

Министерство образования и молодежной политики Свердловской области  
ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Специальность: 09.02.07 Информационные системы и программирование

Практическая работа №7  
по дисциплине «Технология разработки программного обеспечения»  
на тему «Диспетчеры компоновки»

Выполнил:

студент гр. ПР-31

\_\_\_\_\_/Пинигина Д.А.

проверил:

\_\_\_\_\_/Гариев Д.О.

дата:

\_\_\_\_\_

оценка:

\_\_\_\_\_

## СОДЕРЖАНИЕ

ЦЕЛЬ .....	3
СОЗДАНИЕ ПРОЕКТА .....	4
CANVAS .....	6
WRAPPANEL .....	9
STACKPANEL.....	11
DOCKPANEL .....	13
GRID .....	15
Объединение ячеек.....	17
ЗАДАНИЕ.....	19
Родительский диспетчер Grid.....	19
Последние документы.....	20
Canvas дочерний.....	21
StackPanel дочерний .....	22
Дочерний WrapPanel .....	23
ВЫВОД.....	28

## **ЦЕЛЬ**

Цель: узнать, что такое диспетчер компоновки и его виды, разобраться в их отличиях и научиться создавать приложения с помощью диспетчеров.

## СОЗДАНИЕ ПРОЕКТА

Для начала работы с проектом необходимо создать проект, в котором мы будем работать. Для этого заходим в Visual Studio, выбираем вкладку *Создание проекта* (Рисунок 1), из предложенных вариантов выбираем *Приложение WPF* (Рисунок 2), далее указываем название проекта и его будущее расположение. Нажимаем *Создать*, и проект автоматически откроется (Рисунок 3).

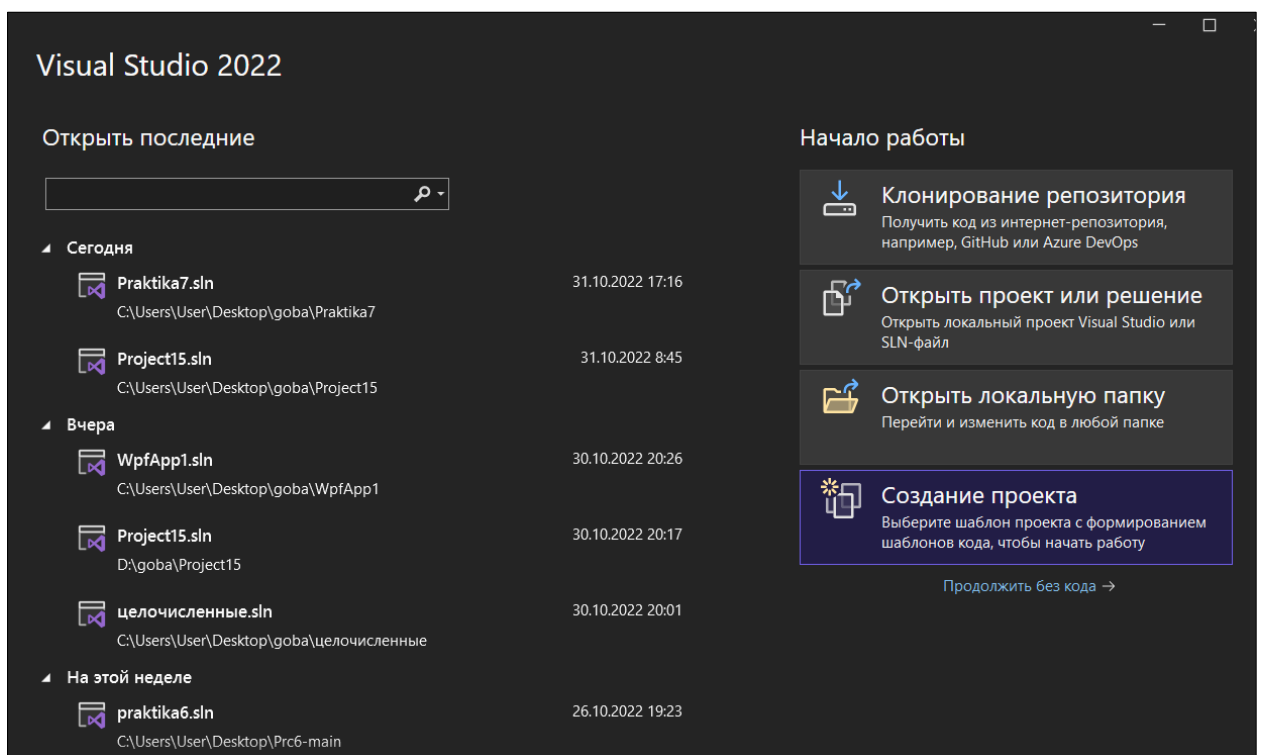


Рисунок 1 — Visual Studio

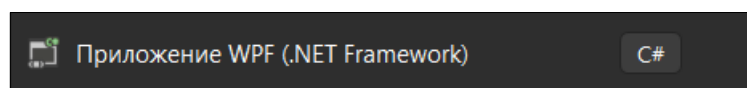


Рисунок 2 — WPF

Имя проекта

Prartika7

Расположение

C:\Users\User\Desktop\goba

Имя решения ⓘ

Prartika7

☐ Поместить решение и проект в одном каталоге

Платформа

.NET Framework 4.7.2

Рисунок 3 — Создание проекта

## CANVAS

Данный диспетчер компоновки имеет абсолютное позиционирование, т. е. если уменьшить окно, то содержимое станет невидимым до тех пор, пока не изменять размер окна (Рисунок 4). Для расположения используются Canvas.Left, Canvas.Top, Canvas.Bottom и т. д. Canvas также не поддерживает стили.

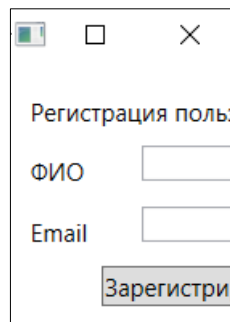


Рисунок 4 — Canvas

Примерный код (см. также Рисунок 5):

```
<Canvas>
    <Label Canvas.Left="10" Canvas.Top="10"
Content="Регистрация пользователя"></Label>
    <Label Canvas.Left="10" Canvas.Top="40" Con-
tent="ФИО"></Label>
    <TextBox Canvas.Left="70" Canvas.Top="40"
Width="200"></TextBox>
    <Label Canvas.Left="10" Canvas.Top="70" Con-
tent="Email"></Label>
    <TextBox Canvas.Left="70" Canvas.Top="70"
Width="200"></TextBox>
    <Button Canvas.Left="50" Canvas.Top="100" Con-
tent="Зарегистрироваться"></Button>
</Canvas>
```

Результатом будет окно для заполнения данных (Рисунок 6).

```
<Window x:Class="Praktika7.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:Praktika7"
  mc:Ignorable="d"
  Title="MainWindow" Height="450" Width="800">
  <Canvas>
    <Label Canvas.Left="10" Canvas.Top="10" Content="Регистрация пользователя"></Label>
    <Label Canvas.Left="10" Canvas.Top="40" Content="ФИО"></Label>
    <TextBox Canvas.Left="70" Canvas.Top="40" Width="200"></TextBox>
    <Label Canvas.Left="10" Canvas.Top="70" Content="Email"></Label>
    <TextBox Canvas.Left="70" Canvas.Top="70" Width="200"></TextBox>
    <Button Canvas.Left="50" Canvas.Top="100" Content="Зарегистрироваться"></Button>
  </Canvas>
</Window>
```

Рисунок 5 — Код Canvas

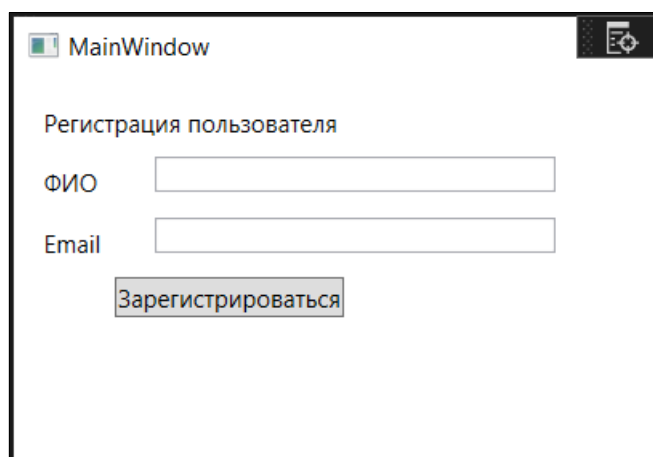


Рисунок 6 — Результат Canvas

Можно привязать элемент по-другому, используя другие атрибуты.

Пример кода (см. также Рисунок 7):

```
<Canvas>
    <Label Canvas.Left="10" Canvas.Top="10"
Content="Регистрация пользователя"></Label>
    <Label Canvas.Left="10" Canvas.Top="40" Con-
tent="ФИО"></Label>
    <TextBox Canvas.Left="70" Canvas.Top="40"
Width="200"></TextBox>
```

```

        <Label Canvas.Left="10" Canvas.Top="70" Content="Email"></Label>
        <TextBox Canvas.Left="70" Canvas.Top="70" Width="200"></TextBox>
        <Button Canvas.Right="10" Canvas.Bottom="10" Content="Зарегистрироваться"></Button>
    </Canvas>

```

Полученный результат отличается лишь положением кнопки (Рисунок 8).

```

<Window x:Class="Praktika7.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:Praktika7"
  mc:Ignorable="d"
  Title="MainWindow" Height="450" Width="800">
    <Canvas>
        <Label Canvas.Left="10" Canvas.Top="10" Content="Регистрация пользователя"></Label>
        <Label Canvas.Left="10" Canvas.Top="40" Content="ФИО"></Label>
        <TextBox Canvas.Left="70" Canvas.Top="40" Width="200"></TextBox>
        <Label Canvas.Left="10" Canvas.Top="70" Content="Email"></Label>
        <TextBox Canvas.Left="70" Canvas.Top="70" Width="200"></TextBox>
        <Button Canvas.Right="10" Canvas.Bottom="10" Content="Зарегистрироваться"></Button>
    </Canvas>
</Window>

```

Рисунок 7 — Изменение Canvas

Рисунок 8 — Измененный результат



## WRAPPANEL

Данный диспетчер компоновки располагает элементы слева направо или сверху вниз (всё зависит от выбранной ориентации). Также элементы будут смещаться при уменьшении окна, в отличие от Canvas.

Пример кода (см. также Рисунок 9):

```
<WrapPanel>
    <Label Content="Регистрация
пользователя"></Label>
    <Label Content="ФИО"></Label>
    <TextBox Width="200"></TextBox>
    <Label Content="Email"></Label>
    <TextBox Width="200"></TextBox>
    <Button Content="Зарегистрироваться"></Button>
</WrapPanel>
```

Результат нужно будет обязательно проверить (Рисунок 10 Рисунок 11 Рисунок 12).

```
title="MainWindow" height="450" width="800">
<WrapPanel>
    <Label Content="Регистрация пользователя"></Label>
    <Label Content="ФИО"></Label>
    <TextBox Width="200"></TextBox>
    <Label Content="Email"></Label>
    <TextBox Width="200"></TextBox>
    <Button Content="Зарегистрироваться"></Button>
</WrapPanel>
</Window>
```

Рисунок 9 — Код WrapPanel

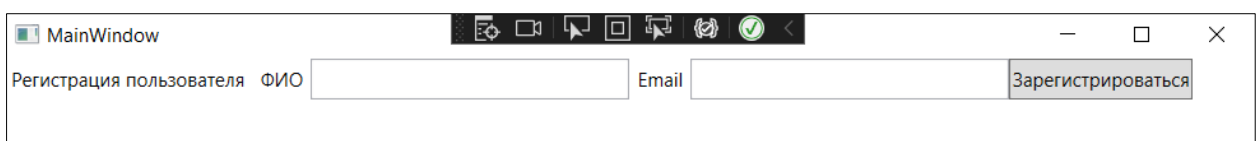


Рисунок 10 — В строку

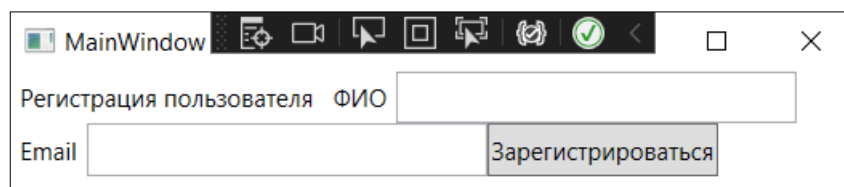


Рисунок 11 — Перемещение элементов

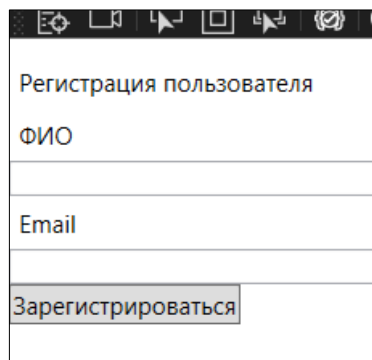


Рисунок 12 —Перемещение элементов в столбец

## STACKPANEL

StackPanel располагает элементы столбиком или строкой (в зависимости от выбранной ориентации). Можно указывать как будут растягиваться компоненты (HorizontalAlignment). Схож с Canvas тем, что при уменьшении экрана элементы, которые не попадают в окно, просто не отображаются (Рисунок 13).

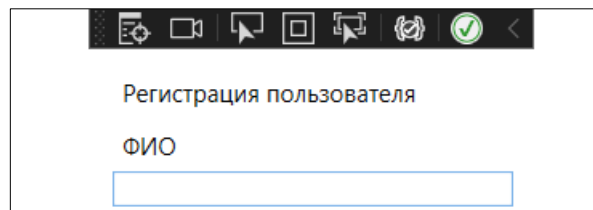


Рисунок 13 — Проверка StackPanel

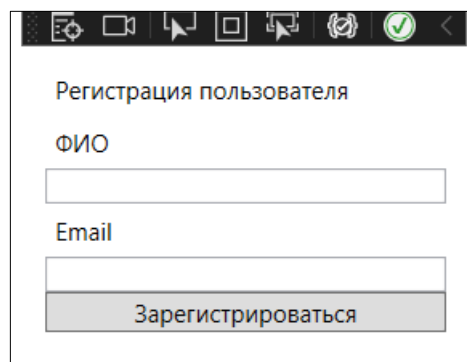
Пример кода (см. также Рисунок 14):

```
<StackPanel HorizontalAlignment="Center">  
    <Label Content="Регистрация  
пользователя"></Label>  
    <Label Content="ФИО"></Label>  
    <TextBox Width="200"></TextBox>  
    <Label Content="Email"></Label>  
    <TextBox Width="200"></TextBox>  
    <Button Content="Зарегистрироваться"></Button>  
</StackPanel>
```

```
<StackPanel HorizontalAlignment="Center">  
    <Label Content="Регистрация пользователя"></Label>  
    <Label Content="ФИО"></Label>  
    <TextBox Width="200"></TextBox>  
    <Label Content="Email"></Label>  
    <TextBox Width="200"></TextBox>  
    <Button Content="Зарегистрироваться"></Button>  
</StackPanel>
```

Рисунок 14 —Пример кода StackPanel

Результат (Рисунок 15).



The image shows a screenshot of a mobile application interface for user registration. At the top, there is a black header bar with several white icons: a camera, a magnifying glass, a square, a double arrow, a circular arrow, and a green checkmark. Below the header, the title 'Регистрация пользователя' (User Registration) is displayed. Underneath the title, there are two input fields: the first is labeled 'ФИО' (Full Name) and the second is labeled 'Email'. Both fields are empty. At the bottom of the form, there is a grey button with the text 'Зарегистрироваться' (Register).

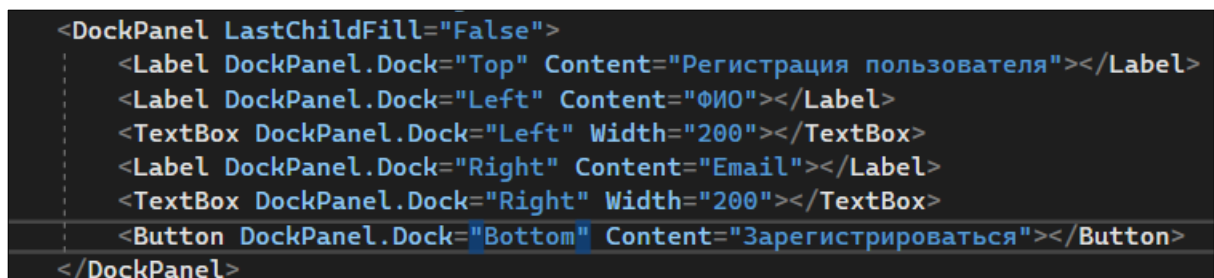
Рисунок 15 — Результат StackPanel

## DOCKPANEL

Данный диспетчер компоновки прикрепляет элементы к разным сторонам окна (Left, Right, Bottom, Top). DockPanel содержит атрибут LastChildFill, который определяет, будет ли последний элемент занимать оставшееся пространство.

Примерный код (см. также Рисунок 16):

```
<DockPanel LastChildFill="False">
    <Label DockPanel.Dock="Top" Content="Регистрация
пользователя"></Label>
    <Label DockPanel.Dock="Left" Content="ФИО"></La-
bel>
    <TextBox
                                DockPanel.Dock="Left"
Width="200"></TextBox>
    <Label
                                DockPanel.Dock="Right"
Content="Email"></Label>
    <TextBox
                                DockPanel.Dock="Right"
Width="200"></TextBox>
    <Button
                                DockPanel.Dock="Bottom"
Content="Зарегистрироваться"></Button>
</DockPanel>
```



```
<DockPanel LastChildFill="False">
  <Label DockPanel.Dock="Top" Content="Регистрация пользователя"></Label>
  <Label DockPanel.Dock="Left" Content="ФИО"></Label>
  <TextBox DockPanel.Dock="Left" Width="200"></TextBox>
  <Label DockPanel.Dock="Right" Content="Email"></Label>
  <TextBox DockPanel.Dock="Right" Width="200"></TextBox>
  <Button DockPanel.Dock="Bottom" Content="Зарегистрироваться"></Button>
</DockPanel>
```

Рисунок 16 — DockPanel код

Результат выйдет довольно специфичным, но тоже имеет место быть (Рисунок 17).

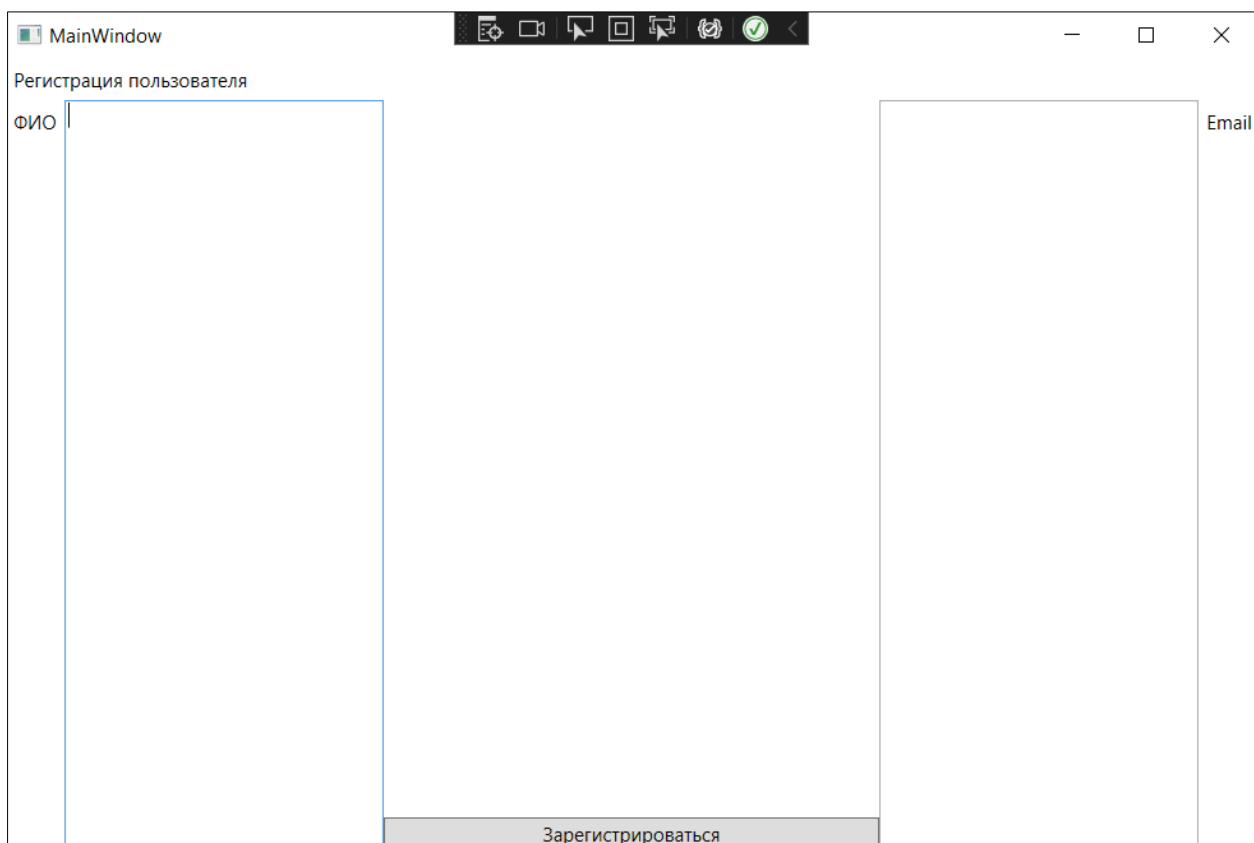


Рисунок 17 — Результат DockPanel

## GRID

Данный диспетчер компоновки представляет из себя сетку, в которой расположены все элементы. Используются атрибуты Column и Span. С помощью ColumnDefinitions (Row) мы указываем количество строк и столбцов, в которых будут располагаться дочерние элементы.

Примерный код содержимого (см. также Рисунок 18):

```
<Grid>
    <Grid.RowDefinitions>
        <RowDefinition></RowDefinition>
        <RowDefinition></RowDefinition>
        <RowDefinition></RowDefinition>
        <RowDefinition></RowDefinition>
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
        <ColumnDefinition></ColumnDefinition>
        <ColumnDefinition></ColumnDefinition>
    </Grid.ColumnDefinitions>
    <Label    Grid.Row="0"    Grid.Column="0"    Con-
tent="Регистрация пользователя" />
    <Label    Grid.Row="1"    Grid.Column="0"    Con-
tent="ФИО" />
    <TextBox      Grid.Row="1"      Grid.Column="1"
Width="200" />
    <Label    Grid.Row="2"    Grid.Column="0"    Con-
tent="Email" />
    <TextBox      Grid.Row="2"      Grid.Column="1"
Width="200" />
```

```

        <Button Grid.Row="3" Grid.Column="0" Content="Зарегистрироваться" />
    </Grid>

```

```

<Grid>
    <Grid.RowDefinitions>
        <RowDefinition></RowDefinition>
        <RowDefinition></RowDefinition>
        <RowDefinition></RowDefinition>
        <RowDefinition></RowDefinition>
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
        <ColumnDefinition></ColumnDefinition>
        <ColumnDefinition></ColumnDefinition>
    </Grid.ColumnDefinitions>
    <Label Grid.Row="0" Grid.Column="0" Content="Регистрация пользователя" />
    <Label Grid.Row="1" Grid.Column="0" Content="ФИО" />
    <TextBox Grid.Row="1" Grid.Column="1" Width="200" />
    <Label Grid.Row="2" Grid.Column="0" Content="Email" />
    <TextBox Grid.Row="2" Grid.Column="1" Width="200" />
    <Button Grid.Row="3" Grid.Column="0" Content="Зарегистрироваться" />
</Grid>

```

Рисунок 18 — Код Grid

Результат пока выглядит неопрятно, но вскоре это будет исправлено (Рисунок 19).

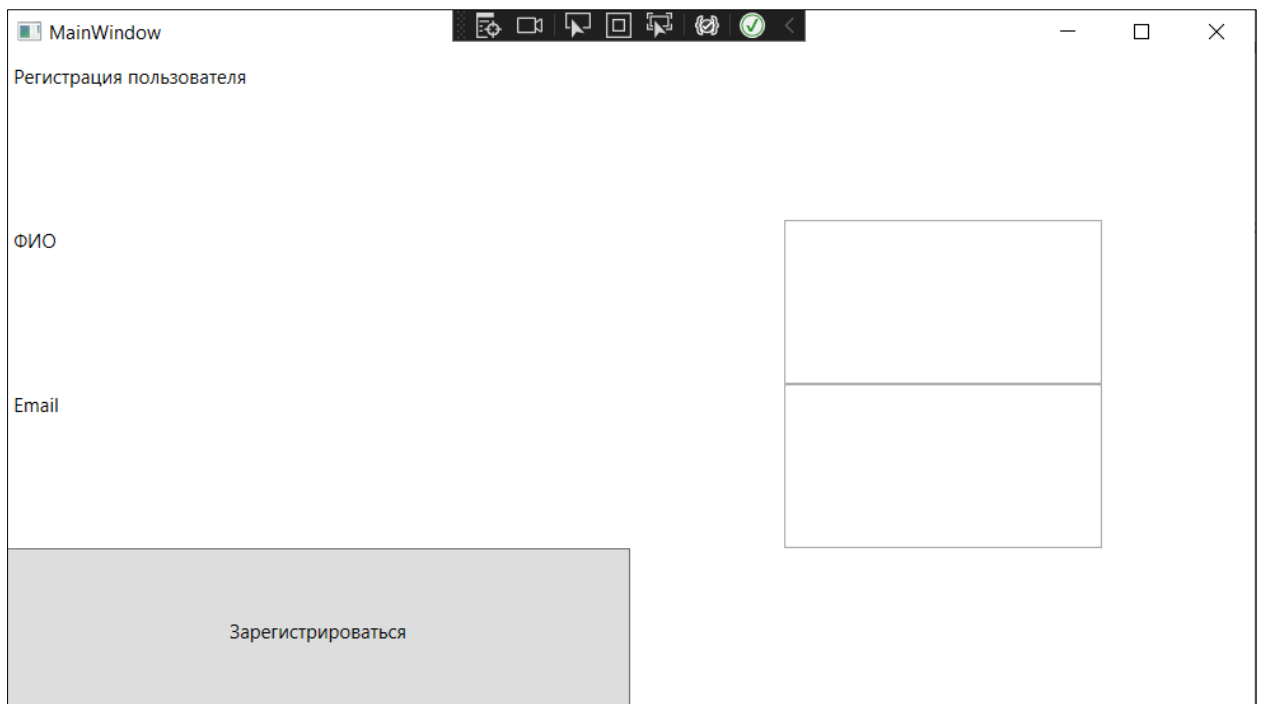


Рисунок 19 — Внешний вид Grid



## Объединение ячеек

Объединение ячеек происходит с помощью ColumnSpan и RowSpan.

Примерный код представлен ниже (см. также Рисунок 20):

```
<Grid>
    <Grid.RowDefinitions>
        <RowDefinition></RowDefinition>
        <RowDefinition></RowDefinition>
        <RowDefinition></RowDefinition>
        <RowDefinition></RowDefinition>
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
        <ColumnDefinition></ColumnDefinition>
        <ColumnDefinition></ColumnDefinition>
    </Grid.ColumnDefinitions>
    <Label Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="2" HorizontalAlignment="Center"
    Content="Регистрация пользователя" />
    <Label Grid.Row="1" Grid.Column="0" Content="ФИО" />
    <TextBox Grid.Row="1" Grid.Column="1" Width="200" />
    <Label Grid.Row="2" Grid.Column="0" Content="Email" />
    <TextBox Grid.Row="2" Grid.Column="1" Width="200" />
    <Button Grid.Row="3" Grid.Column="0" Grid.ColumnSpan="2" HorizontalAlignment="Center"
    Content="Зарегистрироваться" />
```

</Grid>

Результатом будут растянутые элементы на несколько столбиков и столбцов (Рисунок 21).

```
<Grid>
  <Grid.RowDefinitions>
    <RowDefinition></RowDefinition>
    <RowDefinition></RowDefinition>
    <RowDefinition></RowDefinition>
    <RowDefinition></RowDefinition>
  </Grid.RowDefinitions>
  <Grid.ColumnDefinitions>
    <ColumnDefinition></ColumnDefinition>
    <ColumnDefinition></ColumnDefinition>
  </Grid.ColumnDefinitions>
  <Label Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="2" HorizontalAlignment="Center"
    Content="Регистрация пользователя" />
  <Label Grid.Row="1" Grid.Column="0" Content="ФИО" />
  <TextBox Grid.Row="1" Grid.Column="1" Width="200" />
  <Label Grid.Row="2" Grid.Column="0" Content="Email" />
  <TextBox Grid.Row="2" Grid.Column="1" Width="200" />
  <Button Grid.Row="3" Grid.Column="0" Grid.ColumnSpan="2" HorizontalAlignment="Center"
    Content="Зарегистрироваться" />
</Grid>
```

Рисунок 20 — Код Grid Span

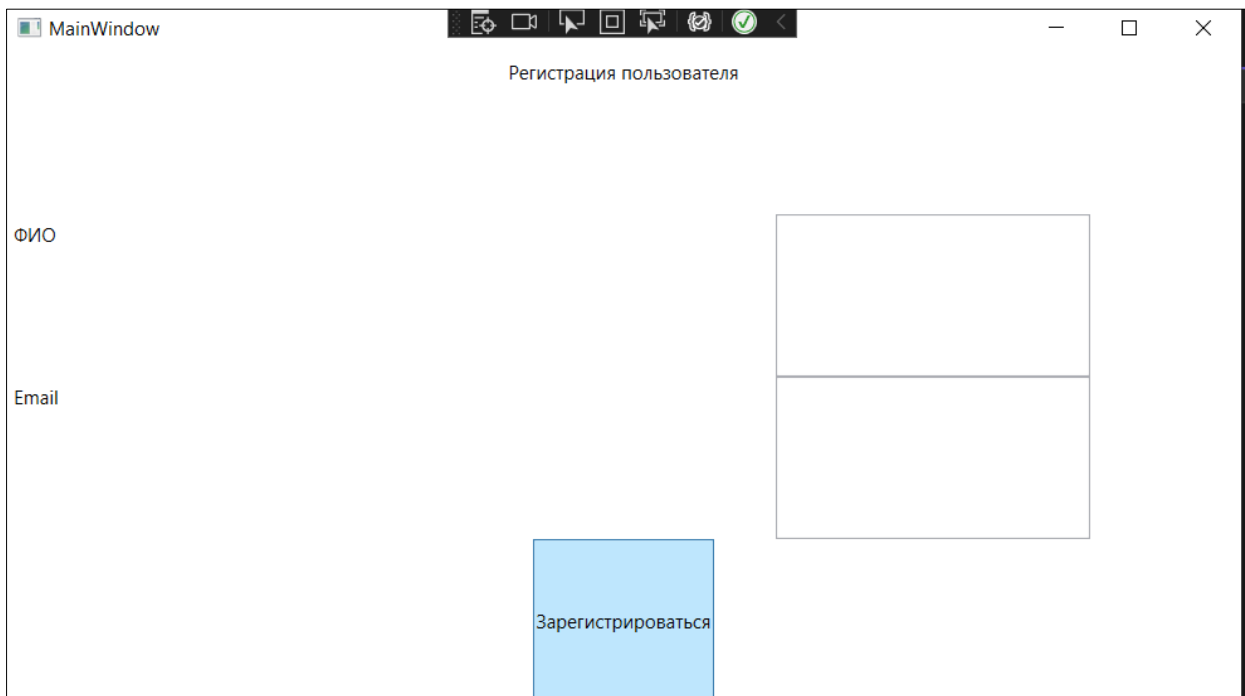


Рисунок 21 — Внешний вид Grid

## ЗАДАНИЕ

Теперь нам нужно самостоятельно сделать приложения, используя представленные ниже диспетчеры компоновки.

### Родительский диспетчер Grid

Для выполнения задания создадим диспетчер компоновки Grid. В нём будут располагаться дочерние диспетчеры. С помощью Grid.RowDefinitions и Grid.ColumnDefinitions укажем количество строк и столбцов, которых будут располагаться дочерние элементы. Строк будет 3, а столбцов 2. Размеры можем указать Auto, но кроме второго столбика. Там мы укажем \*.

Код данной части работы (см. также Рисунок 22):

```
<Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto">
        </RowDefinition>
        <RowDefinition Height="Auto">
        </RowDefinition>
        <RowDefinition Height="Auto">
        </RowDefinition>
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="Auto"></ColumnDefi-
nition>
        <ColumnDefinition Width="*"></ColumnDefini-
tion>
    </Grid.ColumnDefinitions>
</Grid>
```

```

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

```

Рисунок 22 — Код родительского диспетчера

## Последние документы

Данные элементы расположим в дочернем элементе StackPanel. Укажем вертикальную ориентацию, и расположим его в первой строке и первом столбце, указав, что данный диспетчер займёт 3 строки.

Внутри StackPanel можем указать элементы. В нашем случае это только Label.

Полученный код (см. также Рисунок 23):

```

<StackPanel      Orientation="Vertical"      Grid.Row="0"
Grid.Column="0" Grid.RowSpan="3">
    <Label      Content="Последние
документы"></Label>
    <Label Content="1.txt"></Label>
    <Label Content="2.txt"></Label>
    <Label Content="3.txt"></Label>
    <Label Content="4.txt"></Label>
    <Label Content="5.txt"></Label>
    <Label Content="6.txt"></Label>
</StackPanel>

```

```

<StackPanel Orientation="Vertical" Grid.Row="0" Grid.Column="0" Grid.RowSpan="3">
  <Label Content="Последние документы"></Label>
  <Label Content="1.txt"></Label>
  <Label Content="2.txt"></Label>
  <Label Content="3.txt"></Label>
  <Label Content="4.txt"></Label>
  <Label Content="5.txt"></Label>
  <Label Content="6.txt"></Label>

```

Рисунок 23 — Дочерний диспетчер StackPanel

В результате получим столбик из строк (Рисунок 24).

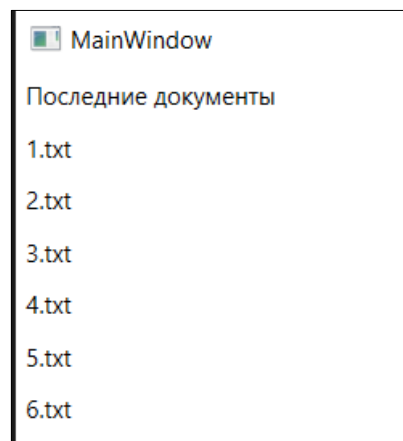


Рисунок 24 — Внешний вид родительского диспетчера StackPanel

## Canvas дочерний

Теперь можем продолжить создавать оставшиеся дочерние элементы.

Теперь создадим кнопки, которые располагаются в строку. Для этого можно воспользоваться диспетчером Canvas.

Расположим наш дочерний диспетчер в первую строку и во второй столбец. Внутри него указываем элементы и их расположение. В нашем случае – это кнопки с текстом (Content) и с координатами Canvas.Top и Canvas.Left.

В результате получим следующий код (см. также Рисунок 25):

```

<Canvas Grid.Column="1" Grid.Row="0">
  <Button Content="Открыть" Canvas.Left="0"
Canvas.Top="5"></Button>

```

```

        <Button Content="Сохранить" Canvas.Left="63"
Canvas.Top="5" ></Button>
        <Button Content="Закрыть" Canvas.Left="138"
Canvas.Top="5" ></Button>
        <Button Content="Обработать" Can-
vas.Left="200" Canvas.Top="5"></Button>
        <Button Content="О программе" Can-
vas.Left="281" Canvas.Top="5" ></Button>
    </Canvas>

```

```

<Canvas Grid.Column="1" Grid.Row="0">
    <Button Content="Открыть" Canvas.Left="0" Canvas.Top="5"></Button>
    <Button Content="Сохранить" Canvas.Left="63" Canvas.Top="5" ></Button>
    <Button Content="Закрыть" Canvas.Left="138" Canvas.Top="5" ></Button>
    <Button Content="Обработать" Canvas.Left="200" Canvas.Top="5"></Button>
    <Button Content="О программе" Canvas.Left="281" Canvas.Top="5" ></Button>
</Canvas>

```

Рисунок 25 — Код Canvas

В результате уже выполнена половина работы (Рисунок 26).

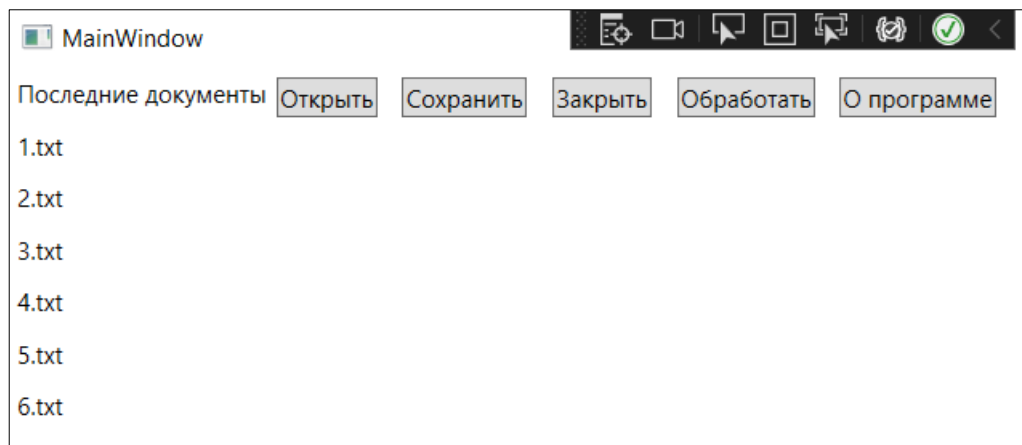


Рисунок 26 — Внешний вид Canvas

## StackPanel дочерний

Создадим ещё один дочерний диспетчер. Указываем горизонтальную ориентацию и располагаем элемент во второй строке и во втором столбце.

Внутри располагаем элементы Label, но в первом указываем Bold с помощью атрибута FontWeight.

Код результата (см. также Рисунок 27):

```
<StackPanel Orientation="Horizontal" Grid.Row="1"
Grid.Column="1">
    <Label Content="1.txt" Font-
Weight="Bold"></Label>
    <Label Content="2.txt"></Label>
    <Label Content="3.txt"></Label>
</StackPanel>
```

```
<StackPanel Orientation="Horizontal" Grid.Row="1" Grid.Column="1">
  <Label Content="1.txt" FontWeight="Bold"></Label>
  <Label Content="2.txt"></Label>
  <Label Content="3.txt"></Label>
</StackPanel>
```

Рисунок 27 — Ещё один дочерний StackPanel

Результат всё ближе движется к финалу (Рисунок 28).

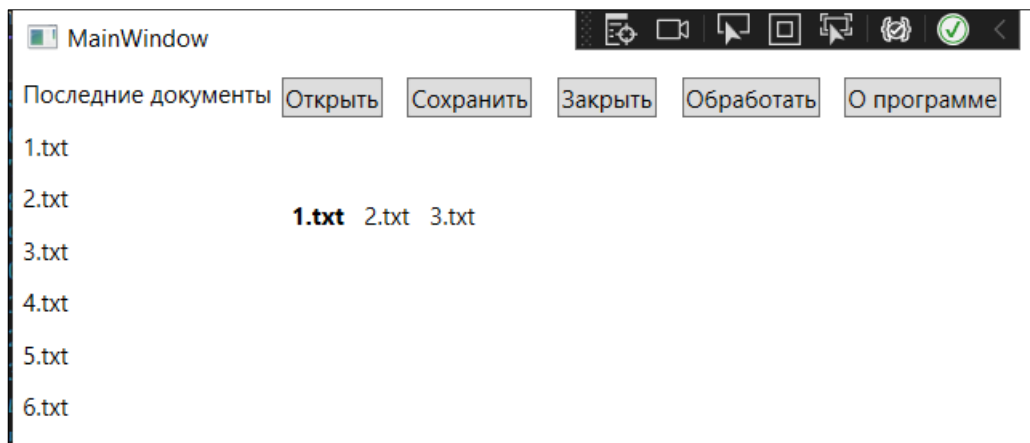


Рисунок 28 — Дочерний StackPanel

## Дочерний WrapPanel

Последний дочерний элемент будет WrapPanel, т.к. нам нужно, чтобы элементы смещались при уменьшении окна.

Располагаем диспетчер в третьей строке и во втором столбце. Ориентацию оставим по умолчанию. Внутри будет Label. Указываем фон (Background), ширина и высота будут по 50, текст располагаем по центру. (HorizontalAlignment или Vertical).

Полученный код:

```
<Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto">
        </RowDefinition>
        <RowDefinition Height="Auto">
        </RowDefinition>
        <RowDefinition Height="Auto">
        </RowDefinition>
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="Auto"></ColumnDefi-
nition>
        <ColumnDefinition Width="*"></ColumnDefini-
tion>
    </Grid.ColumnDefinitions>
    <StackPanel Orientation="Vertical" Grid.Row="0"
Grid.Column="0" Grid.RowSpan="3">
        <Label Content="Последние
документы"></Label>
        <Label Content="1.txt"></Label>
        <Label Content="2.txt"></Label>
        <Label Content="3.txt"></Label>
        <Label Content="4.txt"></Label>
        <Label Content="5.txt"></Label>
```



```

        <Label Content="6.txt"></Label>
    </StackPanel>
    <Canvas Grid.Column="1" Grid.Row="0">
        <Button Content="Открыть" Canvas.Left="0"
Canvas.Top="5"></Button>
        <Button Content="Сохранить" Canvas.Left="63"
Canvas.Top="5" ></Button>
        <Button Content="Заккрыть" Canvas.Left="138"
Canvas.Top="5" ></Button>
        <Button Content="Обработать" Can-
vas.Left="200" Canvas.Top="5"></Button>
        <Button Content="0 программе" Can-
vas.Left="281" Canvas.Top="5" ></Button>
    </Canvas>
    <StackPanel Orientation="Horizontal"
Grid.Row="1" Grid.Column="1">
        <Label Content="1.txt" Font-
Weight="Bold"></Label>
        <Label Content="2.txt"></Label>
        <Label Content="3.txt"></Label>
    </StackPanel>
    <WrapPanel Grid.Row="2" Grid.Column="1" >
        <Label Content="A" Background="Yellow"
Width="50" Height="50" HorizontalContentAlignment="Center"
VerticalContentAlignment="Center" Margin="0,0,0,0"></Label>
        <Label Content="B" Background="Red"
Width="50" Height="50" HorizontalContentAlignment="Center"
VerticalContentAlignment="Center" Margin="20,0,0,0"></Label>

```

```

        <Label      Content="C"      Background="Green"
Width="50"  Height="50"  HorizontalContentAlignment="Center"
VerticalContentAlignment="Center" Margin="20,0,0,0"></Label>
        <Label      Content="D"      Background="Blue"
Width="50"  Height="50"  HorizontalContentAlignment="Center"
VerticalContentAlignment="Center" Margin="20,0,0,0"></Label>
        <Label      Content="E"      Background="Gray"
Width="50"  Height="50"  HorizontalContentAlignment="Center"
VerticalContentAlignment="Center" Margin="20,0,0,0"></Label>
        <Label      Content="F"      Background="LightYellow"
Width="50"  Height="50"  HorizontalContentAlignment="Center"
VerticalContentAlignment="Center" Margin="20,0,0,0"></Label>
    </WrapPanel>
</Grid>

```

В конце мы получаем примерно такую картину, состоящую из кнопок и текста (Рисунок 29). Также не забываем проверить WrapPanel (Рисунок 30).

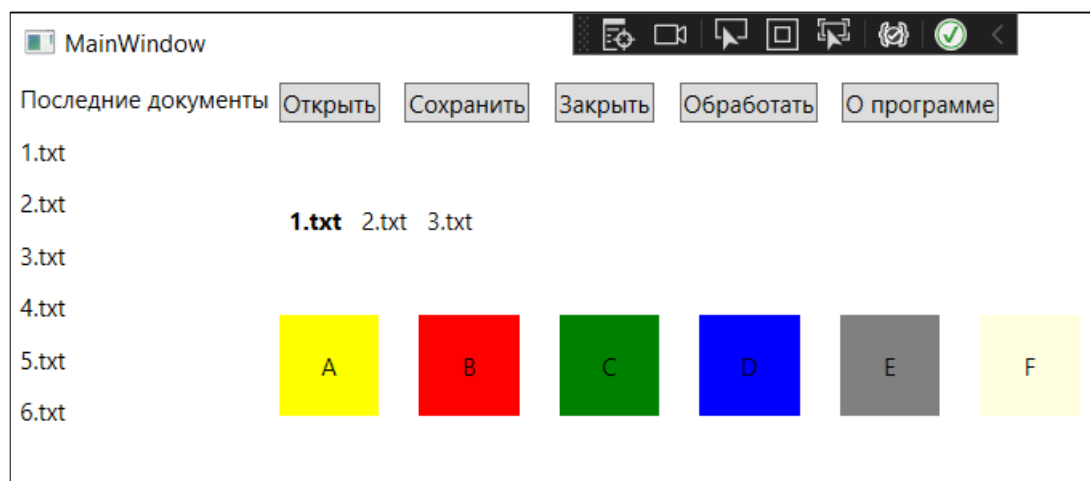


Рисунок 29 — Окончательный результат

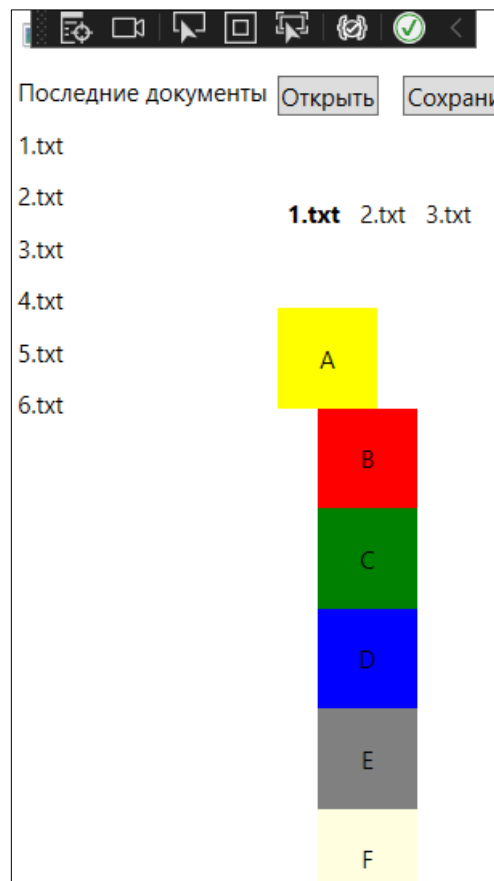


Рисунок 30 — Проверка WrapPanel

## **ВЫВОД**

Вывод: мы узнали, что такое диспетчер компоновки и его виды, разобрались в их отличиях и научились создавать приложения с помощью диспетчеров.