



Lab. 1.6. Aplikacja webowa (ASP .NET Framework)

Tworzenie aplikacji – formularza w trybie Design;
 Toolbox; Web Forms, znaczniki ASP .NET; elementy
 formularza: label, TextBox, Button; atrybuty (ID), Text;
 obsługa zdarzenia Click dla przycisku, Interpolated
 String, uruchomienie IIS, testowanie aplikacji w Edge

C#

16.05.2018

KRZYSZTOF MOLENDĄ - MICROSOFT IMAGINE ACADEMY@WSEI

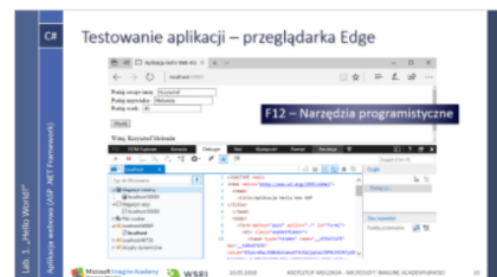
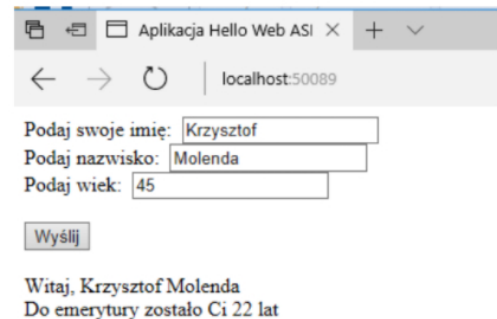
1

Strona 1 / 10

C#

Cel

- › Celem tego ćwiczenia będzie przekształcenie opracowanej w Lab. 1.1 aplikacji konsolowej na aplikację webową, jednostronicową, w technologii ASP.
- › ASP (Active Server Pages) to zbiór technologii Microsoft .NET, przeznaczonych do budowy różnorodnych aplikacji Internetowych.
- › Proces tworzenia aplikacji ASP .NET – w najprostszym przypadku – w zasadzie nie odbiega znacząco od procesu tworzenia aplikacji desktopowych z GUI. Proces kodowania wspomagany jest pracą w trybie Design – również korzystamy z Toolbox'a oraz Properties.
- › Aplikacja ASP .NET składa się ze statycznej strony w HTML, której kod jest uzupełniony dyrektywami i znacznikami ASP, wprowadzającymi do strony „dynamiczność”. Warstwa prezentacji (pliki .aspx) oddzielona jest od logiki aplikacji (pliki .cs) – *code-behind*. Logika aplikacji tworzona jest w C#.
- › Aplikacja ASP .NET uruchamiana jest w środowisku serwera IIS (Internet Information Services). Visual Studio dostarcza wymagane środowiska dla potrzeb tworzenia i testowania aplikacji.



16.05.2018

KRZYSZTOF MOLENDĄ - MICROSOFT IMAGINE ACADEMY@WSEI

2

Strona 2 / 10

C#

Tworzenie nowego projektu

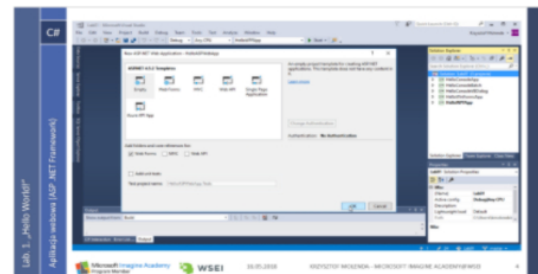
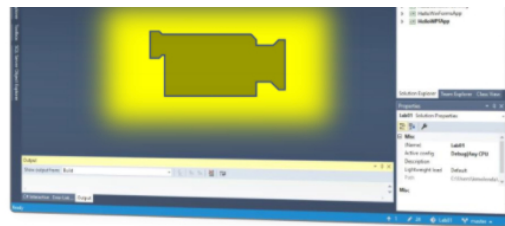
- › Do istniejącej solucji dodaj nowy projekt:
 - W oknie **Solution Explorer** prawokliknij myszką na **Solution 'Lab01'** i z menu kontekstowego wybierz

screencast

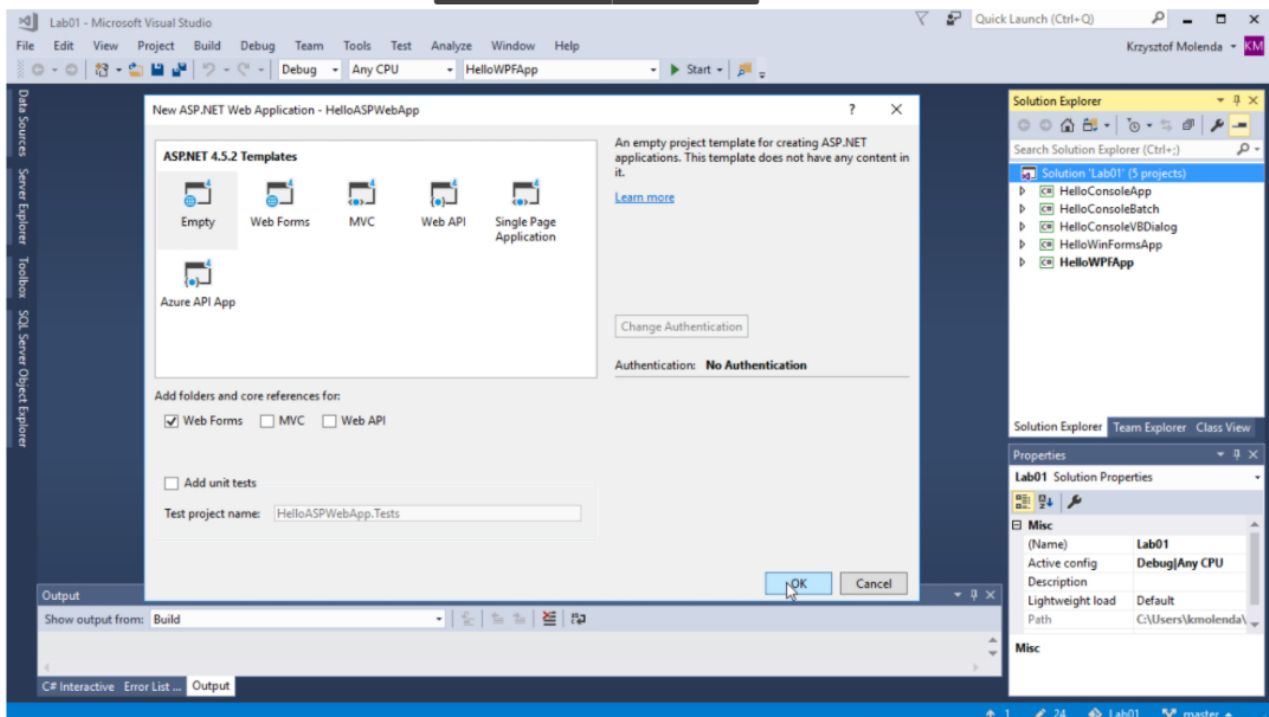


Solution Explorer z menu kontekstowego wybierz **Add → New Project ...**

- Utwórz projekt według szablonu **Visual C# → ASP .NET Web Application (.NET Framework)** nadając mu nazwę **HelloASPWebApp**
- Wybierz szablon pusty (**Empty**) i zaznacz **check box Web Forms**.
- Uczyń ten projekt aktywnym (**StartUp**).
- › Dodaj do projektu nowy formularz web: **Add → Web Form**. Nadaj mu nazwę **Default.aspx**. Kreatory wygeneruje kod dodatkowo kod C#: **Default.aspx.cs** oraz **Default.aspx.designer.cs**.
- › Z plikiem **Default.aspx** możesz pracować w trybie **Design**, **Source** oraz **Split**. Przełącz się na tryb **Design**. Uaktywnij **Toolbox**.



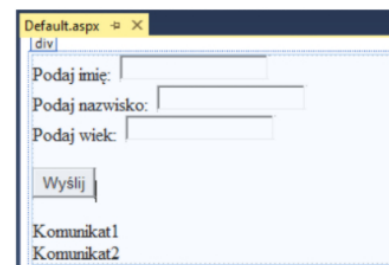
C#



C#

Edycja okna formularza

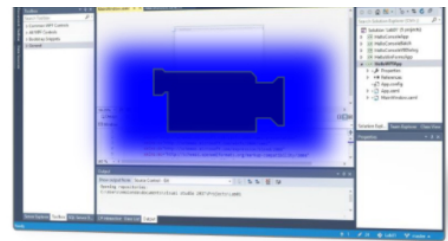
- › Pracując w trybie **Design** w pliku **Default.aspx** zaprojektuj formularz podobny do tego z poprzedniego Lab. W obszarze projektowania:
 - Wpisz tekst **Podaj imię:**, a po nim dwie spacje.
 - Korzystając z **Toolbox**-a dwukrotnie kliknij kontrolkę **TextBox**.
 - Naciśnij **Enter**, aby przejść do nowego wiersza
 - Powtórz kroki dla **nazwiska**, naciśnij **Enter**
 - Powtórz kroki dla **wieku**, naciśnij dwa razy **Enter**
 - Korzystając z **Toolbox**-a wstaw kontrolkę **Button**, naciśnij dwa razy **Enter**
 - Wstaw dwie kontrolki typu **Label** – jedna pod drugą
- › W ASP .NET programowe nazwy kontrolki (zmiennne w C#) to atrybut (**ID**). Treść wyświetlana przez kontrolkę, to atrybut **Text**.



screencast



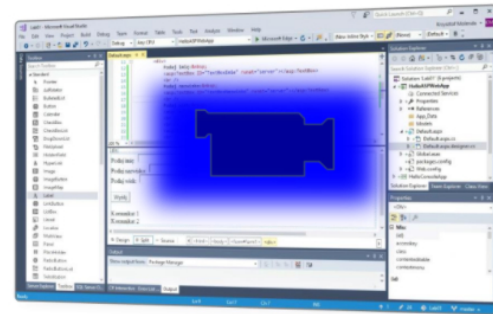
- Nadaj kontrolkom nazwy, odpowiednio: TextBoxImie, TextBoxNazwisko, TextBoxWiek, ButtonOK, LabelKom1, LabelKom2.
- Przejdź do trybu **Source**. Zmiany w wyglądzie aplikacji możesz dokonać w kodzie HTML. Pomiędzy znacznikami HTML `<title></title>` wprowadź tytuł strony **Aplikacja Hello Web .NET**



C#

Programowanie funkcjonalności formularza

- Do aplikacji dodamy funkcjonalność: *po kliknięciu w przycisk*,
 - zamiast **Komunikat 1** pojawi się powitanie: „Witaj, imię *nazwisko*!”
 - zamiast **Komunikat 2** pojawi się tekst dotyczący emerytury.
- Odpowiedni kod realizujący tę funkcjonalność skojarzony będzie z przyciskiem ButtonOK (formalnie – obsługa zdarzenia Click dla tego obiektu) i umieszczony w pliku **Default.aspx.cs**.
 - Dwukrotnie kliknij na kontrolce przycisku na formularzu – wygenerowany zostanie szkielet kodu procedury obsługi zdarzenia.
 - Wprowadź logikę obsługi zdarzenia.
- Komentarz: zwróć uwagę na nową metodę tworzenia napisu: `$"Do {67-wiek} lat"`. Jest to tzw. **Interpolated String** (od C# 6.0). W symbolach zastępczych zawartych w `{ }` możesz odwoływać się do nazw zmiennych będących w zasięgu instrukcji.



screencast



kod C#

C#

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace HelloASPWebApp
{
    public partial class Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

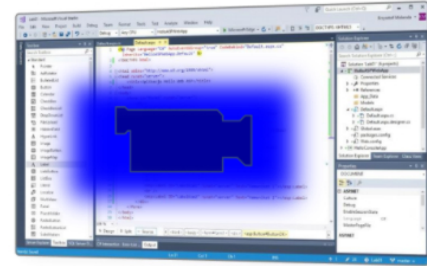
        }

        protected void ButtonOK_Click(object sender, EventArgs e)
        {
            LabelKom1.Text = "Witaj, " + TextBoxImie.Text + " " + TextBoxNazwisko.Text;

            var wiek = Convert.ToInt32(TextBoxWiek.Text);
            if (wiek < 67)
                LabelKom2.Text = $"Do emerytury zostało Ci {67 - wiek} lat";
            else
                LabelKom2.Text = "Jesteś emerytem!";
        }
    }
}
```


Testowanie aplikacji

- › Aplikacje Web testowane są w środowisku Web serwera (IIS – *Internet Information Services*). Visual Studio dostarcza stosownego wsparcia (lokalny serwer IIS).
- › Uruchomienie aplikacji (F5 lub Ctrl-F5) powoduje:
 - Sprawdzenie kodu ASP i skompilowanie kodu C#.
 - Uruchomienie lokalnego serwera IIS na określonym porcie.
 - Uruchomienie strony Default.aspx w środowisku serwera i przeglądarce internetowej.
 - Zakończenie aplikacji – czerwony kwadrat i zamknięcie okna przeglądarki.
- › Nieestetyczny wygląd – po uruchomieniu aplikacji wyświetlane są nic nie mówiące opisy [Komunikat 1](#) oraz [Komunikat 2](#). Z takimi domyślnymi wartościami zainicjowane zostały zmienne labelKom1 oraz labelKom2. Popraw to w kodzie [Default.aspx](#).
- › W przypadku podania nienumerycznej (niecałkowitej) wartości w polu [wiek](#) zgłaszany jest wyjątek i aplikacja przerywa działanie – sprawdź.
- › Drobne poprawki w kodzie [aspx](#) możesz wykonywać bez konieczności ponownego uruchamiania aplikacji (kod jest interpretowany) – wystarczy odświeżenie przeglądarki. „Grubsze” zmiany – w kodzie C# – wymagają jej zamknięcia i ponownego uruchomienia.
- › Udostępnienie aplikacji wymaga odpowiedniego internetowego środowiska uruchomieniowego (Build→Publish...).



screencast



kod aspx

Wyzwanie (dla bardziej zaawansowanych): spróbuj samodzielnie wykonać aplikację ASP opisaną w:
 › [Tutorial MSDN](#) (Build Your First ASP.NET Application with ASP.NET Web Forms)

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="HelloASPWebApp.Default" %>
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>Aplikacja Hello Web ASP</title>
</head>
<body>
<form id="form1" runat="server">
<div>
Podaj imię:&nbsp;
<asp:TextBox ID="TextBoxImie" runat="server"></asp:TextBox>
<br />
Podaj nazwisko:&nbsp;
<asp:TextBox ID="TextBoxNazwisko" runat="server"></asp:TextBox>
<br />
Podaj wiek:&nbsp;
<asp:TextBox ID="TextBoxWiek" runat="server"></asp:TextBox>
<br />
<br />
<asp:Button ID="ButtonOK" runat="server" Text="Wyślij"
OnClick="ButtonOK_Click" />
<br />
<asp:Label ID="LabelKom1" runat="server" Text="Komunikat 1"></asp:Label>
<br />
<asp:Label ID="LabelKom2" runat="server" Text="Komunikat 2"></asp:Label>
</div>
</form>
</body>
</html>
```

Testowanie aplikacji – przeglądarka Edge



Podaj nazwisko: Molenda

Podaj wiek: 45

Wyślij

Witaj, Krzysztof Molenda