

Практическая работа №21

Навигация по страницам

1 Цель работы

1.1 Научиться выполнять навигацию по страницам приложения 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 ViewLocator

5.1.1 Создайте в корне проекта класс со следующим содержимым:

```
public class ViewLocator : IDataTemplate
{
    public bool SupportsRecycling => false;
    public Control Build(object data)
    {
        var name =
            data.GetType().FullName
                .Replace("ViewModel", "View");
        var type = Type.GetType(name);
        if (type != null)
        {
            return (Control)Activator
                .CreateInstance(type);
        }
        else
        {
            return new TextBlock
                { Text = "Not Found: " + name };
        }
    }
    public bool Match(object data)
    {
        return data is ViewModelBase;
    }
}
```

```
}
```

5.1.2 Добавить в App.axaml следующие строчки:

```
<Application
  xmlns:local="using:ИмяПроекта"
  ...
/>

<Application.DataTemplates>
  <local:ViewLocator/>
</Application.DataTemplates>
```

ViewLocator позволяет автоматически создавать объект View для объекта ViewModel в приложении

5.2 Переходы между страницами

5.2.1 В MainViewModel создайте ObservableProperty поле currentPage типа ViewModelBase

5.2.2 Создайте 2 страницы RegistrationView и AuthorizationView, расположите на них соответствующие элементы управления

5.2.3 В MainWindow создайте элемент управления ContentControl и выполните привязку его свойства Content к значению CurrentPage

5.2.4 Создайте в MainWindow две кнопки, создайте и привяжите к ним команды которые будут присваивать CurrentPage соответствующее значение (RegistrationViewModel или AuthorizationViewModel)

5.2.5 Протестируйте преходы между страницами

5.3 Сервис навигации

5.3.1 Создайте новый класс NavigationService

5.3.2 В классе определите навигационный стек в виде Stack<ViewModelBase> и 2 метода для управления этим стеком: универсальный NavigateTo<T>, где T – класс-наследник ViewModelBase и GoBack

```
public void NavigateTo<T>(T viewModel, Action<T>?
action = null) where T : ViewModelBase
{
    //TODO
}

public void GoBack(int steps = 1)
{
    //TODO
}
```

5.3.3 При выполнении навигации в метод может передаваться делегат Action<T>, который позволит произвести настройки для страницы (установить значения некоторых свойств). Для вызова делегата необходимо выполнить

```
action?.Invoke(viewModel);
```

5.3.4 Добавьте экземпляр сервиса в качестве статического публичного члена в MainViewModel

5.3.5 Измените реализацию навигации в приложении с использованием созданного сервиса.

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 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Для чего применяется ViewLocator?

8.2 Как реализовать переход между страницами, обратный переход?

8.3 Как передать данные между страницами?