

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

**ЛАБОРАТОРНАЯ РАБОТА №1
ВИЗУАЛЬНОЕ ПРОЕКТИРОВАНИЕ В C#**

СОДЕРЖАНИЕ

Лабораторная работа №1	1
1 Настройка окна приложения	3
2 Размещение элементов интерфейса	3
3 Размещение строки ввода (TextBox)	4
4 Размещение надписей (TextBlock)	4
5 События и обработчики	4
6 Обработка события нажатия кнопки (Click)	4
7 Обработка события загрузки окна (Opened)	5
8 Запуск и работа программы	5
9 Динамическое изменение свойств	5
10 Индивидуальные задания	5

1 Настройка окна приложения

После создания проекта Avalonia App у вас автоматически появится главное окно — MainWindow.axaml, а также связанный с ним файл кода MainWindow.axaml.cs. Файл .axaml содержит описание пользовательского интерфейса в формате XAML, а файл .axaml.cs — код на C#, в котором можно обрабатывать события элементов интерфейса.

Изменять свойства окна можно двумя способами:

- через XAML (описание разметки);
- через код C# (в классе MainWindow).

Пример изменения заголовка и размеров окна:

```
<Window xmlns="https://github.com/avaloniaui"
        Title="Лаба №1. ФИО, группа"
        Width="400" Height="300">
</Window>
```

Цвет фона можно задать свойством Background:

```
<Window Background="AntiqueWhite">
```

2 Размещение элементов интерфейса

Элементы управления (кнопки, поля ввода, надписи и т.д.) размещаются внутри контейнеров разметки Avalonia. Наиболее распространённые контейнеры:

- StackPanel — размещает элементы вертикально или горизонтально;
- Grid — сетка с рядами и столбцами;
- Canvas — позволяет позиционировать элементы с помощью координат.

Пример:

```
<Window xmlns="https://github.com/avaloniaui"
        Title="Лаб. раб. №1. Ст. гр. 311 ФИО"
        Width="400" Height="300">
  <StackPanel Margin="10">
    <TextBlock Text="Введите значение X:"/>
    <TextBox Name="textBox1"/>
    <Button Content="Выполнить" Click="OnButtonClick"/>
  </StackPanel>
</Window>
```

3 Размещение строки ввода (TextBox)

Для ввода и вывода текстовых или числовых данных используется элемент TextBox.

Пример размещения трёх полей ввода:

```
<StackPanel Margin="10">
    <TextBox Name="textBox1"/>
    <TextBox Name="textBox2"/>
    <TextBox Name="textBox3"/>
</StackPanel>
```

Получить введенный текст можно в коде:

```
string s = textBox1.Text;
```

Задать шрифт можно в XAML:

```
<TextBox FontSize="16" FontFamily="Arial"/>
```

4 Размещение надписей (TextBlock)

В Avalonia вместо Label чаще используется TextBlock — элемент, который отображает текстовые надписи.

Пример:

```
<TextBlock Text="Введите значение X:" FontSize="16"/>
```

Текст можно менять динамически из кода:

```
textBlock1.Text = "Результат вычислений:";
```

5 События и обработчики

Каждый элемент интерфейса может вызывать события (например, Click у кнопки). Чтобы обработать событие, в XAML указывается имя метода, а сам метод реализуется в файле .axaml.cs.

6 Обработка события нажатия кнопки (Click)

Пример:

```
<Button Name="button1" Content="Выполнить" Click="OnButtonClick"/>
```

```
private void OnButtonClick(object? sender, RoutedEventArgs e)
{
    MessageBox.Avalonia.MessageBoxManager
        .GetMessageBoxStandardWindow("Сообщение", "Привет!")
}
```

```
        .Show();  
    }
```

(Для вывода сообщений можно использовать пакет `MessageBox.Avalonia`.)

7 Обработка события загрузки окна (Opened)

Событие, аналогичное `Load` в WinForms, — это `Opened` в Avalonia.

```
<Window xmlns="https://github.com/avaloniaui"  
        Title="..."  
        Opened="OnWindowOpened">
```

```
private void OnWindowOpened(object? sender, EventArgs e)  
{  
    this.Background = Brushes.AntiqueWhite;  
}
```

8 Запуск и работа программы

Программу можно запускать из любой IDE или терминала командой:

```
dotnet run
```

После успешной сборки появится окно приложения Avalonia. Для завершения программы просто закройте окно.

9 Динамическое изменение свойств

Свойства элементов можно менять в коде во время выполнения:

```
textBlock1.Text = "Привет!";  
this.Background = Brushes.LightBlue;
```

10 Индивидуальные задания