

[Początek](#) (podstawowe informacje, przygotowanie środowiska Visual Studio 2017 + Xamarin)

1.Wyświetlenie tekstu na nowej własnej stronie

2.Dodanie przycisku i zadeklarowanie działania

3.Wygląd UI

4.Dodanie przycisku nawigacji do następnej strony

5.Przekazanie informacji do następnej strony

6.Wyświetlanie stron jako "Tabbed" i "Carousel"

7.Widok - Lista

Aplikacje

[Twórcy](#)

Początek

// Z początku pojawi się około 31 błędów - są ponieważ nic nie jest zadeklarowane w aplikacji portable

// NIE "NAPRAWIAJ BŁĘDÓW" - zniszczysz aplikację i nie będzie ona cross platformowa

// Prawy przycisk myszy na Rozwiązanie "nazwa twojej aplikacji" -> zarządzaj pakietami nuget
rozwiązania ->

// zainstaluj wszystkie aktualizacje z źródła M.V.S. Offline.. i aktualizację xamarin z nuget

// po pierwsze upewnij się o działanie emulatora androida (intel atom x86 - 10 x szybciej działa od arm)

// jeżeli nie - uruchom CMD jako administrator, wpisz "bcdedit /set hypervisorlaunchtype off" i
uruchom komputer ponownie

// Jeżeli nie działa UWP na maszynie „lokalnej” - naciśnij prawym przyciskiem myszy na „nazwa twojej
apki”.UWP -> wdróż projekt -> uruchom na maszynie lok (ctrl + F5)

// aby uruchomić projekt w kilku środowiskach - Prawy przycisk myszy na Rozwiązanie "nazwa twojej
aplikacji" -> ustaw projekty startowe -> w zakładce wiele projektów startowych ... "uruchom"...

JEŻELI WSZYSTKO POPRAWNIE ZROBIŁEŚ PO URUCHOMIENIU APLIKACJI W TRYBIE DEBUG POJAWI SIĘ
NA EMULATORZE NAPIS Welcome to Xamarin Forms

/// inne podstawowe informacje

Aplikacje Cross-Platformowe Visual Studio 2017 + Xamarin

XAML - Extensible Application Markup Language

Komentarz w języku C

// tekst w c

Komentarz w języku XAML

<!-- tekst w xaml ... -->

Deklaracja tekstu XAML

```
<Label Text="wpisz swój text"
      HorizontalOptions="Center"
      VerticalOptions="Center"
/>
```

HorizontalOptions = " " - opcje poziome

VerticalOptions = " " - opcje pionowe

1. Wyświetlenie tekstu na nowej własnej stronie

„Pusta aplikacja” Nazwa aplikacji AppXross1

W Eksploratorze Rozwiązań naciśnij prawym przyciskiem myszy na AppXross1 -> Dodaj -> nowy element -> Cross-Platform -> Forms Blank Content Page Xaml i nazwij go „Domowa”

App.xaml.cs wpisz

```
public App ()
{
    MainPage = new Domowa(); //Zadeklarowanie naszej strony jako domowej

    //InitializeComponent();           pominelem

    //MainPage = new AppXross1.MainPage();   pominelem
}
```

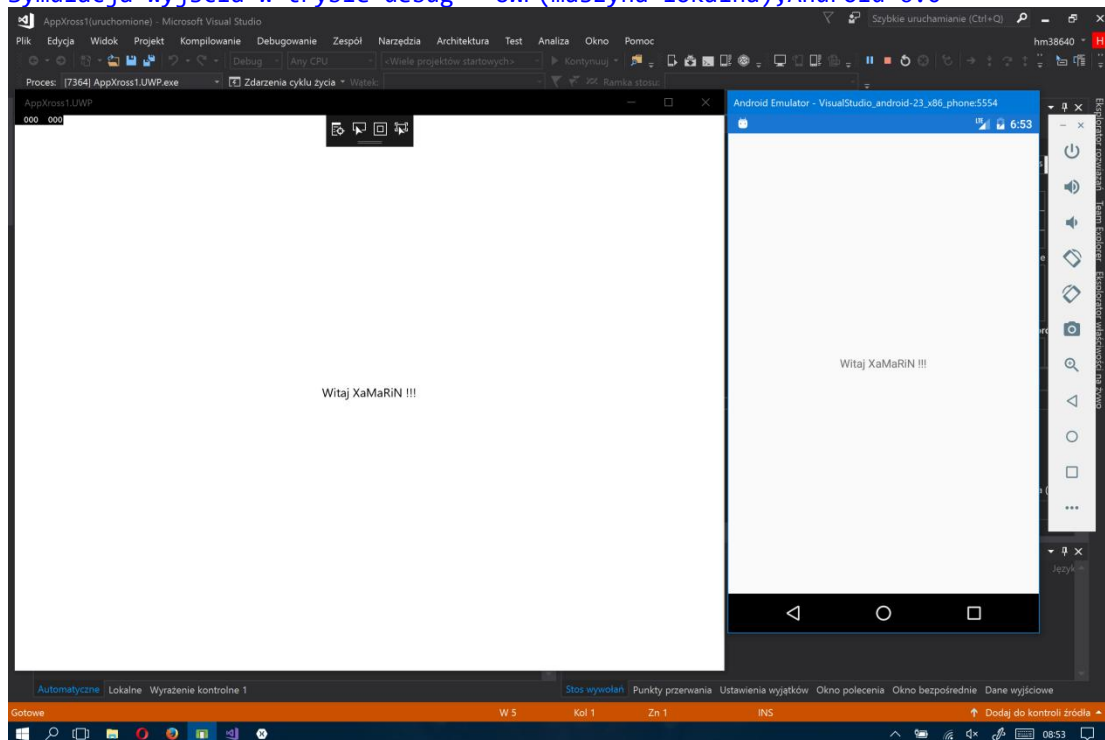
Domowa.xaml wpisz :

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
              xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
              x:Class="AppXross1.Domowa">

    <Label Text="Witaj XaMaRiN !!!" VerticalOptions="Center"
HorizontalOptions="Center" /> <!--Zadeklarowany text w xaml-->

</ContentPage>
```

Symulacja wyjścia w trybie debug - UWP(maszyna lokalna),Android 6.0



2.Dodanie przycisku i zadeklarowanie działania

„Pusta aplikacja” Nazwa aplikacji AppXross1

W Eksploratorze Rozwiązań naciśnij prawym przyciskiem myszy na AppXross1 -> Dodaj -> nowy element -> Cross-Platform -> Forms Blank Content Page Xaml i nazwij go „Domowa”

App.xaml.cs wpisz

```
public App ()
{
    MainPage = new Domowa(); //Zadeklarowanie naszej strony jako domowej

    //InitializeComponent();          pominielem

    //MainPage = new AppXross1.MainPage();    pominielem
}
```

Domowa.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
              xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
              x:Class="AppXross1.Domowa">

    <StackLayout>
        <!-- powłoka w ktorej bdzie realizowane to co jest wewnatrz -->

        <Label Text="Podaj swoje imię"/>
        <!-- Deklaruje widoczny tekst na samej górze aplikacji-->

        <Entry x:Name="GlownyWpis"/>
        <!-- Nasze pole tekstowe w ktorym napiszemy swoje imie -->

        <Button Clicked="Button_Clicked"
                Text="Przywitaj się!"/>
        <!-- przycisk w xaml-->
        <!-- Po wpisaniu Clicked=" <nowa procedura obsługi zdarzeń>"
        zostanie utworzona w Domowa.xaml.cs funkcja określająca działanie przycisku
        -->

        <Label x:Name="GlownyLabel"/>
        <!-- będzie realizować dane zadanie -> patrz Domowa.xaml.cs
        (możesz też kliknąć prawym przyciskiem myszy -> przejdź do definicji) -->

    </StackLayout>

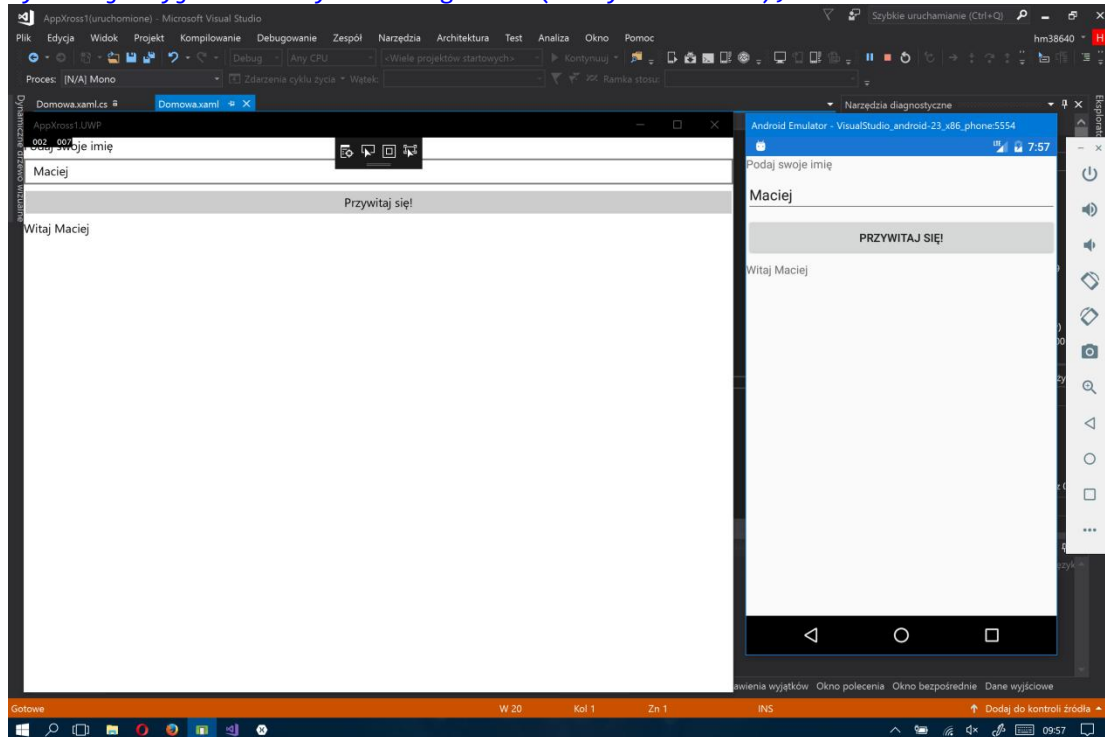
</ContentPage>
```

Domowa.xaml.cs

```
private void Button_Clicked(object sender, EventArgs e)
{
    string text = GlownyWpis.Text;
    //odniesienie do entry x.Name
}
```

```
    GlownyLabel.Text = "Witaj " + text;  
    //odniesienie do Label x:Name="GlownyLabel" + powyższy string text  
}
```

Symulacja wyjścia w trybie debug - UWP(maszyna lokalna),Android 6.0



3. Wygląd UI

„Pusta aplikacja” Nazwa aplikacji AppXross1

W Eksploratorze Rozwiązań naciśnij prawym przyciskiem myszy na AppXross1 -> Dodaj -> nowy element -> Cross-Platform -> Forms Blank Content Page Xaml i nazwij go „Domowa”

App.xaml.cs wpisz

```
public App ()
{
    MainPage = new Domowa(); //Zadeklarowanie naszej strony jako domowej

    //InitializeComponent();           pominalem

    //MainPage = new AppXross1.MainPage();   pominalem
}
```

Domowa.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="AppXross1.Domowa"
    BackgroundColor="White">
    <!-- Bez BackgroundColor o takim samym kolorze co stack layout
    pojawiają się paski o innych kolorach (chyba że nie zastosuje się VerticalOptions
    w stacklayout) -->

    <StackLayout BackgroundColor="White"
        Padding="60"
        VerticalOptions="Center">
        <!--Kolor powłoki aplikacji-->
        <!--Padding powoduje, że przyciski i tekst oddalą się od krawędzi ekranu w
        ""-->
        <!--VerticalOptions - centruje powłokę aplikacji (Bez BackgroundColor będą
        paki) -->

        <!-- powłoka w ktorej bdzie realizowane to co jest wewnatrz -->

        <Label Text="Podaj swoje imię"/>
        <!-- Deklaruje widoczny tekst na samej górze aplikacji-->

        <Grid Padding="0,0,0,50">
            <!-- Padding nadaje przerwę od głównego wpisu "0,0,0,50" lewo, góra, prawo,
            dół-->

            <Entry x:Name="GlownyWpis"
                BackgroundColor="Gray"/>
            <!--Kolor wpisu-->

        </Grid>

        <!-- Nasze pole tekstowe w którym napiszemy swoje imie -->

        <Button Clicked="Button_Clicked"
            Text="Przywitaj się!"
            BackgroundColor="Green"
            TextColor="White"/>
    </StackLayout>
</ContentPage>
```

```

<!--Kolor przycisku i kolor tekstu-->

<!-- przycisk w xaml-->
<!-- Po wpisaniu Clicked=" <nowa procedura obsługi zdarzeń>"
zostanie utworzona w Domowa.xaml.cs funkcja określająca działanie przycisku
-->

<Label x:Name="GlownyLabel"
        TextColor="Gray"
        FontSize="40"/>
<!--Kolor tekstu i rozmiar czcionki w "" -->

<!-- będzie realizować dane zadanie -> patrz Domowa.xaml.cs
(mozesz też kliknąć prawym przyciskiem myszy -> przejdź do definicji) -->

</StackLayout>

</ContentPage>

```

Domowa.xaml.cs

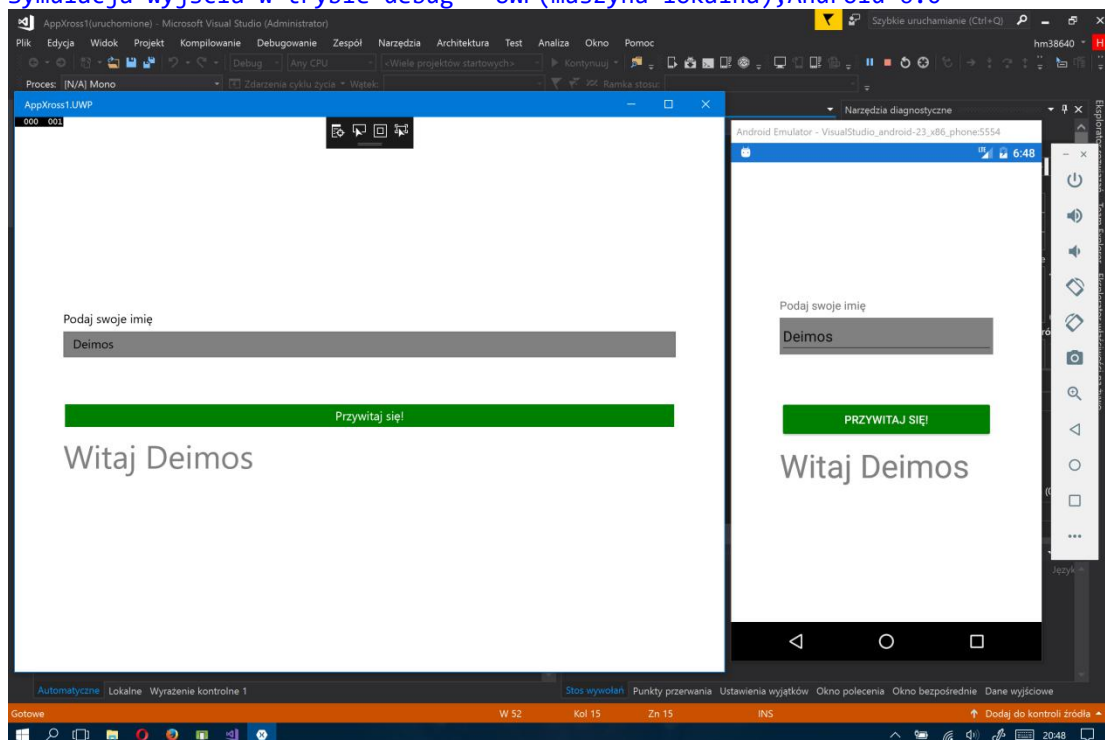
```

private void Button_Clicked(object sender, EventArgs e)
{
    string text = GlownyWpis.Text;
    //odniesienie do entry x.Name

    GlownyLabel.Text = "Witaj " + text;
    //odniesienie do Label x:Name="GlownyLabel" + powyższy string text
}

```

Symulacja wyjścia w trybie debug - UWP(maszyna lokalna),Android 6.0



4. Dodanie przycisku nawigacji do następnej strony

„Pusta aplikacja” Nazwa aplikacji AppXross1

W Eksploratorze Rozwiązań naciśnij prawym przyciskiem myszy na AppXross1 -> Dodaj -> nowy element -> Cross-Platform -> Forms Blank Content Page Xaml i nazwij go „Domowa”

W Eksploratorze Rozwiązań naciśnij prawym przyciskiem myszy na AppXross1 -> Dodaj -> nowy element -> Cross-Platform -> Forms Blank Content Page Xaml i nazwij go „Page1”

Page1.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
              xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
              x:Class="AppXross1.Page1"

              Title="Druga strona">
    <!--Nadaliśmy tytuł naszej stronki -->

</ContentPage>
```

Domowa.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
              xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
              x:Class="AppXross1.Domowa"
              BackgroundColor="White">

    <StackLayout BackgroundColor="White"
                  Padding="60"
                  VerticalOptions="Center">

        <Label Text="Podaj swoje imię"/>

        <Grid Padding="0,0,0,50">

            <Entry x:Name="GlownyWpis"
                    BackgroundColor="Gray"/>

        </Grid>

        <Button Clicked="Button_Clicked"
                Text="Przywitaj się!"
                BackgroundColor="Green"
                TextColor="White"/>

        <Label x:Name="GlownyLabel"
                TextColor="Gray"
                FontSize="40"/>

        <Button Text="Przejdź do drugiej strony"
                Clicked="Button_Clicked_1"
                BackgroundColor="Lavender">
            <!--tekst na przycisku + nowa definicja zdarzenia Clicked tj
            Button_Clicked_1 -->
```



```
</Button>

</StackLayout>

</ContentPage>
```

Domowa.xaml.cs

```
private void Button_Clicked(object sender, EventArgs e)
{
    string text = GlownyWpis.Text;
    //odniesienie do entry x.Name

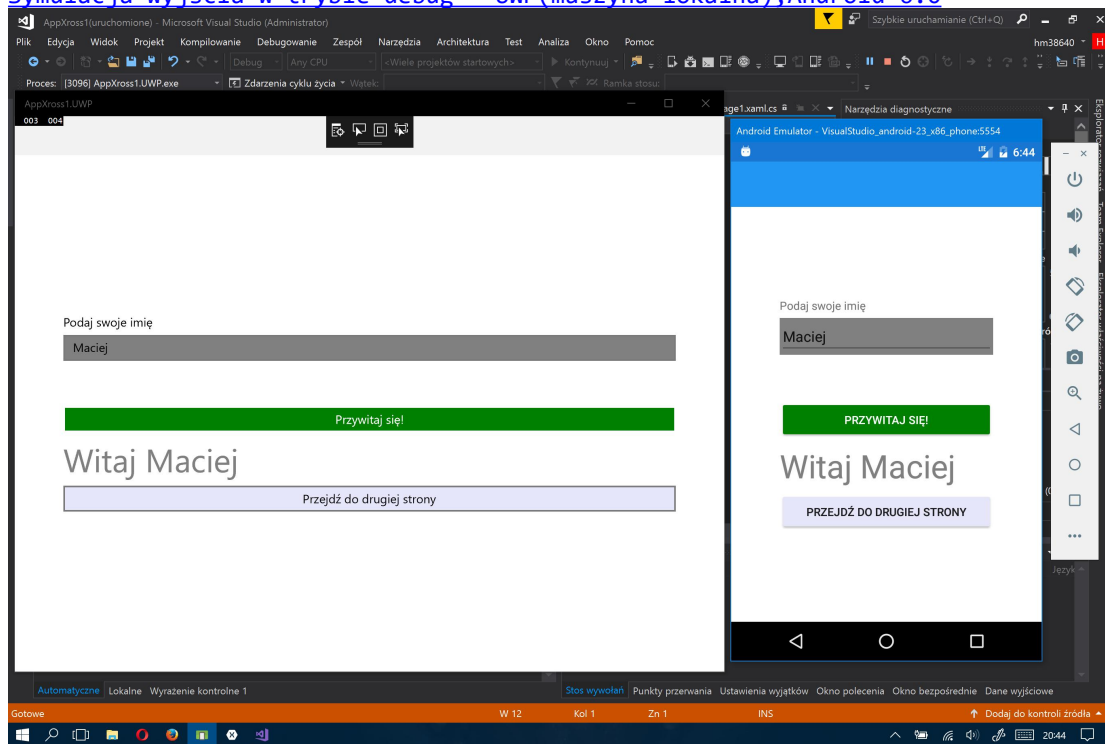
    GlownyLabel.Text = "Witaj " + text;
    //odniesienie do Label x:Name="GlownyLabel" + powyższy string text
}

private async void Button_Clicked_1(object sender, EventArgs e) //musi być
asynchroniczne z powodu metody await
{
    await Navigation.PushAsync(new Page1()); //działanie naciśnięcia
przycisku -> przekierowuje do Page1
}
```

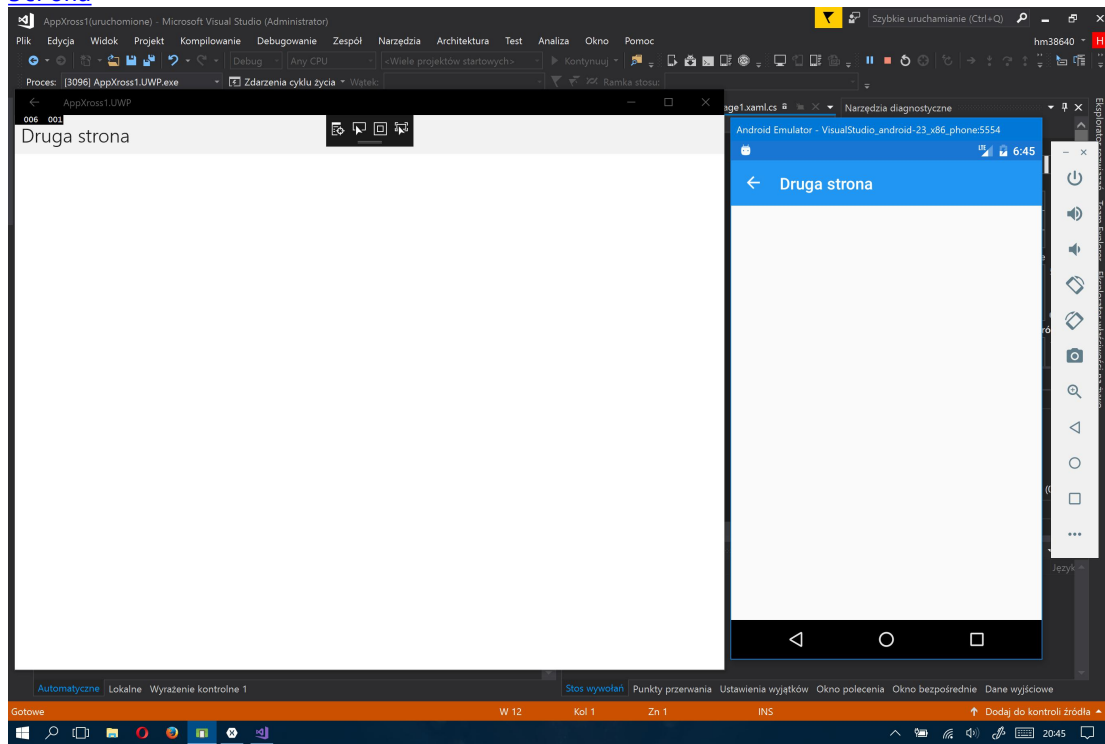
App.xaml.cs

```
public App ()
{
    MainPage = new NavigationPage(new Domowa()); //Zadeklarowanie naszej
strony jako domowej
}
```

[Symulacja wyjścia w trybie debug - UWP\(maszyna lokalna\),Android 6.0](#)



druga strona



5.Przekazanie informacji do następnej strony

„Pusta aplikacja” Nazwa aplikacji AppXross1

W Eksploratorze Rozwiązań naciśnij prawym przyciskiem myszy na AppXross1 -> Dodaj -> nowy element -> Cross-Platform -> Forms Blank Content Page Xaml i nazwij go „Domowa”

W Eksploratorze Rozwiązań naciśnij prawym przyciskiem myszy na AppXross1 -> Dodaj -> nowy element -> Cross-Platform -> Forms Blank Content Page Xaml i nazwij go „Page1”

Domowa.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
              xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
              x:Class="AppXross1.Domowa"
              BackgroundColor="White">

    <StackLayout BackgroundColor="White"
                  Padding="60"
                  VerticalOptions="Center">

        <Label Text="Podaj swoje imię"/>

        <Grid Padding="0,0,0,50">

            <Entry x:Name="GlownyWpis"
                    BackgroundColor="Gray"/>
            <!--Parametr z GlownyWpis (to co wpisujemy w szarym obszarze)
            zostanie przekazany na następną stronę -> patrz Domowa.xaml.cs -->

        </Grid>

        <Button Clicked="Button_Clicked"
                Text="Przywitaj się!"
                BackgroundColor="Green"
                TextColor="White"/>

        <Label x:Name="GlownyLabel"
                TextColor="Gray"
                FontSize="40"/>

        <Button Text="Przejdź do drugiej strony"
                Clicked="Button_Clicked_1"
                BackgroundColor="Lavender">
            <!--tekst na przycisku + nowa definicja zdarzenia Clicked tj
            Button_Clicked_1 -->
        </Button>

    </StackLayout>

</ContentPage>
```

Domowa.xaml.cs

```
private void Button_Clicked(object sender, EventArgs e)
{
    string text = GlownyWpis.Text;
```

```

        //odniesienie do entry x.Name

        GlownyLabel.Text = "Witaj " + text;
        //odniesienie do Label x:Name="GlownyLabel" + powyższy string text
    }

    private async void Button_Clicked_1(object sender, EventArgs e) //musi być
asynchroniczne z powodu metody await
    {
        await Navigation.PushAsync(new Page1(GlownyWpis.Text));
        //działanie naciśnięcia przycisku -> przekierowuje do Page1
        //wpisany tekst z GlownyWpis (patrz -> domowa.xaml) zostanie przekazany
do Page1
    }

```

App.xaml.cs

```

public App ()
{
    MainPage = new NavigationPage(new Domowa());
    //Zadeklarowanie naszej strony jako domowej
    //Strona Nawigacyjna jest zadeklarowana jako podstrona domowej
}

```

Page1.xaml

```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="AppXross1.Page1"

    Title="Druga strona">
    <!--Nadaliśmy tytuł naszej stronki -->

    <Label x:Name="DrogiLabel"
        TextColor="Gray"/>
    <!--Label wyświetla text w Xamarin.Forms-->
    <!--Zadeklarujemy go jako parametr w Page1.xaml.cs-->

</ContentPage>

```

Page1.xaml.cs

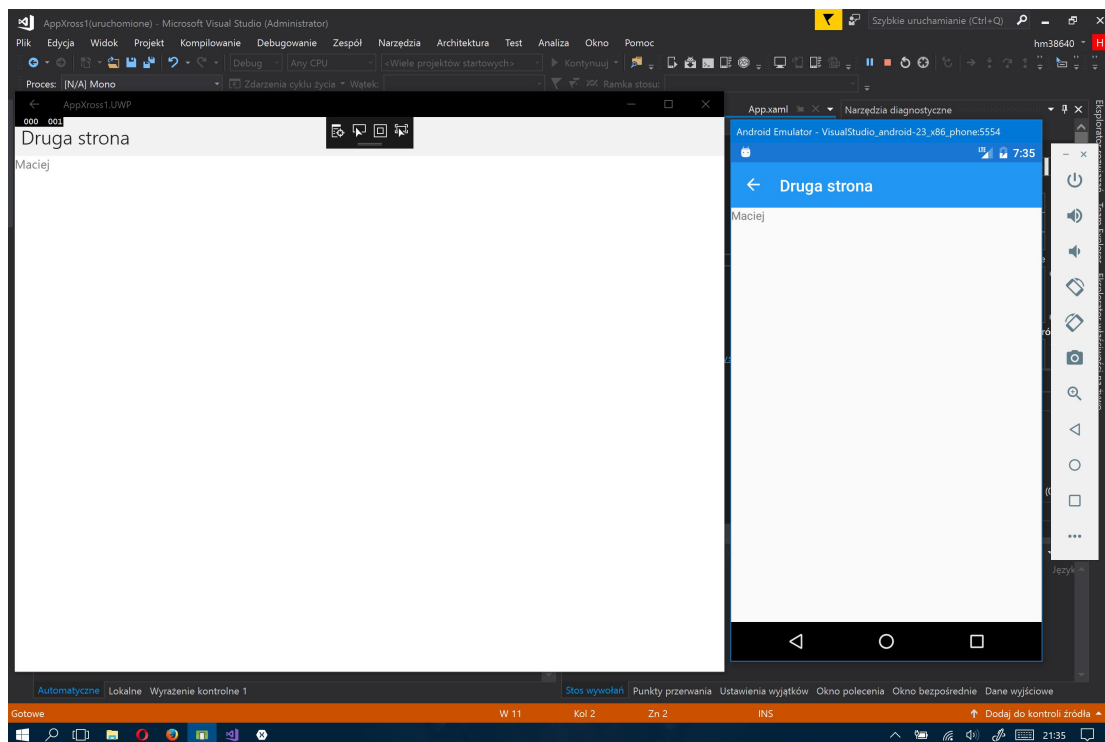
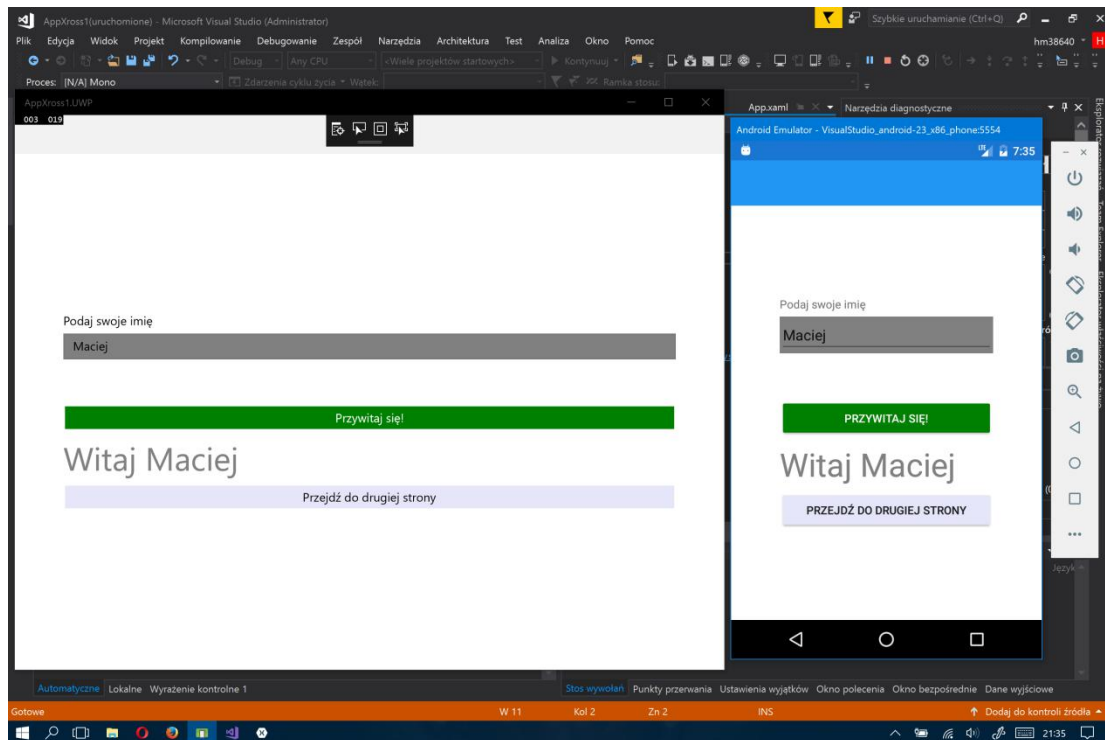
```

    public Page1 (string parameter)
        //A string is a sequential collection of Unicode characters that is used to
        represent text.
        //Więcej informacji na temat stringa jako klasy i konstruktora na
https://developer.xamarin.com/api/type/System.String/

    {
        InitializeComponent ();

        DrogiLabel.Text = parameter; //odnośnik od Label w Page1.xaml jako
parametru
    }

```



6. Wyświetlanie stron jako "Tabbed" i "Carousel"

-> Aby wyświetlać strony w trybie „Tabbed” należy w aplikacji z pkt 5 zmienić

App.xaml.cs

```
public App ()
{
    MainPage = new TabbedPage //zadeklarowanie strony "tabowanej"
    {
        Children = //zadeklarowanie stron jako zakładki
        {
            new Domowa(),
            new Page1("Witaj !!!")
        }
    };
}
```

Domowa.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="AppXross1.Domowa"
    BackgroundColor="White"
    Title="Strona Domowa"> <!--Zostało dodane -->
```

-> Aby wyświetlać strony w trybie „Carousel” należy w aplikacji z pkt 5 zmienić

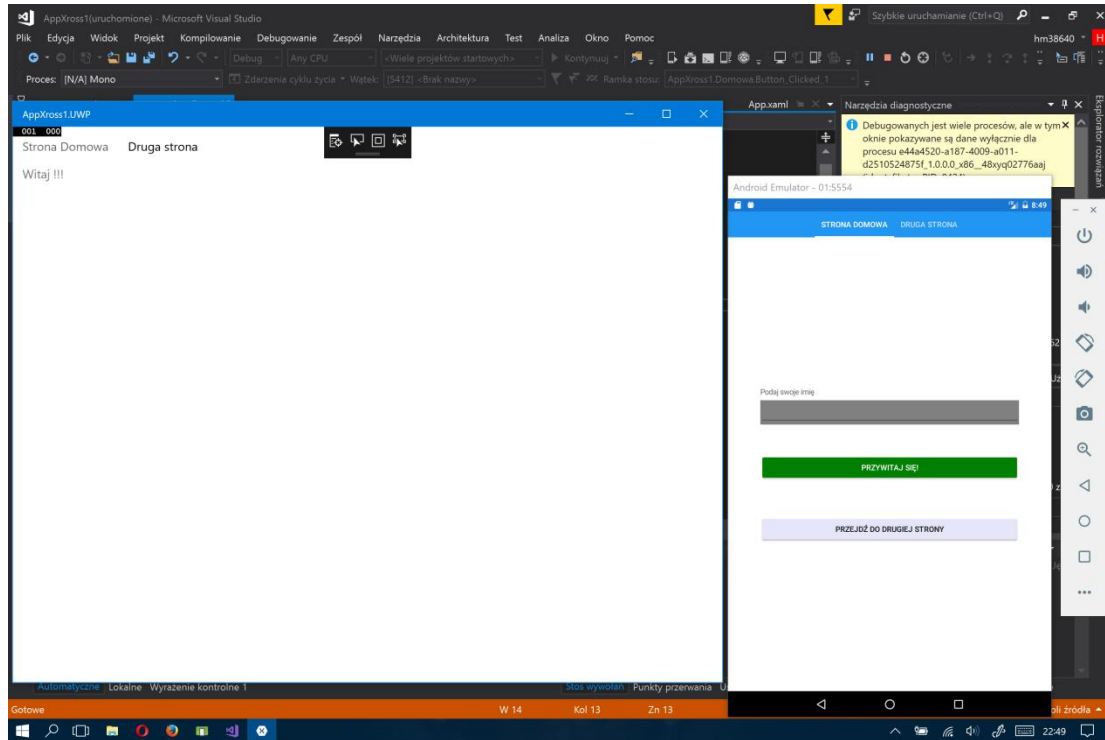
App.xaml.cs

```
public App ()
{
    MainPage = new CarouselPage//zadeklarowanie strony "Carousel"
    {
        Children = //zadeklarowanie stron jako zakładki
        {
            new Domowa(),
            new Page1("Witaj !!!")
        }
    };
}
```

Domowa.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="AppXross1.Domowa"
    BackgroundColor="White"
    Title="Strona Domowa"> <!--Zostało dodane -->
```

Strona tabowana



7.Widok - Lista

„Pusta aplikacja” Nazwa aplikacji AppXross7

W Eksploratorze Rozwiązań naciśnij prawym przyciskiem myszy na AppXross1 -> Dodaj -> nowy element -> Cross-Platform -> Forms Blank Content Page Xaml i nazwij go „Domowa”

Domowa.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
              xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
              x:Class="AppXross7.Domowa"

              BackgroundColor="White"
              Title="Strona Domowa">
    <!--Kolor powłoki "BackgroundColor" - biały
    Nazwa storny "Title"-->

    <ListView x:Name="GlownyListView" />
    <!--Dodajemy Widok listowy pod daną nazwą
    i deklarujemy w Domowa.xaml.cs-->

</ContentPage>
```

Domowa.xaml.cs

```
public Domowa ()
{
    InitializeComponent ();

    GlownyListView.ItemsSource = new List<string>
    {
        "Maciej", "Deimos", "Mojżesz", "3perełki ;)", "Jestem", "Byłem",
        "Będę"
    };
    //przypomnienie info o string :
    https://developer.xamarin.com/api/type/System.String/
    //info o itemssource
    https://developer.xamarin.com/api/property/Xamarin.Forms.ItemsView<TVisual>.ItemsSource/
}
```

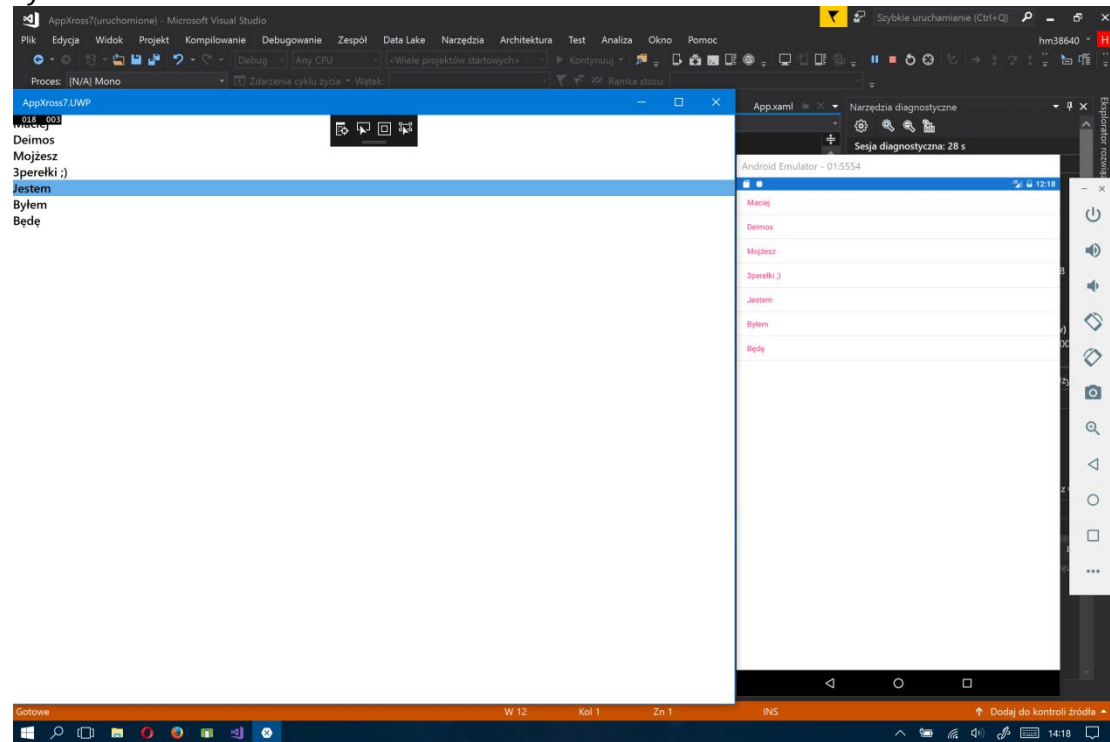
App.xaml.cs

```
public App ()
{
    MainPage = new Domowa();

    //InitializeComponent();           pomiń

    //MainPage = new AppXross7.MainPage(); pomiń
}
```


wynik



Aplikacje

1

2

3

4

5

6

7

Created by:

Maciej Hejlasz

Facebook

