot;

        public string ListNot
        {
            get { return _ListNot; }
            set
            {
                if (_ListNot == value)
                {
                    return;
                }
                _ListNot = value;
                OnPropertyChanged("ListNot");
            }
        }

        public event PropertyChangedEventHandler? PropertyChanged;

        protected virtual void OnPropertyChanged(string propertyName = "")
        {
            if (PropertyChanged != null)
            {
                PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
            }
        }
    }
}

```

### *FileIOService.cs*

```

using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.IO;
using System.Linq;
using System.Text;

```

```

using System.Threading.Tasks;
using Заметки.Models;

namespace Заметки.Services
{
    class FileIOService
    {
        private readonly string PATH;

        public FileIOService (string path)
        {
            PATH = path;
        }

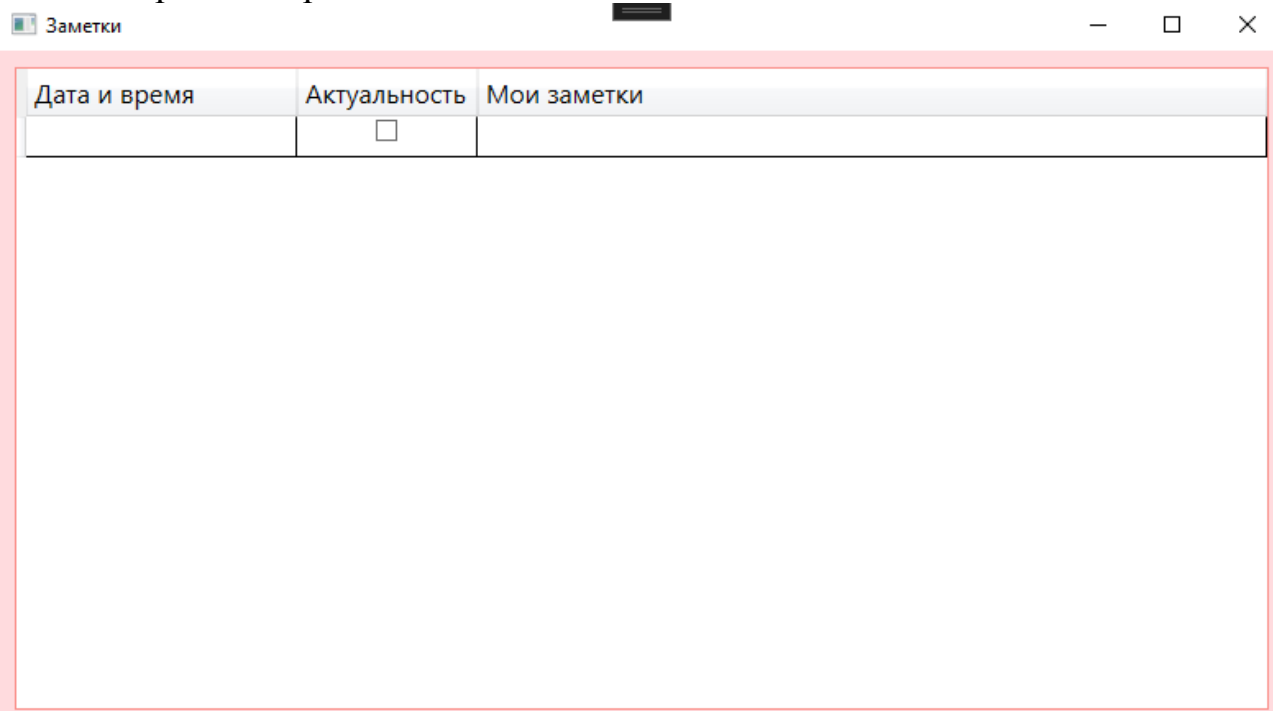
        public BindingList<NotesModel> LoadData()
        {
            var fileExists = File.Exists(PATH); // проверка на существование файла
            if (!fileExists)
            {
                File.CreateText(PATH).Dispose(); // создадим, если нет
                return new BindingList<NotesModel>();
            }
            using (var reader = File.OpenText(PATH))
            {
                var fileText = reader.ReadToEnd();
                return JsonConvert.DeserializeObject<BindingList<NotesModel>>(fileText);
            }
        }

        public void SaveData(object notData)
        {
            using (StreamWriter writer = File.CreateText(PATH))
            {
                string output = JsonConvert.SerializeObject(notData);
                writer.Write(output);
            }
        }
    }
}

```

## Демонстрация работы

### 1. Открываем приложение для заметок.



2. Попробуем добавить запись (записываем текст в редактируемое поле «мои заметки», нажимаем enter), при добавлении автоматически подставляется текущая дата и время.

Заметки

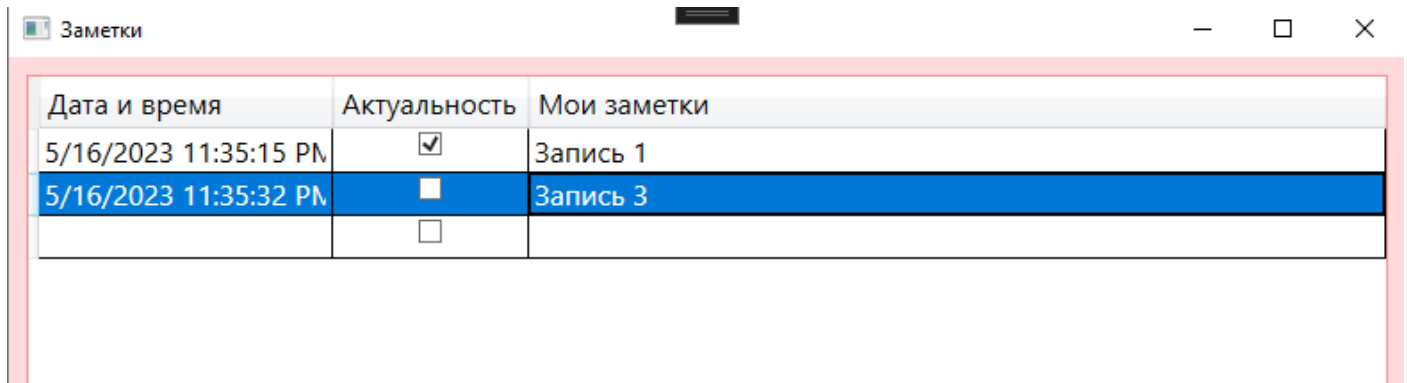
Дата и время	Актуальность	Мои заметки
5/16/2023 11:35:15 PM	<input type="checkbox"/>	Запись 1
5/16/2023 11:35:25 PM	<input type="checkbox"/>	Запись 2
5/16/2023 11:35:32 PM	<input type="checkbox"/>	Запись 3
	<input type="checkbox"/>	

3. Актуальность записей можно отметить во второй колонке.

Заметки

Дата и время	Актуальность	Мои заметки
5/16/2023 11:35:15 PM	<input checked="" type="checkbox"/>	Запись 1
5/16/2023 11:35:25 PM	<input checked="" type="checkbox"/>	Запись 2
5/16/2023 11:35:32 PM	<input type="checkbox"/>	Запись 3
	<input type="checkbox"/>	

4. Если нужно удалить лишние записи, можно нажать на них, а затем клавишу «delete».



5. Записи сохраняются, так что при повторном открытии программы не теряются.

