# Практическая работа №21 Навигация по страницам

1. Цель работы
   1. Научиться выполнять навигацию по страницам приложения Avalonia UI.
2. Литература
   1. Avalonia documentation – Текст : электронный // AvaloniaUI, 2024. – URL: https://docs.avaloniaui.net/
3. Подготовка к работе
   1. Повторить теоретический материал (см.п.2).
   2. Изучить описание лабораторной работы.
4. Основное оборудование
   1. Персональный компьютер.
5. Задание
   1. ViewLocator
      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;

}

}

* + 1. 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

* 1. Переходы между страницами
     1. В MainViewModel создайте ObservableProperty поле сurrentPage типа ViewModelBase
     2. Создайте 2 страницы RegistrationView и AuthorizationView, расположите на них соответствующие элементы управления
     3. В MainWindow создайте элемент управления ContentControl и выполните привязку его свойства Content к значению CurrentPage
     4. Создайте в MainWindow две кнопки, создайте и привяжите к ним команды которые будут присваивать CurrentPage соответствующее значение (RegistrationViewModel или AuthorizationViewModel)
     5. Протестируйте преходы между страницами
  2. Сервис навигиции
     1. Создайте новый класс NavigationService
     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

}

* + 1. 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);

* + 1. Добавьте экземпляр сервиса в качестве статического публичного члена в MainViewModel
    2. Измените реализацию навигации в приложении с использованием созданного сервиса.

1. Порядок выполнения работы
   1. Выполнить все задания из п.5.
   2. Ответить на контрольные вопросы.
2. Содержание отчета
   1. Титульный лист
   2. Цель работы
   3. Ответы на контрольные вопросы
   4. Вывод
3. Контрольные вопросы
   1. Для чего применяется ViewLocator?
   2. Как реализовать переход между страницами, обратный переход?
   3. Как передать данные между страницами?