

Министерство образования Тульской области

**Государственное профессиональное образовательное учреждение
Тульской области «Донской политехнический колледж»**

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

для студентов, обучающихся по специальности
09.02.01 Компьютерные системы и комплексы
по выполнению курсовой работы по дисциплине
ОП.09 Основы алгоритмизации и программирования

Автор:

С.М. Гвоздев, преподаватель
ГПОУ ТО «ДПК»

2023 г.

Лист согласования:

Автор разработки:

Гвоздев Сергей Михайлович, преподаватель ГПОУ ТО «ДПК»

Рецензенты:

- Панченко Т.А., заместитель директора по учебно-методической работе ГПОУ ТО «ДПК»
- Филатова Е.А., старший методист ГПОУ ТО «ДПК»

Методические рекомендации для студентов 2 курса специальности 09.02.01 "Компьютерные системы и комплексы" содержат краткие теоретические сведения, основные понятия и определения, этапы проектирования, создания приложения с использованием технологии WPF, на платформе .NET 7 с использованием библиотеки «Material Design In XAML Toolkit».

СОГЛАСОВАНО

на заседании предметной
(цикловой) комиссии дисциплин
профессионального цикла отделения
«Информационная безопасность и
администрирование»

Протокол № 1
от «03» февраля 2023 г.

Председатель ПЦК Гвоздев С.М.

СОДЕРЖАНИЕ

Введение.....	4
1. Создание проекта	6
1.1. Разметка MainWindow	6
2. Реализация «MainPage» и «TaskPage»	13
3. Реализация директории «PageTask».....	20

Введение

Методические рекомендации составлены в соответствии с рабочей программой ОП.09 «Основы алгоритмизации и программирования», разработанные в рамках основных образовательных программ специальности 09.02.01 " Компьютерные системы и комплексы ".

Целью освоения дисциплины «Основы алгоритмизации и программирования» формирование у обучающихся знаний современных информационных, компьютерных и сетевых технологий, умений разработки алгоритмов и программ на языке высокого уровня С# хранения, обработки и анализа информации из различных источников и баз данных.

Задача дисциплины – обучение студентов анализу и алгоритмизации решаемых задач, оформлению решения задачи в графическом виде (в виде схем алгоритмов), программированию любого алгоритма, задачи, метода, проектированию и отладке достаточно сложных программ.

В результате освоения дисциплины обучающийся должен демонстрировать следующие результаты образования:

Знать

- общие принципы построения и использования языков программирования, их классификацию;
- современные интегрированные среды разработки программ;
- процесс создания программ;
- стандарты языков программирования;
- среду программирования на языке высокого уровня С#;
- общую характеристику языка С#: назначение, принципы построения и использования.

Уметь

- формализовать поставленную задачу и создавать информационную модель для
- решаемой задачи;

- разрабатывать и тестировать алгоритм модели информационной деятельности человека;
- применять полученные знания к различным предметным областям;
- составлять и оформлять программы на языках программирования;
- тестировать и отлаживать программы.

Материал разбит на 3 раздела, каждый из которых содержит краткие теоретические сведения и практический пример разработки модуля приложения, а также разметку и исходный код для самостоятельного изучения.

1. Создание проекта

1.1. Разметка MainWindow

Создадим новый проект – выбираем «Приложение WPF (Майкрософт)» и нажимаем «Далее».

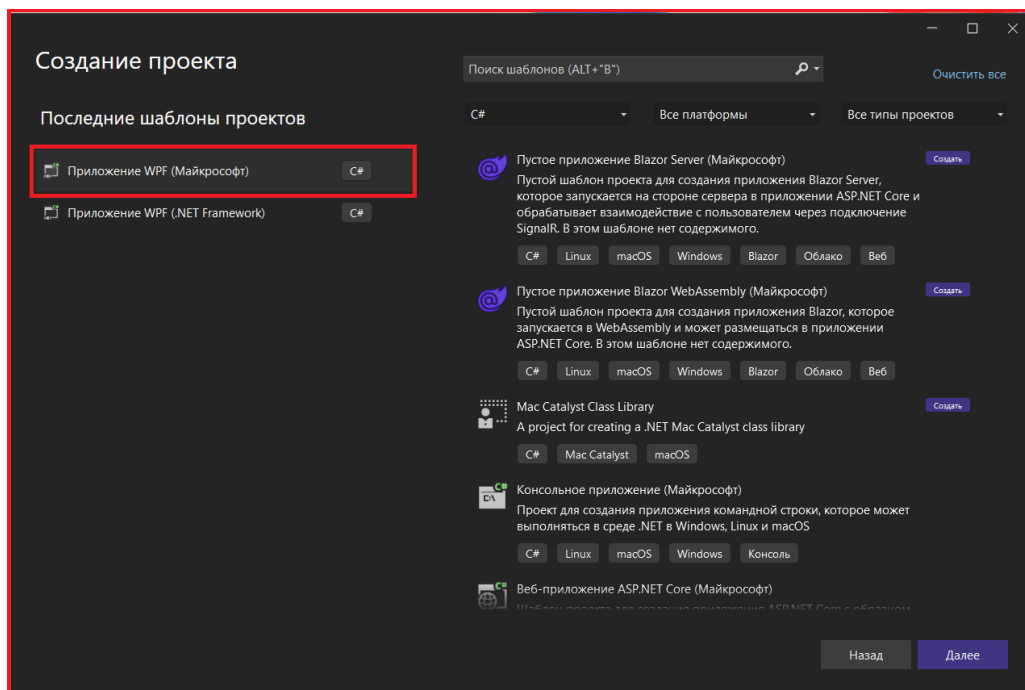


Рисунок 1.1 – Создание проекта

В окне «Настроить новый проект» заполняем поля:

- Имя проекта;
- Расположение;

После заполнения всех полей нажимаем на кнопку «Далее»

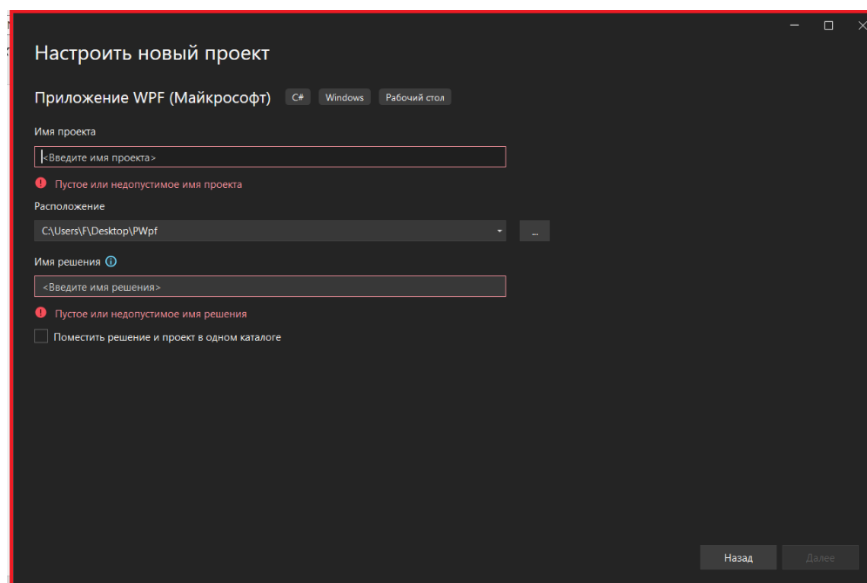


Рисунок 1.2 – Настройка нового проекта

Выбираем платформу «.NET 7» и нажимаем создать.

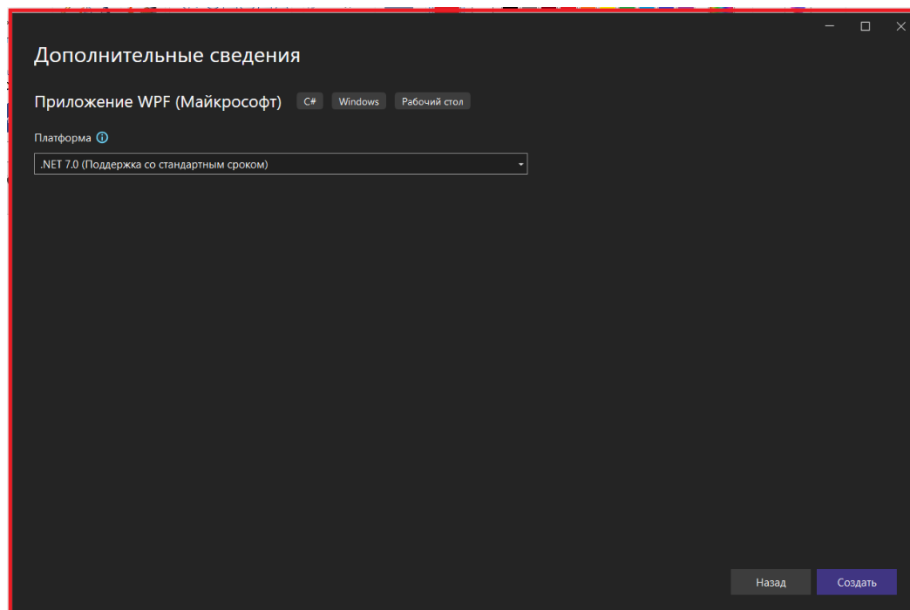


Рисунок 1.3 – Выбор платформы проекта

Создадим общую структуру проекта. Она будет состоять из следующих директорий:

- Core – директория которая содержит вспомогательные классы;
- Images – директория которая содержит графические ресурсы приложения;
- AppStyle – директория которая содержит стили приложения;
- View – директория которая содержит: окна, страницы и пользовательские элементы используемые в приложении.

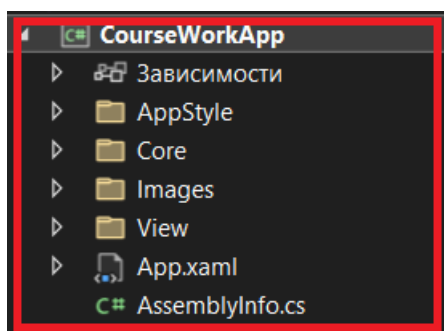


Рисунок 1.4 – Структура проекта

Теперь необходимо добавить графическую библиотеку «Material Design». Для этого ПКМ нажимаем по проекту в появившемся окне выбираем «Управление пакетами NuGet...», выбираем вкладку «Обзор», в

поиске вводим «Material Design» и устанавливаем два компонента «MaterialDesignThemes» и «MaterialDesignColors».

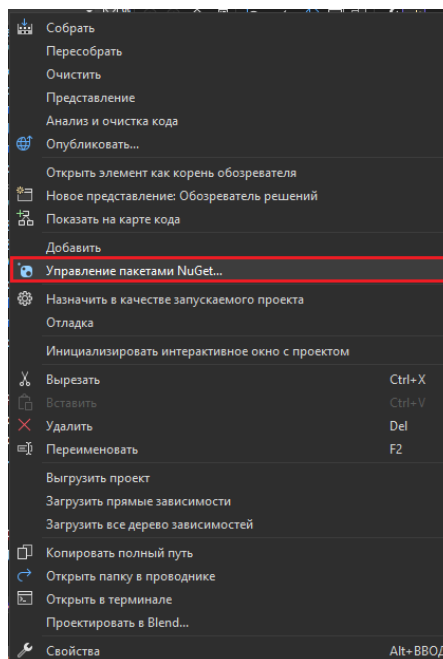


Рисунок 1.5 – Контекстное меню «Управление пакетами NuGet...»

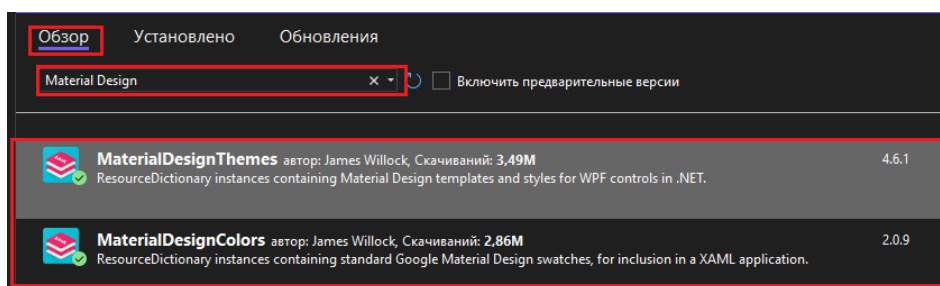


Рисунок 1.6 – Установка «MaterialDesignThemes» и «MaterialDesignColors».

Теперь перенесем «MainWindow» в директорию «View» и изменим файл «App.xaml». Изменим путь к «MainWindow» в «StartupUri» и добавим справочник ресурсов, а также добавим новое пространство имен «materialDesign» и подключить файл с общими стилями.

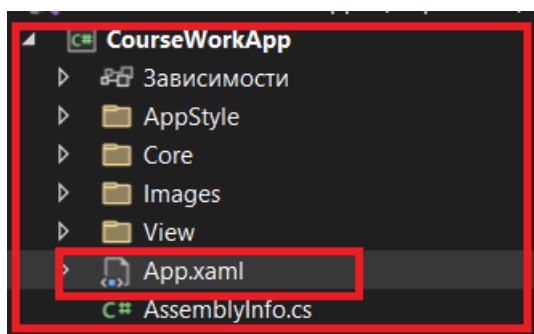


Рисунок 1.7 – Выбор файла «App.xaml»


```
Application x:Class="CourseWorkApp.App"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
StartupUri="View/MainWindow.xaml"
<Application.Resources>
  <ResourceDictionary>
    <ResourceDictionary.MergedDictionaries>
      <materialDesign:BundledTheme BaseTheme="Light" PrimaryColor="{x:Null}" SecondaryColor="Lime" />
      <ResourceDictionary Source="pack://application:,,,/MaterialDesignThemes.Wpf;component/Themes/MaterialDesignTheme.Defaults.xaml" />
      <ResourceDictionary Source="AppStyle/MainStyle.xaml"/>
    </ResourceDictionary.MergedDictionaries>
  </ResourceDictionary>
</Application.Resources>
</Application>
```

Рисунок 1.8 – Редактирование файла «App.xaml»

Далее перейдем к описанию базовых стилей приложения. Для этого переходим в директорию «AppStyle» и редактируем словарь ресурсов «MainStyle».

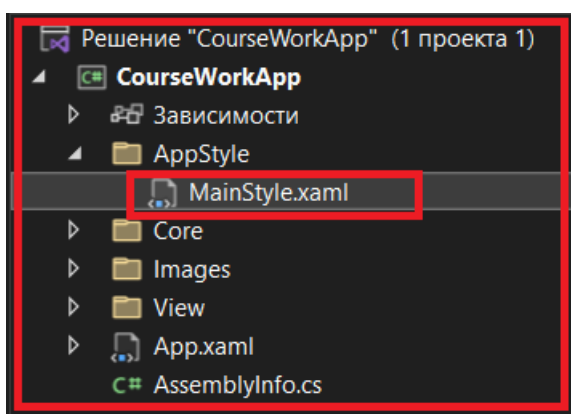


Рисунок 1.9 – Редактирование словаря ресурсов «MainStyle.xaml»

```
<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes">

  <Style TargetType="{x:Type TextBlock}" x:Key="menuButtonText">
    <Setter Property="Margin" Value="0 7 0 0" />
  </Style>

  <Style TargetType="Button" x:Key="menuButton">
    <Setter Property="Margin" Value="0 7 0 0" />
    <Setter Property="FontSize" Value="14" />
    <Setter Property="Width" Value="120" />
    <Setter Property="Height" Value="85" />
    <Setter Property="Background" Value="Transparent" />
    <Setter Property="Foreground" Value="#A9A9A9" />
    <Setter Property="FocusVisualStyle" Value="{x:Null}" />
    <Setter Property="Cursor" Value="Hand" />
    <Setter Property="Template">
      <Setter.Value>
        <ControlTemplate TargetType="Button">
          <Border Background="{TemplateBinding Background}"
            CornerRadius="15"
            Padding="15">
            <ContentPresenter HorizontalAlignment="Center"
              VerticalAlignment="Center"/>
          </Border>
        </ControlTemplate>
      </Setter.Value>
    </Setter>
  </Style>
```

Рисунок 1.10 – Редактирование словаря ресурсов «MainStyle.xaml»

```
<Style.Triggers>
  <Trigger Property="IsMouseOver" Value="True">
    <Setter Property="Background" Value="#0B7373" />
    <Setter Property="Foreground" Value="White" />
  </Trigger>

  <Trigger Property="IsMouseCaptured" Value="True">
    <Setter Property="Background" Value="#0B7373" />
    <Setter Property="Foreground" Value="White" />
  </Trigger>
</Style.Triggers>
</Style>

<Style TargetType="materialDesign:PackIcon" x:Key="menuButtonIcon">
  <Setter Property="Width" Value="20" />
  <Setter Property="Height" Value="20" />
  <Setter Property="HorizontalAlignment" Value="Center" />
  <Setter Property="Foreground" Value="{Binding Path=Foreground,
    RelativeSource={RelativeSource FindAncestor,
    AncestorType={x:Type Button}}}" />
</Style>
```

Рисунок 1.11 – Редактирование словаря ресурсов «MainStyle.xaml»

Затем переходим в директорию «Core» и создадим класс «ConnectionCore», который мы будем использовать для навигации.

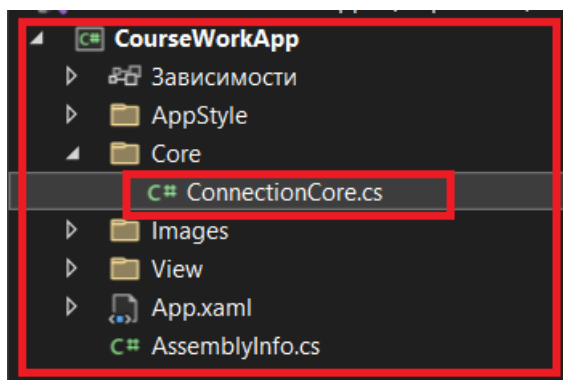


Рисунок 1.12 – Класс «ConnectionCore»

```
namespace CourseWorkApp.Core
{
  Ссылка: 33
  public static class ConnectionCore
  {
    Ссылка: 33
    public static Frame? CoreFrame { get; set; }
  }
}
```

Рисунок 1.13 – Реализация класса «ConnectionCore»

Далее переходим в директорию «View» и создадим две новые директории «Pages» и «PageTask», далее создадим пять страниц под задания,

главную страницу и страницу содержащую информацию о всех пяти заданиях.

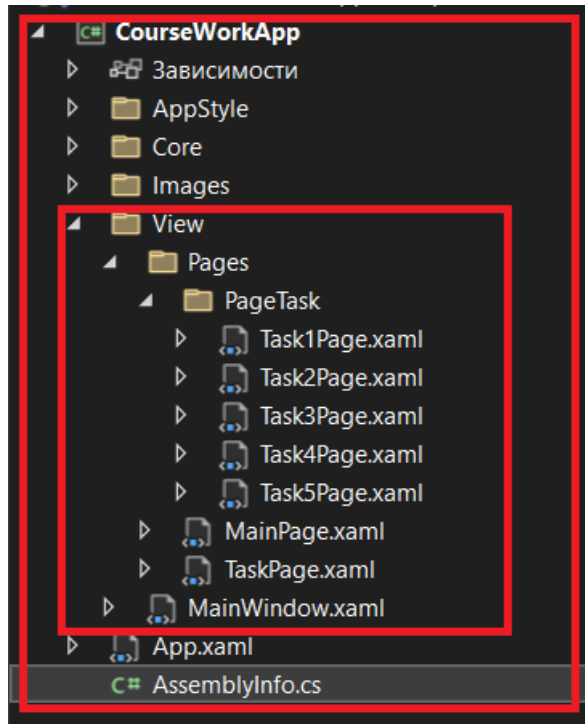


Рисунок 1.14 – Директория «View»

Теперь перейдём к разметке «MainWindow.xaml» и реализации логики «MainWindow.xaml.cs».

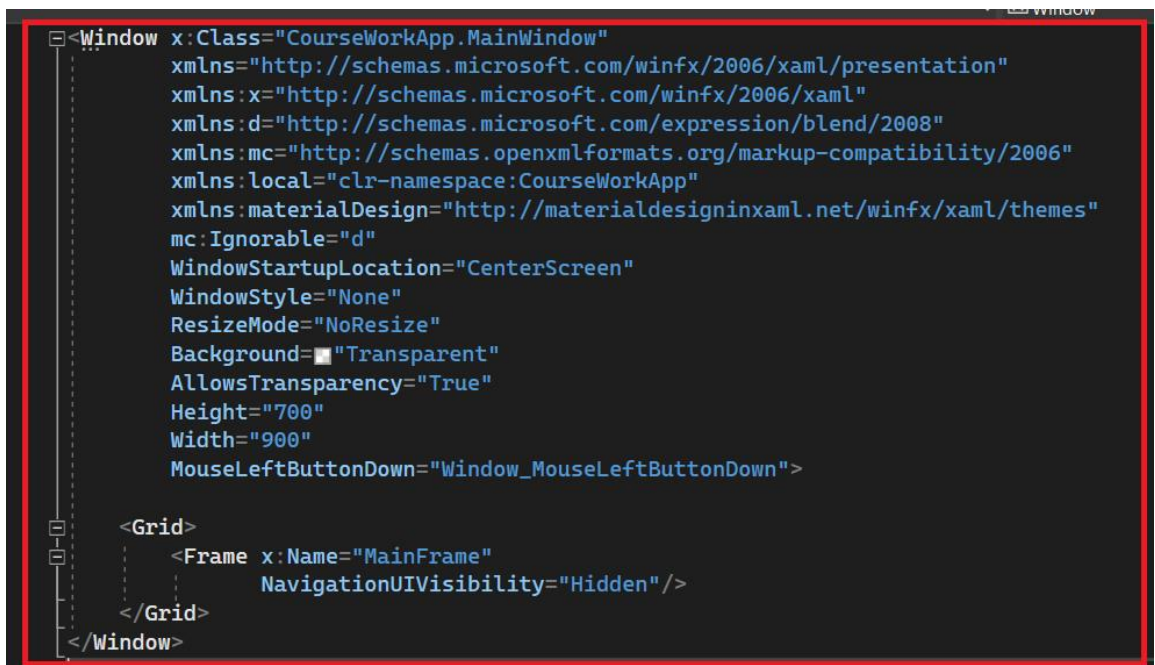


Рисунок 1.15 – Разметка «MainWindow.xaml»

```
Ссылка: 3
public partial class MainWindow : Window
{
    Ссылка: 0
    public MainWindow()
    {
        InitializeComponent();
        ConnectionCore.CoreFrame = MainFrame;
        MainFrame.Navigate(new MainPage());
    }

    Ссылка: 1
    private void Window_MouseLeftButtonDown(object sender, System.Windows.Input.MouseButtonEventArgs e)
    {
        if (e.ClickCount == 2)
        {
            Application.Current.Shutdown();
        }
    }
}
```

Рисунок 1.16 – Реализации логики «MainWindow.xaml.cs»

На этом создание проекта и разметка MainWindow закончена.

2. Реализация «MainPage» и «TaskPage»

Перейдем к разметке «MainPage.xaml».

```
<Page x:Class="CourseWorkApp.View.Pages.MainPage"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:local="clr-namespace:CourseWorkApp.View.Pages"
xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
mc:Ignorable="d"
d:DesignHeight="700"
d:DesignWidth="900"
FontFamily="Consolas">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="1*" />
            <RowDefinition Height="1*" />
            <RowDefinition Height="1*" />
        </Grid.RowDefinitions>
        <Image Grid.RowSpan="3"
Source="/Images/main_bg.jpg"
Stretch="UniformToFill"/>
        <Border Grid.Row="0"
HorizontalAlignment="Center"
VerticalAlignment="Center"
Background="#0B7373"
Width="500"
Height="100"
CornerRadius="15"
Effect="{StaticResource MaterialDesignElevationShadow1}">
            <StackPanel HorizontalAlignment="Center"
VerticalAlignment="Center">
                <TextBlock Text="Курсовая работа"
FontWeight="SemiBold"
Foreground="White"
FontSize="40"/>
            </StackPanel>
        </Border>
    </Grid>

```

Рисунок 2.1 – Разметка «MainPage.xaml»

```

                </StackPanel>
            </Border>
        <Border Grid.Row="1"
HorizontalAlignment="Center"
VerticalAlignment="Center"
Background="#0B7373"
Width="800"
Height="150"
CornerRadius="15"
Effect="{StaticResource MaterialDesignElevationShadow1}">
            <StackPanel HorizontalAlignment="Center"
VerticalAlignment="Center">
                <TextBlock Text="по дисциплине: "
FontWeight="SemiBold"
Foreground="White"
FontSize="35"/>
                <TextBlock Text="Основы алгоритмизации и программирования"
Margin="0 5 0 0"
FontWeight="SemiBold"
Foreground="White"
FontSize="35"/>
            </StackPanel>
        </Border>
    </Page>

```

Рисунок 2.2 – Разметка «MainPage.xaml»

```

<StackPanel Grid.Row="2"
    HorizontalAlignment="Center"
    VerticalAlignment="Center">
    <Button Background="#0B7373"
        Foreground="White"
        x:Name="BtnTask"
        Click="BtnTask_Click"
        Style="{StaticResource MaterialDesignRaisedButton}"
        materialDesign:ButtonProgressAssist.Value="-1"
        materialDesign:ButtonProgressAssist.IndicatorForeground="White"
        materialDesign:ButtonProgressAssist.IsIndicatorVisible="True"
        materialDesign:ButtonProgressAssist.IsIndeterminate="True"
        Height="60">
        <StackPanel Orientation="Horizontal">
            <materialDesign:PackIcon Kind="Eye"
                Margin="0 0 3 0"
                Width="50"
                Height="40"/>
            <TextBlock Text="Перейти к заданиям"
                FontSize="25"
                VerticalAlignment="Center">
                <TextBlock.Style>
                    <Style TargetType="TextBlock">
                        <Style.Triggers>
                            <Trigger Property="IsMouseOver" Value="True">
                                <Setter Property="TextDecorations" Value="Underline" />
                            </Trigger>
                        </Style.Triggers>
                    </Style>
                </TextBlock.Style>
            </TextBlock>
        </StackPanel>
    </Button>
</StackPanel>
</Grid>

```

Рисунок 2.3 – Разметка «MainPage.xaml»

После разметки перейдем к реализации метода «BtnTask_Click».

```

Ссылка: 9
public partial class MainPage : Page
{
    Ссылка: 7
    public MainPage()
    {
        InitializeComponent();
    }

    Ссылка: 1
    private void BtnTask_Click(object sender, RoutedEventArgs e)
    {
        ConnectionCore.CoreFrame?.Navigate(new TaskPage());
    }
}

```

Рисунок 2.4 – Реализация метода «BtnTask_Click»

Далее переходим к разметке «TaskPage». Эта страница содержит задания и ссылки на их решение.

```

<Page x:Class="CourseWorkApp.View.Pages.TaskPage"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:local="clr-namespace:CourseWorkApp.View.Pages"
xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
mc:Ignorable="d"
d:DesignHeight="700"
d:DesignWidth="900"
FontFamily="Consolas">
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="170" />
            <ColumnDefinition Width="*" />
        </Grid.ColumnDefinitions>
        <Image Grid.ColumnSpan="2"
Source="/Images/main_bg.jpg"
Stretch="UniformToFill"/>
        <Border Background="#FDFDFD"
Grid.Column="0"
BorderThickness="1"
BorderBrush="#DBDBDB"
Width="140"
CornerRadius="15"
Height="600">
            <StackPanel>
                <Button Style="{DynamicResource menuButton}"
Margin="0 10 0 0"
x:Name="BtnTask1"
Click="BtnTask1_Click">

```

Рисунок 2.5 – Разметка «TaskPage.xaml»

```

                <StackPanel>
                    <materialDesign:PackIcon Kind="TaskAdd"
Style="{DynamicResource menuButtonIcon}"/>
                    <TextBlock Text="Задание %1"
Style="{DynamicResource menuButtonText}"/>
                </StackPanel>
            </Button>
            <Button Style="{DynamicResource menuButton}"
x:Name="BtnTask2"
Click="BtnTask2_Click">
                <StackPanel>
                    <materialDesign:PackIcon Kind="TaskAdd"
Style="{DynamicResource menuButtonIcon}"/>
                    <TextBlock Text="Задание %2"
Style="{DynamicResource menuButtonText}"/>
                </StackPanel>
            </Button>
            <Button Style="{DynamicResource menuButton}"
x:Name="BtnTask3"
Click="BtnTask3_Click">
                <StackPanel>
                    <materialDesign:PackIcon Kind="TaskAdd"
Style="{DynamicResource menuButtonIcon}"/>
                    <TextBlock Text="Задание %3"
Style="{DynamicResource menuButtonText}"/>
                </StackPanel>
            </Button>

```

Рисунок 2.6 – Разметка «TaskPage.xaml»

```

<Button Style="{DynamicResource menuButton}"
        x:Name="BtnTask4"
        Click="BtnTask4_Click">
    <StackPanel>
        <materialDesign:PackIcon Kind="TaskAdd"
                                Style="{DynamicResource menuButtonIcon}"/>
        <TextBlock Text="Задание %4"
                  Style="{DynamicResource menuButtonText}"/>
    </StackPanel>
</Button>

<Button Style="{DynamicResource menuButton}"
        x:Name="BtnTask5"
        Click="BtnTask5_Click">
    <StackPanel>
        <materialDesign:PackIcon Kind="TaskAdd"
                                Style="{DynamicResource menuButtonIcon}"/>
        <TextBlock Text="Задание %5"
                  Style="{DynamicResource menuButtonText}"/>
    </StackPanel>
</Button>

<Button Style="{DynamicResource menuButton}"
        x:Name="BtnLogout"
        Click="BtnLogout_Click">
    <StackPanel>
        <materialDesign:PackIcon Kind="Power"
                                Style="{DynamicResource menuButtonIcon}"/>
        <TextBlock Text="Выйти"
                  Style="{DynamicResource menuButtonText}"/>
    </StackPanel>
</Button>
</StackPanel>
</Border>

```

Рисунок 2.7 – Разметка «TaskPage.xaml»

```

<Grid Grid.Column="1">
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto" />
        <RowDefinition Height="Auto" />
        <RowDefinition Height="Auto" />
        <RowDefinition Height="Auto" />
        <RowDefinition Height="Auto" />
    </Grid.RowDefinitions>

    <Border Grid.Row="0"
            HorizontalAlignment="Center"
            VerticalAlignment="Center"
            Margin="0 15 0 0"
            Width="650"
            Height="150"
            Background="■" "#0B7373"
            CornerRadius="15"
            Padding="5"
            Effect="{StaticResource MaterialDesignElevationShadow1}">
        <StackPanel HorizontalAlignment="Left"
                    VerticalAlignment="Center">
            <TextBlock Text="Задание %1 – Создать класс с именем Address."
                      FontSize="17"
                      TextWrapping="Wrap"
                      Foreground="■" "White"
                      FontWeight="SemiBold" />
        </StackPanel>
    </Border>

```

Рисунок 2.8 – Разметка «TaskPage.xaml»


```

<Border Grid.Row="1"
        HorizontalAlignment="Center"
        VerticalAlignment="Top"
        Margin="0 10 0 0"
        Width="650"
        Height="140"
        Background="#0B7373"
        CornerRadius="15"
        Padding="5"
        Effect="{StaticResource MaterialDesignElevationShadow1}">
    <StackPanel HorizontalAlignment="Left"
                VerticalAlignment="Center">
        <TextBlock Text="Задание %2 - Ввести два целочисленных массива"
                  FontSize="20"
                  TextWrapping="Wrap"
                  Foreground="White"
                  FontWeight="SemiBold" />
    </StackPanel>
</Border>

<Border Grid.Row="2"
        HorizontalAlignment="Center"
        VerticalAlignment="Top"
        Margin="0 10 0 0"
        Width="650"
        Height="120"
        Background="#0B7373"
        CornerRadius="15"
        Padding="5"
        Effect="{StaticResource MaterialDesignElevationShadow1}">
    <StackPanel HorizontalAlignment="Left"
                VerticalAlignment="Center">
        <TextBlock Text="Задание %3 - Ввести массив, состоящий из 8 элементов"
                  FontSize="20"
                  TextWrapping="Wrap"
                  Foreground="White"

```

Рисунок 2.9 – Разметка «TaskPage.xaml»

```

        FontWeight="SemiBold" />
    </StackPanel>
</Border>

<Border Grid.Row="3"
        HorizontalAlignment="Center"
        VerticalAlignment="Top"
        Margin="0 10 0 0"
        Width="650"
        Height="100"
        Background="#0B7373"
        CornerRadius="15"
        Padding="5"
        Effect="{StaticResource MaterialDesignElevationShadow1}">
    <StackPanel HorizontalAlignment="Left"
                VerticalAlignment="Center">
        <TextBlock Text="Задание %4 - Отсортировать по убыванию элементы массива"
                  FontSize="20"
                  TextWrapping="Wrap"
                  Foreground="White"
                  FontWeight="SemiBold" />
    </StackPanel>
</Border>

<Border Grid.Row="4"
        HorizontalAlignment="Center"
        VerticalAlignment="Top"
        Margin="0 10 0 0"
        Width="650"
        Height="70"
        Background="#0B7373"
        CornerRadius="15"
        Padding="5"
        Effect="{StaticResource MaterialDesignElevationShadow1}">
    <StackPanel HorizontalAlignment="Left"
                VerticalAlignment="Center">

```

Рисунок 2.10 – Разметка «TaskPage.xaml»

```

VerticalAlignment="Center"
<TextBlock Text="Задание №5 – Дана целочислен
FontSize="20"
TextWrapping="Wrap"
Foreground="White"
FontWeight="SemiBold"/>
</StackPanel>
</Border>

```

Рисунок 2.11 – Разметка «TaskPage.xaml»

Затем переходим в файл «TaskPage.xaml.cs» для реализации методов навигации.

```

public TaskPage()
{
    InitializeComponent();
}

Ссылка: 1
private void BtnTask1_Click(object sender, RoutedEventArgs e)
{
    ConnectionCore.CoreFrame?.Navigate(new Task1Page());
}

Ссылка: 1
private void BtnTask2_Click(object sender, RoutedEventArgs e)
{
    ConnectionCore.CoreFrame?.Navigate(new Task2Page());
}

Ссылка: 1
private void BtnTask3_Click(object sender, RoutedEventArgs e)
{
    ConnectionCore.CoreFrame?.Navigate(new Task3Page());
}

Ссылка: 1
private void BtnTask4_Click(object sender, RoutedEventArgs e)
{
    ConnectionCore.CoreFrame?.Navigate(new Task4Page());
}

```

Рисунок 2.12 – Реализация «TaskPage.xaml.cs»

```

private void BtnTask5_Click(object sender, RoutedEventArgs e)
{
    ConnectionCore.CoreFrame?.Navigate(new Task5Page());
}
}
Ссылка: 1
private void BtnLogout_Click(object sender, RoutedEventArgs e)
{
    if (MessageBox.Show("Вы действительно хотите выйти?",
        "Системное сообщение",
        MessageBoxButton.YesNo,
        MessageBoxImage.Question) == MessageBoxResult.Yes)
    {
        ConnectionCore.CoreFrame?.Navigate(new MainPage());
    }
}
}

```

Рисунок 2.13 – Реализация «TaskPage.xaml.cs»

Запустим приложение и проверим навигацию.

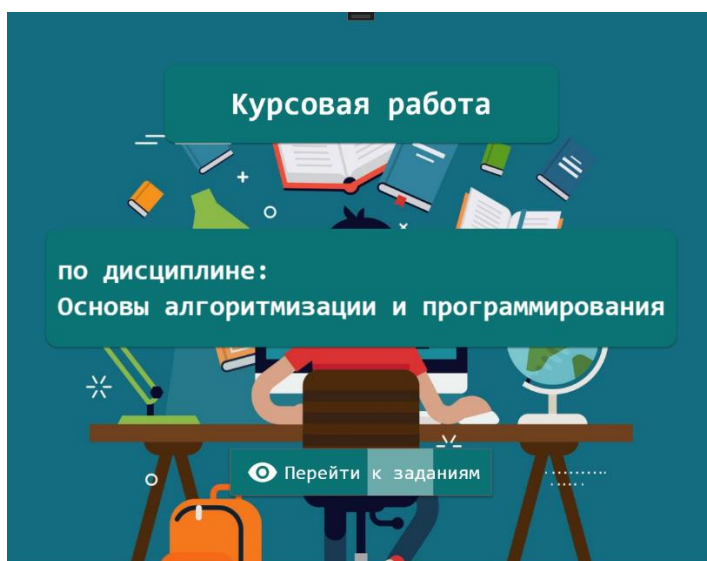


Рисунок 2.14 – MainPage

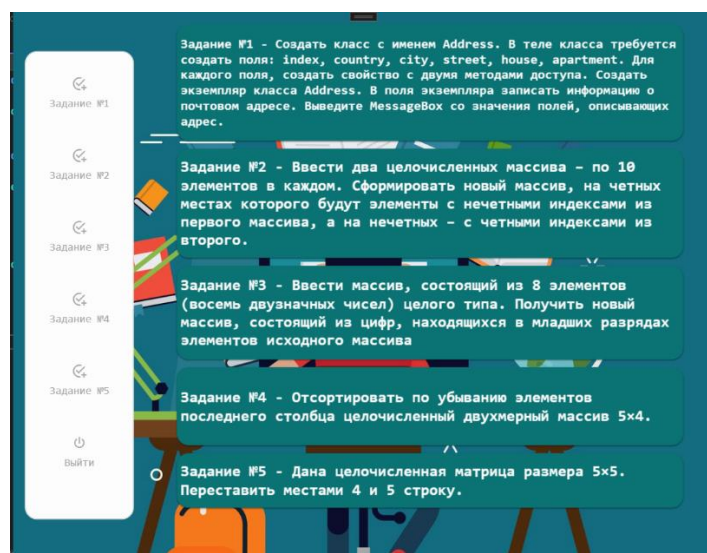


Рисунок 2.15 – TaskPage

На этом реализация «MainPage» и «TaskPage» закончена.

3. Реализация директории «PageTask»

Директория «PageTask» содержит пять страниц с заданиями. Реализацию начинаем с разметки «Task1Page».

```
<Page x:Class="CourseWorkApp.View.Pages.PageTask.Task1Page"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:local="clr-namespace:CourseWorkApp.View.Pages.PageTask"
xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
mc:Ignorable="d"
d:DesignHeight="700"
d:DesignWidth="900"
FontFamily="Consolas">
<Grid>
<Grid.ColumnDefinitions...>
<Image Grid.ColumnSpan="2".../>
<Border Background="#FDFDFD"...>
<Border Grid.Column="1"
HorizontalAlignment="Center"
VerticalAlignment="Top"
Margin="0 10 0 0"
Width="650"
Height="150"
Background="#0B7373"
CornerRadius="15"
Padding="5"
Effect="{StaticResource MaterialDesignElevationShadow1}">
<StackPanel HorizontalAlignment="Left"
VerticalAlignment="Center">
<TextBlock Text="Задание "b1 - Создать класс с именем Address.
FontSize="20"
TextWrapping="Wrap"
Foreground="White"
FontWeight="SemiBold"/>
</StackPanel>
</Border>
</Grid>
```

Рисунок 3.1 – Разметка «Task1Page»

```
<Border Grid.Column="1"
CornerRadius="15"
HorizontalAlignment="Center"
VerticalAlignment="Center"
Background="#0B7373"
Effect="{StaticResource MaterialDesignElevationShadow1}"
Width="450"
Height="300">
<StackPanel HorizontalAlignment="Center"
VerticalAlignment="Center">
<StackPanel Orientation="Horizontal">
<materialDesign:PackIcon Kind="CityVariant"
Margin="0 0 3 0"
VerticalAlignment="Center"
Foreground="White"
Width="25"
Height="25"/>
<TextBox Width="280"
Foreground="White"
FontSize="20"
x:Name="TbIndex"
materialDesign:HintAssist.Hint="Введите почтовый индекс"
materialDesign:TextFieldAssist.HasClearButton="True"
materialDesign:TextFieldAssist.UnderlineBrush="White"/>
</StackPanel>
</StackPanel>
```

Рисунок 3.2 – Разметка «Task1Page»

```

<StackPanel Orientation="Horizontal"
  Margin="0 5 0 0">
  <materialDesign:PackIcon Kind="Map"
    Margin="0 0 3 0"
    VerticalAlignment="Center"
    Foreground="■"White"
    Width="25"
    Height="25"/>
  <TextBox Width="280"
    Foreground="■"White"
    FontSize="20"
    x:Name="TbCountry"
    materialDesign:HintAssist.Hint="Введите страну"
    materialDesign:TextFieldAssist.HasClearButton="True"
    materialDesign:TextFieldAssist.UnderlineBrush="■"White"/>
</StackPanel>

<StackPanel Orientation="Horizontal"
  Margin="0 5 0 0">
  <materialDesign:PackIcon Kind="City"
    Margin="0 0 3 0"
    VerticalAlignment="Center"
    Foreground="■"White"
    Width="25"
    Height="25"/>
  <TextBox Width="280"
    Foreground="■"White"
    FontSize="20"
    x:Name="TbCity"
    materialDesign:HintAssist.Hint="Введите город"
    materialDesign:TextFieldAssist.HasClearButton="True"
    materialDesign:TextFieldAssist.UnderlineBrush="■"White"/>
</StackPanel>

```

Рисунок 3.3 – Разметка «Task1Page»

```

<StackPanel Orientation="Horizontal"
  Margin="0 5 0 0">
  <materialDesign:PackIcon Kind="Electricity"
    Margin="0 0 3 0"
    VerticalAlignment="Center"
    Foreground="■"White"
    Width="25"
    Height="25"/>
  <TextBox Width="280"
    Foreground="■"White"
    FontSize="20"
    x:Name="TbStreet"
    materialDesign:HintAssist.Hint="Введите улицу"
    materialDesign:TextFieldAssist.HasClearButton="True"
    materialDesign:TextFieldAssist.UnderlineBrush="■"White"/>
</StackPanel>

<StackPanel Orientation="Horizontal"
  Margin="0 5 0 0">
  <materialDesign:PackIcon Kind="House"
    Margin="0 0 3 0"
    VerticalAlignment="Center"
    Foreground="■"White"
    Width="25"
    Height="25"/>
  <TextBox Width="280"
    Foreground="■"White"
    FontSize="20"
    x:Name="TbHouse"
    materialDesign:HintAssist.Hint="Введите № дома"
    materialDesign:TextFieldAssist.HasClearButton="True"
    materialDesign:TextFieldAssist.UnderlineBrush="■"White"/>
</StackPanel>

```

Рисунок 3.4 – Разметка «Task1Page»

```

<StackPanel Orientation="Horizontal"
  Margin="0 5 0 0">
  <materialDesign:PackIcon Kind="CameraPartyMode"
    Margin="0 0 3 0"
    VerticalAlignment="Center"
    Foreground="■"White"
    Width="25"
    Height="25" />
  <TextBox Width="280"
    Foreground="■"White"
    FontSize="20"
    x:Name="TbApartment"
    materialDesign:HintAssist.Hint="Введите № квартиры"
    materialDesign:TextFieldAssist.HasClearButton="True"
    materialDesign:TextFieldAssist.UnderlineBrush="■"White" />
</StackPanel>

<Button Background="■"#0B7373"
  Foreground="■"White"
  x:Name="BtnGetData"
  Click="BtnGetData_Click"
  Margin="0 10 0 0"
  Style="{StaticResource MaterialDesignRaisedButton}"
  materialDesign:ButtonProgressAssist.Value="-1"
  materialDesign:ButtonProgressAssist.IndicatorForeground="■"White"
  materialDesign:ButtonProgressAssist.IsIndicatorVisible="True"
  materialDesign:ButtonProgressAssist.IsIndeterminate="True"
  Height="40">

```

Рисунок 3.5 – Разметка «Task1Page»

```

<StackPanel Orientation="Horizontal">
  <materialDesign:PackIcon Kind="Check"
    Margin="0 0 3 0"
    Width="25"
    Height="25" />
  <TextBlock Text="Получить данные"
    FontSize="20"
    VerticalAlignment="Center">
    <TextBlock.Style>
      <Style TargetType="TextBlock">
        <Style.Triggers>
          <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="TextDecorations" Value="Underline" />
          </Trigger>
        </Style.Triggers>
      </Style>
    </TextBlock.Style>
  </TextBlock>
</StackPanel>
</Button>

```

Рисунок 3.6 – Разметка «Task1Page»

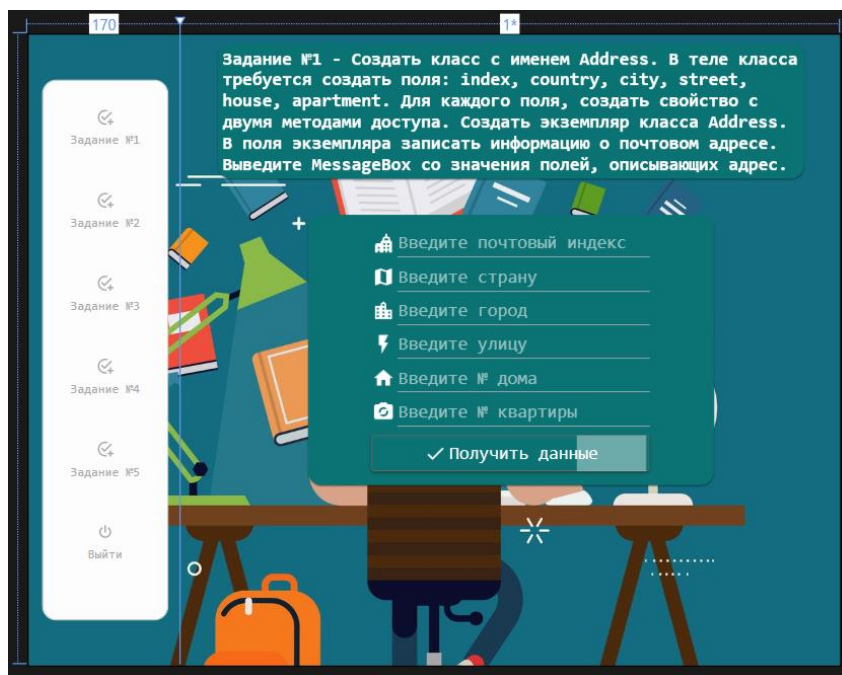


Рисунок 3.7 – Результат разметки «Task1Page»

Для реализации логики решения задания переходим в файл «Task1Page.xaml.cs». По заданию создадим класс «Address» с необходимыми полями, и передадим значение элемента «TextBox» соответствующим полям класса, при этом сделаем необходимые проверки на заполнение элемента «TextBox».

```

public class Address
{
    private int _index;
    private string? _country;
    private string? _city;
    private string? _street;
    private string? _house;
    private string? _apartment;
    Ссылка: 2
    public string? Apartment
    {
        get => _apartment;
        set => _apartment = value;
    }
    Ссылка: 2
    public string? House
    {
        get => _house;
        set => _house = value;
    }
    Ссылка: 2
    public string? Street
    {
        get => _street;
        set => _street = value;
    }
    Ссылка: 2
}

```

Рисунок 3.8 – Результат класса «Address»


```

public string? City
{
    get { return _city; }
    set { _city = value; }
}
Ссылка: 2
public string? Country
{
    get => _country;
    set => _country = value;
}
Ссылка: 2
public int Index
{
    get => _index;
    set => _index = value;
}

```

Рисунок 3.9 – Результат класса «Address»

```

#region Задание №1 – Решение
Ссылка: 1
private void BtnGetData_Click(object sender, RoutedEventArgs e)
{
    if (string.IsNullOrEmpty(TbIndex.Text) ||
        string.IsNullOrEmpty(TbCountry.Text) ||
        string.IsNullOrEmpty(TbCity.Text) ||
        string.IsNullOrEmpty(TbStreet.Text) ||
        string.IsNullOrEmpty(TbHouse.Text) ||
        string.IsNullOrEmpty(TbApartment.Text))
    {
        MessageBox.Show("Ошибка ввода данных",
            "Задание №1",
            MessageBoxButton.OK,
            MessageBoxImage.Error);
    }
    else
    {
        Address address = new();

        address.Index = Convert.ToInt32(TbIndex.Text);
        address.Country = TbCountry.Text;
        address.City = TbCity.Text;
        address.Street = TbStreet.Text;
        address.House = TbHouse.Text;
        address.Apartment = TbApartment.Text;
    }
}

```

Рисунок 3.10 – Решение задания №1

```

        MessageBox.Show($"Введенный индекс – {address.Index}\nВведенная страна – {address.Country}\n +
            $"Введенный город – {address.City}\nВведенная улица – {address.Street}\n +
            $"Введенный № дома – {address.House}\nВведенный № квартиры – {address.Apartment}",
            "Задание №1",
            MessageBoxButton.OK,
            MessageBoxImage.Information);
        ClearFields();
    }
}
Ссылка: 1
private void ClearFields()
{
    TbIndex.Text = string.Empty;
    TbCountry.Text = string.Empty;
    TbCity.Text = string.Empty;
    TbStreet.Text = string.Empty;
    TbHouse.Text = string.Empty;
    TbApartment.Text = string.Empty;
}

```

Рисунок 3.11 – Решение задания №1

Запустим приложение и проверим логику решения первого задания.



Рисунок 3.12 – Реализация «Task1Page»

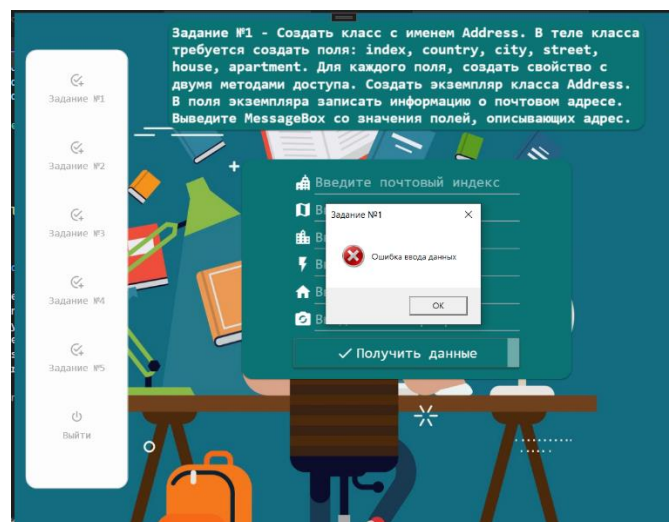


Рисунок 3.13 – Реализация «Task1Page»

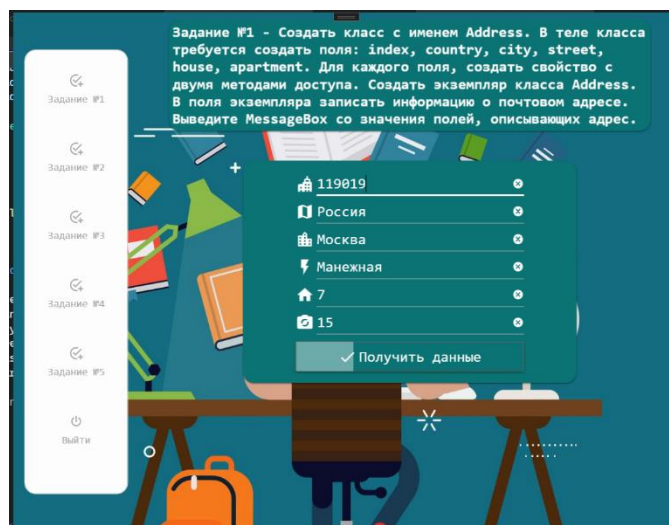


Рисунок 3.14 – Реализация «Task1Page»

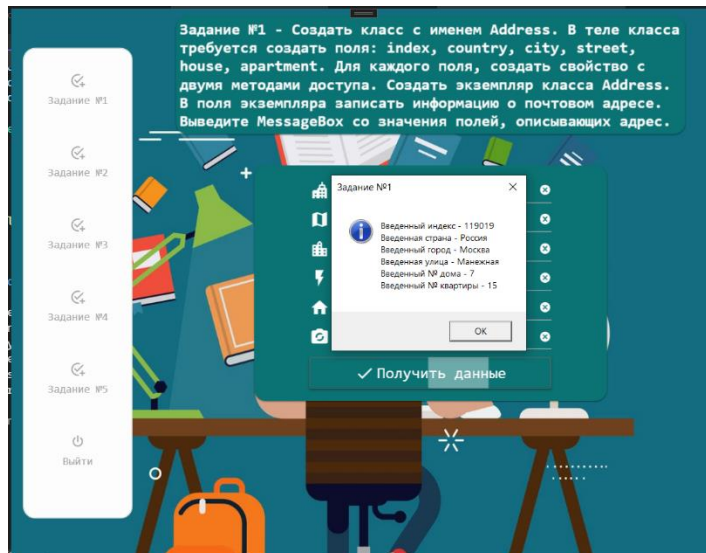


Рисунок 3.15 – Реализация «Task1Page»

Далее переходим к реализации разметки «Task2Page».

```

<Page x:Class="CourseWorkApp.View.Pages.PageTask.Task2Page"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:local="clr-namespace:CourseWorkApp.View.Pages.PageTask"
xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
mc:Ignorable="d"
d:DesignHeight="700"
d:DesignWidth="900"
FontFamily="Consolas">
<Grid>
<Grid.ColumnDefinitions...>
<Image Grid.ColumnSpan="2".../>
<Border Background="#FDFDFD"...>
<Border Grid.Column="1"
HorizontalAlignment="Center"
VerticalAlignment="Top"
Margin="0 10 0 0"
Width="630"
Height="140"
Background="#0B7373"
CornerRadius="15"
Padding="5"
Effect="{StaticResource MaterialDesignElevationShadow1}">
<StackPanel HorizontalAlignment="Left"
VerticalAlignment="Center">
<TextBlock Text="Задание №2 - Ввести два целочисленных массива"
FontSize="20"
TextWrapping="Wrap"
Foreground="White"
FontWeight="SemiBold"/>
</StackPanel>
</Border>

```

Рисунок 3.16 – Разметка «Task2Page»

```

<Border Grid.Column="1"
    HorizontalAlignment="Center"
    VerticalAlignment="Center"
    Width="450"
    Height="300"
    Background="#0B7373"
    CornerRadius="15"
    Padding="5"
    Effect="{StaticResource MaterialDesignElevationShadow1}">
  <StackPanel HorizontalAlignment="Center"
    VerticalAlignment="Center">
    <StackPanel>
      <TextBlock Text="Первый массив:"
        FontSize="20"
        Foreground="White"
        FontWeight="SemiBold" />
      <TextBox x:Name="TbFirsArray"
        Foreground="White"
        IsReadOnly="True"
        FontSize="20"
        Margin="0 5 0 0"
        materialDesign:TextFieldAssist.UnderlineBrush="White" />
    </StackPanel>
    <StackPanel Margin="0 5 0 0">
      <TextBlock Text="Второй массив:"
        FontSize="20"
        Foreground="White"
        FontWeight="SemiBold" />
      <TextBox x:Name="TbSecondArray"
        Foreground="White"
        IsReadOnly="True"
        FontSize="20"
        Margin="0 5 0 0"
        materialDesign:TextFieldAssist.UnderlineBrush="White" />
    </StackPanel>
  </StackPanel>

```

Рисунок 3.17 – Разметка «Task2Page»

```

<Button Background="#0B7373"
    Foreground="White"
    x:Name="BtnGetNewArray"
    Click="BtnGetNewArray_Click"
    Margin="0 10 0 0"
    Style="{StaticResource MaterialDesignRaisedButton}"
    materialDesign:ButtonProgressAssist.Value="-1"
    materialDesign:ButtonProgressAssist.IndicatorForeground="White"
    materialDesign:ButtonProgressAssist.IsIndicatorVisible="True"
    materialDesign:ButtonProgressAssist.IsIndeterminate="True"
    Height="40">
  <StackPanel Orientation="Horizontal">
    <materialDesign:PackIcon Kind="GetApp"
      Margin="0 1 3 0"
      Width="20"
      Height="20" />
    <TextBlock Text="Новый массив"
      FontSize="17"
      VerticalAlignment="Center">
      <TextBlock.Style>
        <Style TargetType="TextBlock">
          <Style.Triggers>
            <Trigger Property="IsMouseOver" Value="True">
              <Setter Property="TextDecorations" Value="Underline" />
            </Trigger>
          </Style.Triggers>
        </Style>
      </TextBlock.Style>
    </TextBlock>
  </StackPanel>
</Button>

```

Рисунок 3.18 – Разметка «Task2Page»

```

<StackPanel Margin="0 5 0 0"
    x:Name="SpNewArray"
    Visibility="Hidden">
    <TextBlock Text="Новый массив:"
        FontSize="20"
        Foreground="White"
        FontWeight="SemiBold"/>
    <TextBox x:Name="TbNewArray"
        Foreground="White"
        IsReadOnly="True"
        FontSize="20"
        Margin="0 5 0 0"
        materialDesign:TextFieldAssist.UnderlineBrush="White"/>
</StackPanel>

```

Рисунок 3.19 – Разметка «Task2Page»

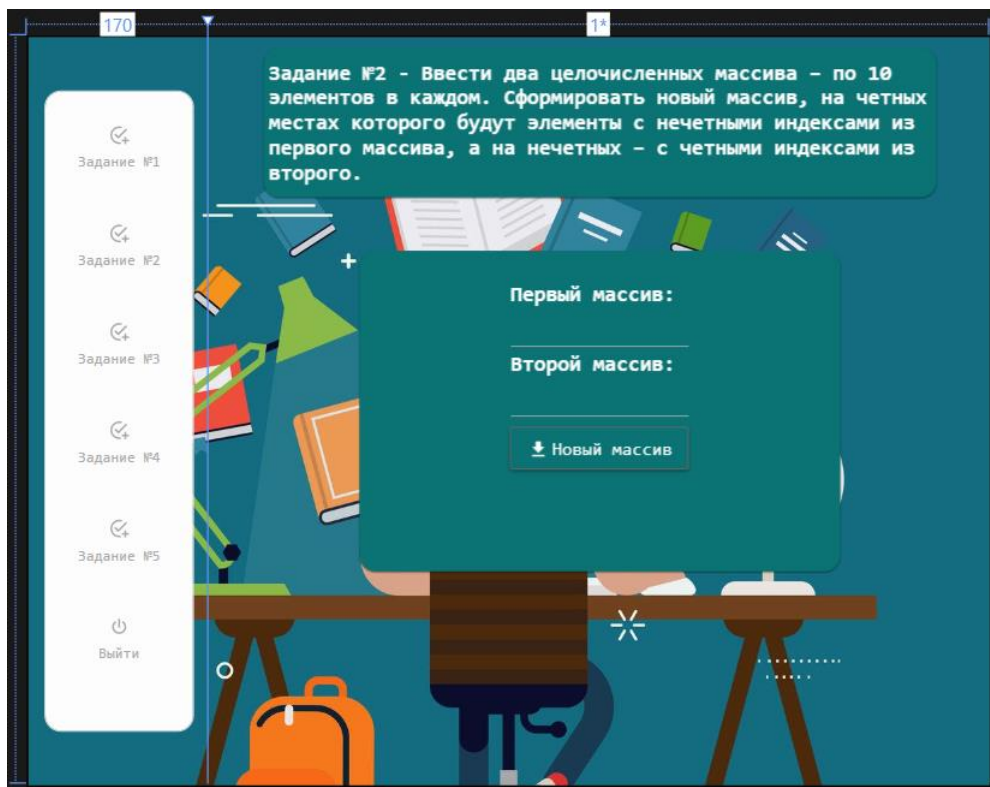


Рисунок 3.20 – Результат разметки «Task2Page»

Для реализации логики решения задания переходим в файл «Task2Page.xaml.cs». По заданию создадим три целочисленных массива, метод инициализации случайными данными и метод с решением.

```

#region Исходные данные
private int[] _firstArray = new int[10];
private int[] _secondArray = new int[10];
private int[] _newArray = new int[10];
#endregion

```

Рисунок 3.21 – Решение задания №2

```

#region Задание №2 – Решение
Ссылка: 1
private void InitializeArrays()
{
    Random random = new();

    for (int i = 0; i < _firstArray.Length; i++)
    {
        _firstArray[i] = random.Next(10, 100);

        TbFirsArray.Text += $" {_firstArray[i]}";
    }

    for (int i = 0; i < _secondArray.Length; i++)
    {
        _secondArray[i] = random.Next(10, 100);

        TbSecondArray.Text += $" {_secondArray[i]}";
    }
}

```

Рисунок 3.22 – Решение задания №2

```

Ссылка: 1
private void BtnGetNewArray_Click(Object sender, RoutedEventArgs e)
{
    SpNewArray.Visibility = Visibility.Visible;

    for (int i = 0; i < _newArray.Length; i++)
    {
        if (i % 2 == 0)
        {
            _newArray[i] = _firstArray[i + 1];
        }
        else
        {
            _newArray[i] = _secondArray[i - 1];
        }

        TbNewArray.Text += $" {_newArray[i]}";
    }
}
#endregion

```

Рисунок 3.23 – Решение задания №2

Запустим приложение и проверим логику решения второго задания.

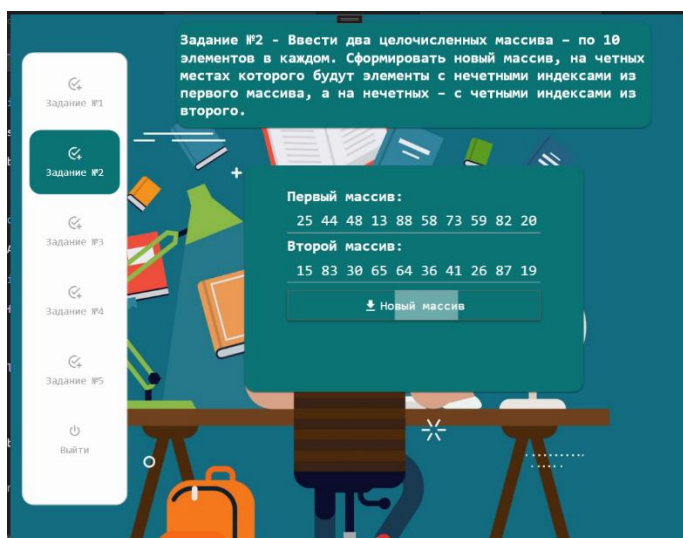


Рисунок 3.24 – Реализация «Task2Page»

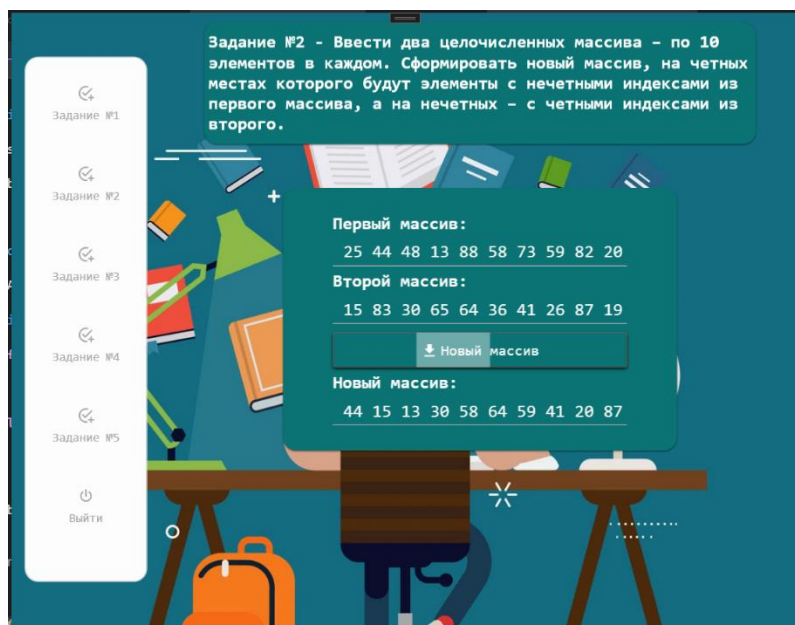


Рисунок 3.25 – Реализация «Task2Page»

Далее переходим к реализации разметки «Task3Page».

```

<Page x:Class="CourseWorkApp.View.Pages.PageTask.Task3Page"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:local="clr-namespace:CourseWorkApp.View.Pages.PageTask"
  xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
  mc:Ignorable="d"
  d:DesignHeight="700"
  d:DesignWidth="900"
  FontFamily="Consolas">
  <Grid>
    <Grid.ColumnDefinitions...>
    <Image Grid.ColumnSpan="2".../>
    <Border Background="#FDFDFD"...>
      <Border Grid.Column="1"
        HorizontalAlignment="Center"
        VerticalAlignment="Top"
        Margin="0 10 0 0"
        Width="600"
        Height="120"
        Background="#0B7373"
        CornerRadius="15"
        Padding="5"
        Effect="{StaticResource MaterialDesignElevationShadow1}">
        <StackPanel HorizontalAlignment="Left"
          VerticalAlignment="Center">
          <TextBlock Text="Задание %3 - Ввести массив, состоящий из 8 э
            FontSize="20"
            TextWrapping="Wrap"
            Foreground="White"
            FontWeight="SemiBold"/>
        </StackPanel>
      </Border>
    </Grid>
  </Page>

```

Рисунок 3.26 – Разметка «Task3Page»

```

<Border Grid.Column="1"
    HorizontalAlignment="Center"
    VerticalAlignment="Center"
    Width="350"
    Height="250"
    Background="#0B7373"
    CornerRadius="15"
    Padding="5"
    Effect="{StaticResource MaterialDesignElevationShadow1}">
    <StackPanel HorizontalAlignment="Center"
        VerticalAlignment="Center">
        <StackPanel>
            <TextBlock Text="Исходный массив: "
                FontSize="20"
                Foreground="White"
                FontWeight="SemiBold" />
            <TextBox x:Name="TbSourceArray"
                Foreground="White"
                IsReadOnly="True"
                FontSize="20"
                Margin="0 5 0 0"
                materialDesign:TextFieldAssist.UnderlineBrush="White" />
        </StackPanel>
        <Button Background="#0B7373"
            Foreground="White"
            x:Name="BtnGetNewArray"
            Click="BtnGetNewArray_Click"
            Margin="0 10 0 0"
            Style="{StaticResource MaterialDesignRaisedButton}"
            materialDesign:ButtonProgressAssist.Value="-1"
            materialDesign:ButtonProgressAssist.IndicatorForeground="White"
            materialDesign:ButtonProgressAssist.IsIndicatorVisible="True"
            materialDesign:ButtonProgressAssist.IsIndeterminate="True"
            Height="40">
    </StackPanel>

```

Рисунок 3.27 – Разметка «Task3Page»

```

<StackPanel Orientation="Horizontal">
    <materialDesign:PackIcon Kind="ArrangeBringToFront"
        Margin="0 1 3 0"
        Width="20"
        Height="20" />
    <TextBlock Text="Новый массив"
        FontSize="17"
        VerticalAlignment="Center">
        <TextBlock.Style>
            <Style TargetType="TextBlock">
                <Style.Triggers>
                    <Trigger Property="IsMouseOver" Value="True">
                        <Setter Property="TextDecorations" Value="Underline" />
                    </Trigger>
                </Style.Triggers>
            </Style>
        </TextBlock.Style>
    </TextBlock>
</StackPanel>
</Button>

<StackPanel Margin="0 10 0 0"
    x:Name="SpNewArray"
    Visibility="Hidden">
    <TextBlock Text="Новый массив: "
        FontSize="20"
        Foreground="White"
        FontWeight="SemiBold" />
    <TextBox x:Name="TbNewArray"
        Foreground="White"
        IsReadOnly="True"
        FontSize="20"
        Margin="0 5 0 0"
        materialDesign:TextFieldAssist.UnderlineBrush="White" />
</StackPanel>
</StackPanel>

```

Рисунок 3.28 – Разметка «Task3Page»

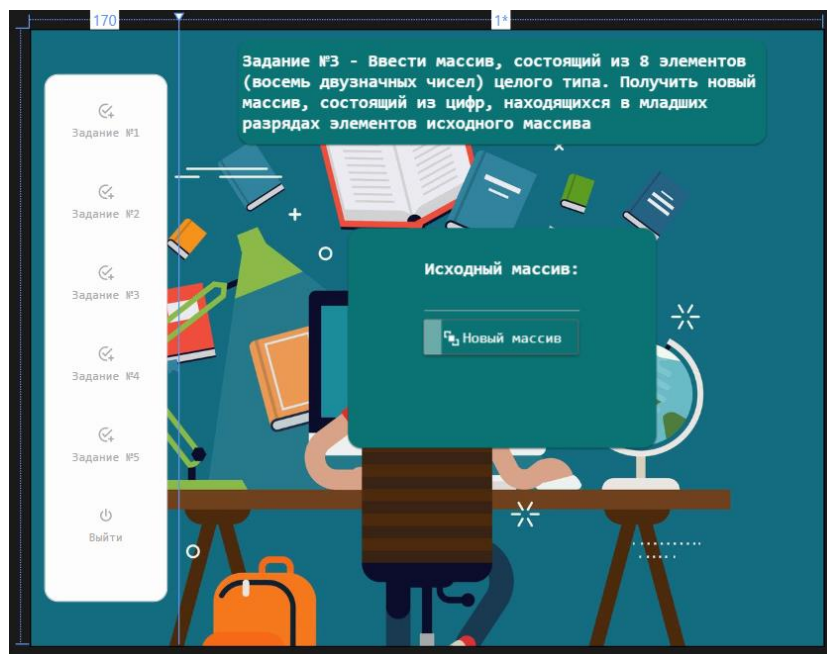


Рисунок 3.28 – Результат разметки «Task3Page»

Для реализации логики решения задания переходим в файл «Task3Page.xaml.cs». По заданию создадим два целочисленных массива, метод инициализации случайными данными и метод с решением.

```
#region Исходные данные
private int[]? _sourceArray;
private int[]? _newArray;
#endregion
```

Рисунок 3.29 – Решение задания №3

```
#region Задание %3 - Решение
Ссылка: 1
private void InitializeArray()
{
    Random random = new();
    _sourceArray = Enumerable.Range(0, 8).Select(g => random.Next(10, 100)).ToArray();

    for (int i = 0; i < _sourceArray.Length; i++)
    {
        TbSourceArray.Text += $" {_sourceArray[i]}";
    }
}
Ссылка: 1
private void BtnGetNewArray_Click(object sender, RoutedEventArgs e)
{
    SpNewArray.Visibility = Visibility.Visible;
    _newArray = _sourceArray?.Select(g => g % 10).ToArray();

    for (int i = 0; i < _newArray?.Length; i++)
    {
        TbNewArray.Text += $" {_newArray[i]}";
    }
}
#endregion
```

Рисунок 3.30 – Решение задания №3

Запустим приложение и проверим логику решения третьего задания.



Рисунок 3.31 – Реализация «Task3Page»

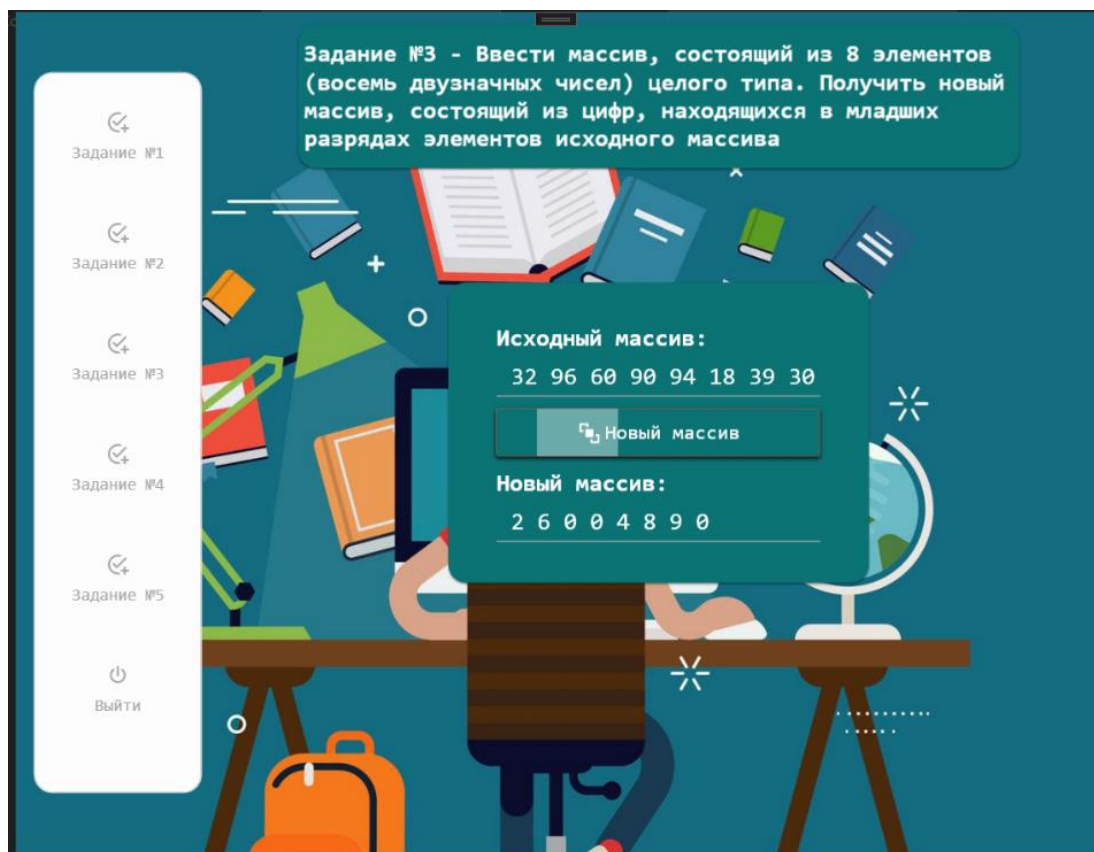


Рисунок 3.32 – Реализация «Task3Page»

Далее переходим к реализации разметки «Task4Page».

```

<Page x:Class="CourseWorkApp.View.Pages.PageTask.Task4Page"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:local="clr-namespace:CourseWorkApp.View.Pages.PageTask"
xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
mc:Ignorable="d"
d:DesignHeight="700"
d:DesignWidth="900"
FontFamily="Consolas">

  <Grid>
    <Grid.ColumnDefinitions...>
    <Image Grid.ColumnSpan="2" .../>
    <Border Background="#FDFDFD"...>

      <Border Grid.Column="1"
        HorizontalAlignment="Center"
        VerticalAlignment="Top"
        Margin="0 10 0 0"
        Width="580"
        Height="100"
        Background="#0B7373"
        CornerRadius="15"
        Padding="5"
        Effect="{StaticResource MaterialDesignElevationShadow1}">

        <StackPanel HorizontalAlignment="Left"
          VerticalAlignment="Center">
          <TextBlock Text="Задание %4 - Отсортировать по убыванию элемент"
            FontSize="20"
            TextWrapping="Wrap"
            Foreground="White"
            FontWeight="SemiBold"/>
        </StackPanel>
      </Border>
    </Grid>
  </Page>

```

Рисунок 3.33 – Разметка «Task4Page»

```

<Border Grid.Column="1"
  HorizontalAlignment="Center"
  VerticalAlignment="Center"
  Width="400"
  Height="400"
  Background="#0B7373"
  CornerRadius="15"
  Padding="5"
  Effect="{StaticResource MaterialDesignElevationShadow1}">

  <StackPanel HorizontalAlignment="Center"
    VerticalAlignment="Center">
    <StackPanel Margin="0 10 0 0">
      <TextBlock Text="Исходный массив:"
        FontSize="20"
        Foreground="White"
        FontWeight="SemiBold"/>

      <TextBlock x:Name="TbSourceArray"
        Foreground="White"
        FontSize="20"
        Margin="0 5 0 0"/>
    </StackPanel>

    <Button Background="#0B7373"
      Foreground="White"
      x:Name="BtnSortArray"
      Click="BtnSortArray_Click"
      Margin="0 10 0 0"
      Style="{StaticResource MaterialDesignRaisedButton}"
      materialDesign:ButtonProgressAssist.Value="-1"
      materialDesign:ButtonProgressAssist.IndicatorForeground="White"
      materialDesign:ButtonProgressAssist.IsIndicatorVisible="True"
      materialDesign:ButtonProgressAssist.IsIndeterminate="True"
      Height="40">

```

Рисунок 3.34 – Разметка «Task4Page»

```

<StackPanel Orientation="Horizontal">
  <materialDesign:PackIcon Kind="Sort"
    Margin="0 13 0"
    Width="20"
    Height="20"/>
  <TextBlock Text="Отсортировать массив"
    FontSize="17"
    VerticalAlignment="Center">
    <TextBlock.Style>
      <Style TargetType="TextBlock">
        <Style.Triggers>
          <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="TextDecorations" Value="Underline" />
          </Trigger>
        </Style.Triggers>
      </Style>
    </TextBlock.Style>
  </TextBlock>
</StackPanel>
</Button>

<StackPanel Margin="0 10 0 0"
  x:Name="SpSortedArray"
  Visibility="Hidden">
  <TextBlock Text="Отсортированный массив:"
    FontSize="20"
    Foreground="White"
    FontWeight="SemiBold" />

  <TextBlock x:Name="TbSortedArray"
    Foreground="White"
    FontSize="20"
    Margin="0 5 0 0" />
</StackPanel>
</StackPanel>

```

Рисунок 3.35 – Разметка «Task4Page»

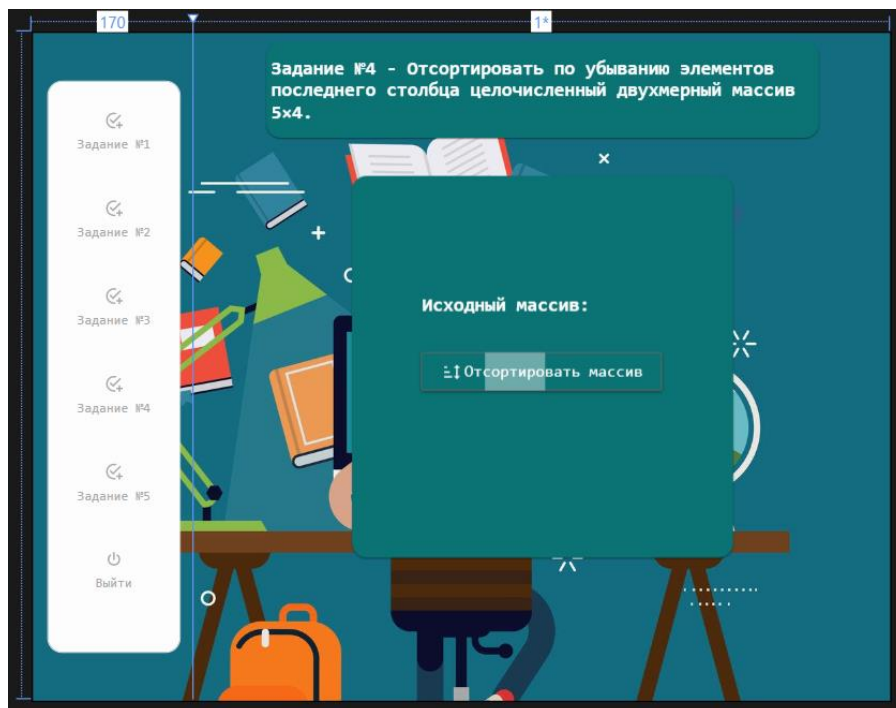


Рисунок 3.36 – Результат разметки «Task4Page»

Для реализации логики решения задания переходим в файл «Task4Page.xaml.cs». По заданию создадим целочисленный двумерный массив, метод инициализации случайными данными и метод с решением.

```

#region Исходные данные
int n, m, t;
private int[,] _sourceArray = new int[5, 4];
#endregion

```

Рисунок 3.37 – Решения задания №4

```

#region Задание №4 – Решение
Ссылка: 1
private void InitializeArray()
{
    Random random = new();

    for (n = 0; n < 5; n++)
    {
        for (m = 0; m < 4; m++)
        {
            _sourceArray[n, m] = random.Next(10, 100);
            TbSourceArray.Text += $"{_sourceArray[n, m]}\t";
        }
        TbSourceArray.Text += "\n";
    }
}

```

Рисунок 3.38 – Решения задания №4

```

private void BtnSortArray_Click(object sender, RoutedEventArgs e)
{
    SpSortedArray.Visibility = Visibility.Visible;

    m = 4;
    n = 5;

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n - 1; j++)
        {
            if (_sourceArray[i, m - 1] > _sourceArray[j, m - 1])
            {
                t = _sourceArray[i, m - 1];
                _sourceArray[i, m - 1] = _sourceArray[j, m - 1];
                _sourceArray[j, m - 1] = t;
            }
        }
    }

    for (n = 0; n < 5; n++)
    {
        for (m = 0; m < 4; m++)
        {
            TbSortedArray.Text += $"{_sourceArray[n, m]}\t";
        }
        TbSortedArray.Text += "\n";
    }
}

```

Рисунок 3.39 – Решения задания №4

Запустим приложение и проверим логику решения четвертого задания.



Рисунок 3.40 – Реализация «Task4Page»

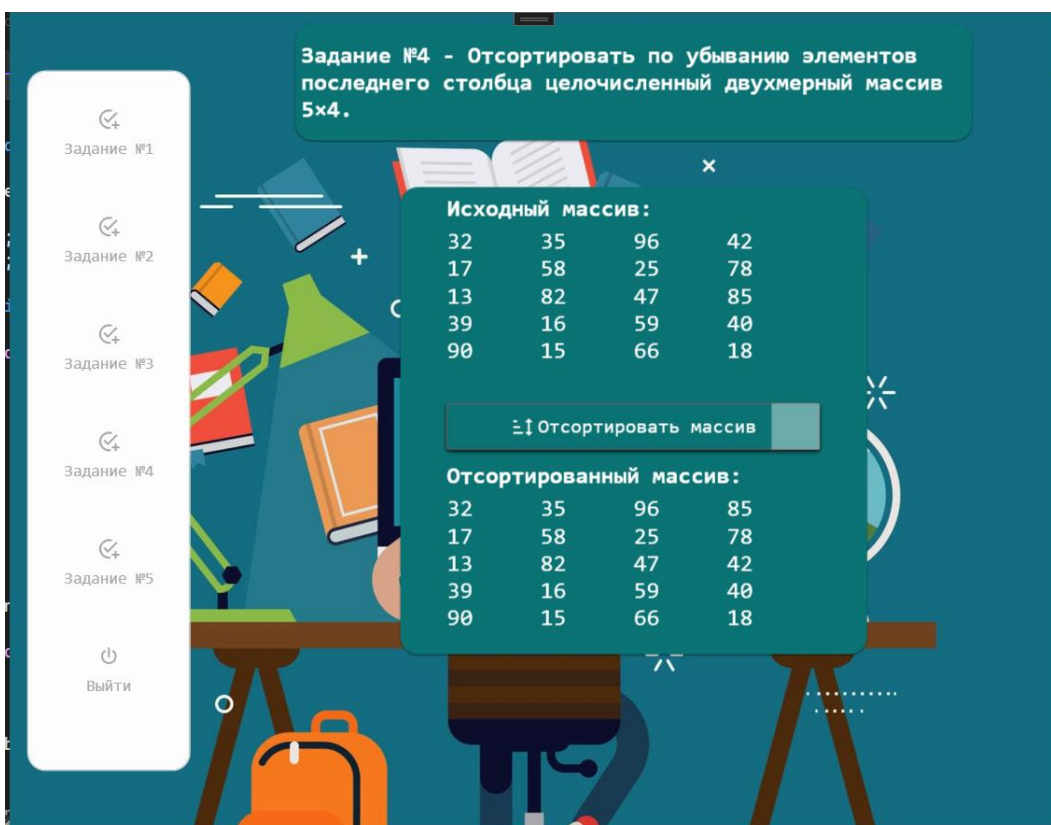


Рисунок 3.41 – Реализация «Task4Page»

Далее переходим к реализации разметки «Task5Page».

```

<Page x:Class="CourseWorkApp.View.Pages.PageTask.Task5Page"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:local="clr-namespace:CourseWorkApp.View.Pages.PageTask"
xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
mc:Ignorable="d"
d:DesignHeight="700"
d:DesignWidth="900"
FontFamily="Consolas">
<Grid>
<Grid.ColumnDefinitions...>
<Image Grid.ColumnSpan="2".../>
<Border Background="#FDFDFD"...>
<Border Grid.Column="1"
HorizontalAlignment="Center"
VerticalAlignment="Top"
Margin="0 10 0 0"
Width="580"
Height="70"
Background=#0B7373"
CornerRadius="15"
Padding="5"
Effect="{StaticResource MaterialDesignElevationShadow1}">
<StackPanel HorizontalAlignment="Left"
VerticalAlignment="Center">
<TextBlock Text="Задание %5 - Дана целочисленная матрица раз
FontSize="20"
TextWrapping="Wrap"
Foreground="White"
FontWeight="SemiBold"/>
</StackPanel>
</Border>

```

Рисунок 3.42 – Разметка «Task5Page»

```

<Border Grid.Column="1"
HorizontalAlignment="Center"
VerticalAlignment="Center"
Width="400"
Height="400"
Background=#0B7373"
CornerRadius="15"
Padding="5"
Effect="{StaticResource MaterialDesignElevationShadow1}">
<StackPanel HorizontalAlignment="Center"
VerticalAlignment="Center">
<StackPanel Margin="0 10 0 0">
<TextBlock Text="Исходный массив:"
FontSize="20"
Foreground="White"
FontWeight="SemiBold"/>
<TextBlock x:Name="TbSourceArray"
Foreground="White"
FontSize="20"
Margin="0 5 0 0"/>
</StackPanel>
<Button Background=#0B7373"
Foreground="White"
x:Name="BtnModified"
Click="BtnModified_Click"
Margin="0 10 0 0"
Style="{StaticResource MaterialDesignRaisedButton}"
materialDesign:ButtonProgressAssist.Value="-1"
materialDesign:ButtonProgressAssist.IndicatorForeground="White"
materialDesign:ButtonProgressAssist.IsIndicatorVisible="True"
materialDesign:ButtonProgressAssist.IsIndeterminate="True"
Height="40">

```

Рисунок 3.43 – Разметка «Task5Page»


```

<StackPanel Orientation="Horizontal">
  <materialDesign:PackIcon Kind="ModeComment"
    Margin="0 1 3 0"
    Width="20"
    Height="20"/>
  <TextBlock Text="Изменить массив"
    FontSize="17"
    VerticalAlignment="Center">
    <TextBlock.Style>
      <Style TargetType="TextBlock">
        <Style.Triggers>
          <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="TextDecorations" Value="Underline" />
          </Trigger>
        </Style.Triggers>
      </Style>
    </TextBlock.Style>
  </TextBlock>
</StackPanel>
</Button>

<StackPanel Margin="0 10 0 0"
  x:Name="SpSModifiedArray"
  Visibility="Hidden">
  <TextBlock Text="Измененный массив:"
    FontSize="20"
    Foreground="White"
    FontWeight="SemiBold"/>

  <TextBlock x:Name="TbSortedArray"
    Foreground="White"
    FontSize="20"
    Margin="0 5 0 0"/>
</StackPanel>
</StackPanel>

```

Рисунок 3.43 – Разметка «Task5Page»

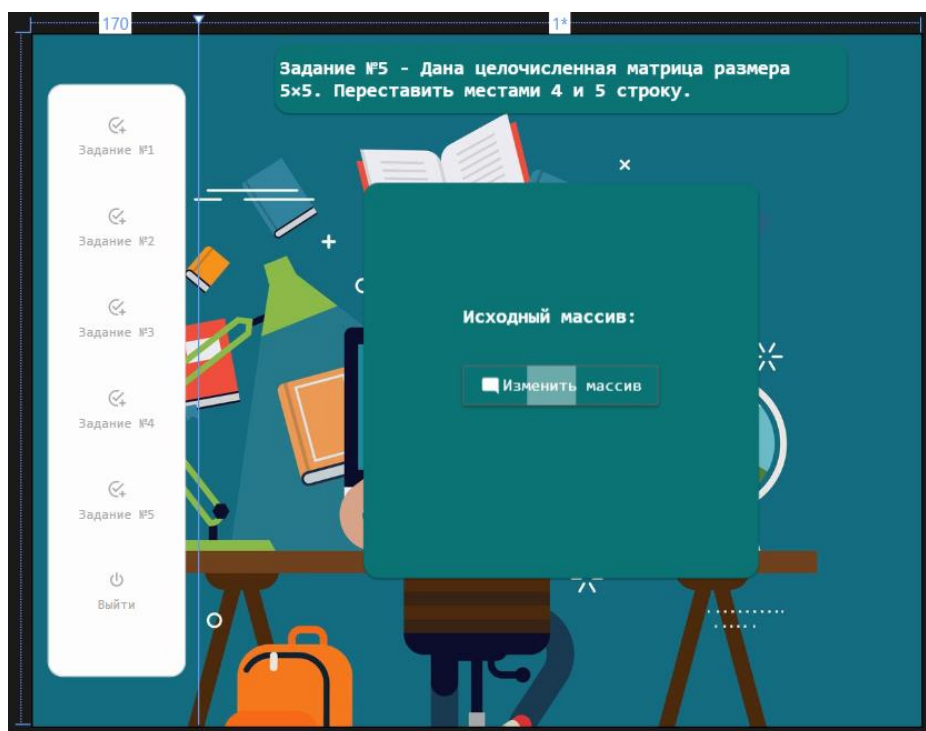


Рисунок 3.43 – Результат разметки «Task5Page»

Для реализации логики решения задания переходим в файл «Task5Page.xaml.cs». По заданию создадим целочисленную матрицу, метод инициализации случайными данными и метод с решением.

```
#region Исходные данные
private int n = 5;
private int[,]? _sourceArray;
#endregion
```

Рисунок 3.44 – Решение задания №5

```
#region Задание №5 – Решение
Ссылка: 1
private void InitializeArray()
{
    _sourceArray = new int[n, n];
    Random random = new ();
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            _sourceArray[i, j] = random.Next(10, 100);
            TbSourceArray.Text += $"{_sourceArray[i, j]}\t";
        }
        TbSourceArray.Text += "\n";
    }
}
Ссылка: 1
```

Рисунок 3.45 – Решение задания №5

```
Ссылка: 1
private void BtnModified_Click(object sender, RoutedEventArgs e)
{
    SpSModifiedArray.Visibility = Visibility.Visible;

    for (int i = 0; i < _sourceArray?.GetLength(1); i++)
    {
        (_sourceArray[3, i], _sourceArray[4, i]) = (_sourceArray[4, i], _sourceArray[3, i]);
    }
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            TbSortedArray.Text += $"{_sourceArray?[i, j]}\t";
        }
        TbSortedArray.Text += "\n";
    }
}
Ссылка: 1
```

Рисунок 3.45 – Решение задания №5

Запустим приложение и проверим логику решения пятого задания.

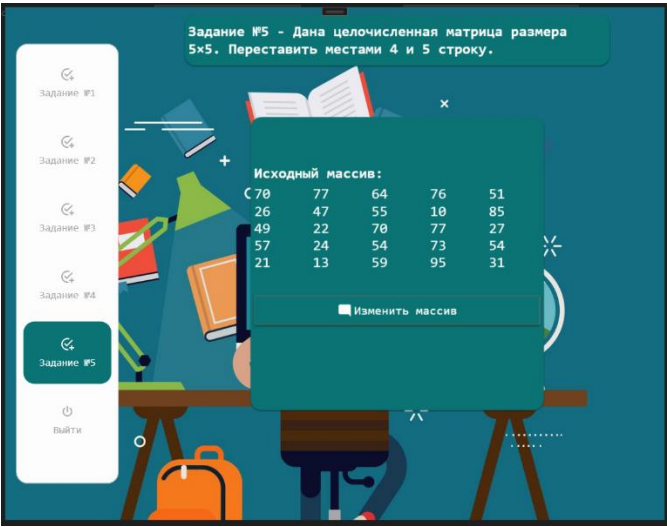


Рисунок 3.46 – Реализация «Task5Page»

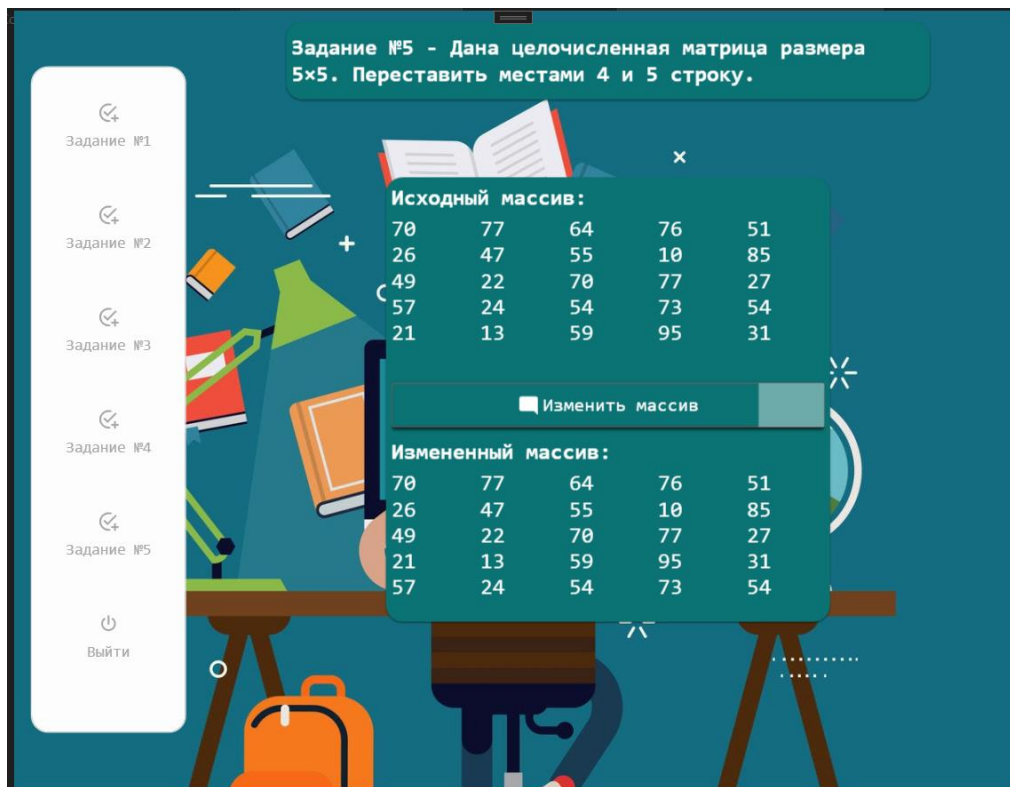


Рисунок 3.47 – Реализация «Task5Page»