

## Практическая работа №4

### 1 Цель работы

1.1 Изучить основы стилизации интерфейса Avalonia UI.

### 2 Литература

2.1 Avalonia documentation – Текст : электронный // AvaloniaUI, 2024. – URL: <https://docs.avaloniaui.net/>

### 3 Подготовка к работе

3.1 Повторить теоретический материал (см.п.2).

3.2 Изучить описание лабораторной работы.

### 4 Основное оборудование

4.1 Персональный компьютер.

### 5 Задание

5.1 Применение стилей в Avalonia

5.1.1 Создайте новый проект Avalonia

5.1.2 В UserControl.Styles создайте новый стиль. Используйте селектор «Button» для задания общего стиля всем кнопкам. Измените цвет фона и шрифт. Протестируйте стиль.

5.1.3 Добавьте стиль для TextBox, который применяется только к тем, у кого установлен атрибут Name="Title" (используйте селектор «TextBox#Title»). Протестируйте стиль.

5.1.4 Создайте стиль, который применяется только к кнопкам внутри StackPanel. Измените цвет фона и шрифт. Используйте селектор «StackPanel Button». Протестируйте стиль.

5.1.5 Создайте стиль для TextBlock, который применяет курсивное начертание, если его родитель Border содержит класс highlight (.highlight TextBlock). Протестируйте стиль.

5.2 Применение псевдоклассов

Так как стили большинства псевдоклассов определены в Fluent-теме Avalonia, а элементы состоят из других элементарных элементов, для переопределения свойств, используемых элементов требуется применять селекторы шаблона, например для изменения цвета границы у disabled TextBox необходимо использовать селектор "TextBox:disabled /template/ Border#PART\_BorderElement". Необходимые свойства можно узнать, изучив стандартные стили и шаблоны элементов управления Fluent:

<https://github.com/AvaloniaUI/Avalonia/tree/master/src/Avalonia.Themes.Fluent/Controls>

5.2.1 Создайте стиль для кнопки, который изменяет её цвет при наведении (:pointerover).

5.2.2 Добавьте стиль для TextBox, который меняет цвет границы при фокусировке (:focus).

5.2.3 Создайте стиль для ListBoxItem, который меняет фон при выборе (:selected).

5.2.4 Создайте стиль для ListBoxItem, который определяет цвет фона для четных элементов (:nth-child(even)).

5.3 Привязка к стилям

5.3.1 Создайте 2 стиля для TextBlock, установите в них различный цвет текста.

5.3.2 Создайте класс MoneyOperation со следующими свойствами описание, сумма, является ли операция доходом или расходом.

5.3.3 Отобразите на странице список MoneyOperation, в шаблоне ItemTemplate которого определите цвет метки с суммой при помощи стиля на основании привязки к свойству логического типа. Пример привязки:

```
<TextBlock
    Classes.class1="{Binding IsClass1}"
    Classes.class2="{Binding !IsClass1}"
    Text="{Binding Title}"/>
```

5.4 Стилизация элементов дочерних классов

5.4.1 Создайте стиль, определяющий фон для всех элементов-наследников Button (:is(Button))

5.5 Стилизация на основе значений

При помощи специализированных селекторов можно создавать стили, применяющиеся на основании значений свойств элементов, например «Button[IsCancel=true]» или «TextBlock[(Grid.Row)=0]»

5.5.1 Создайте стиль, который делает TextBlock в первой строке Grid полужирными.

5.5.2 Создайте стиль, который меняет цвет текста TextBlock в первой колонке Grid.

## 6 Порядок выполнения работы

6.1 Выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

## 7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Вывод