



## Lab. 1.3. Aplikacja konsolowa z interakcją w środowisku graficznym (okienka dialogowe)

Dodawanie referencji do bibliotek  
niestandardowych, współpraca z bibliotekami  
VB.NET, okienka dialogowe VB.NET

C#

15.05.2018

KRZYSZTOF MOLENDĄ - MICROSOFT IMAGINE ACADEMY@WSEI

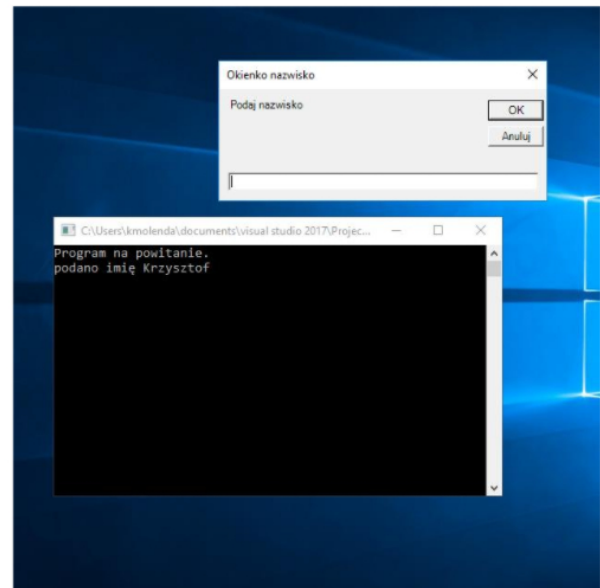
1

Strona 1 / 8

C#

### Cel

- › W ramach tego ćwiczenia zmodyfikujesz opracowaną aplikację w Lab. 1.1 tak, aby komunikacja z użytkownikiem realizowana była za pomocą okien wyskakujących dialogowych w środowisku graficznym.
- › Wykorzystane zostanie API języka Visual Basic: polecenia `InputBox` oraz `MsgBox`. W tym ćwiczeniu zademonstrowana zostanie siła platformy .NET – możliwość łączenia kodu języka C# z kodem (bibliotekami) opracowanym w innych językach programowania wspieranych w ramach platformy .NET.



15.05.2018

KRZYSZTOF MOLENDĄ - MICROSOFT IMAGINE ACADEMY@WSEI

2

Strona 2 / 8

C#

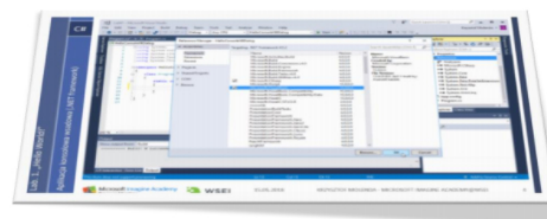
### Tworzenie nowego projektu

- › Do istniejącej solucji dodaj nowy projekt:
  - W oknie `Solution Explorer` prawokliknij myszką na `Solution 'Lab01'` i z menu kontekstowego wybierz `Add → New Project`

screencast



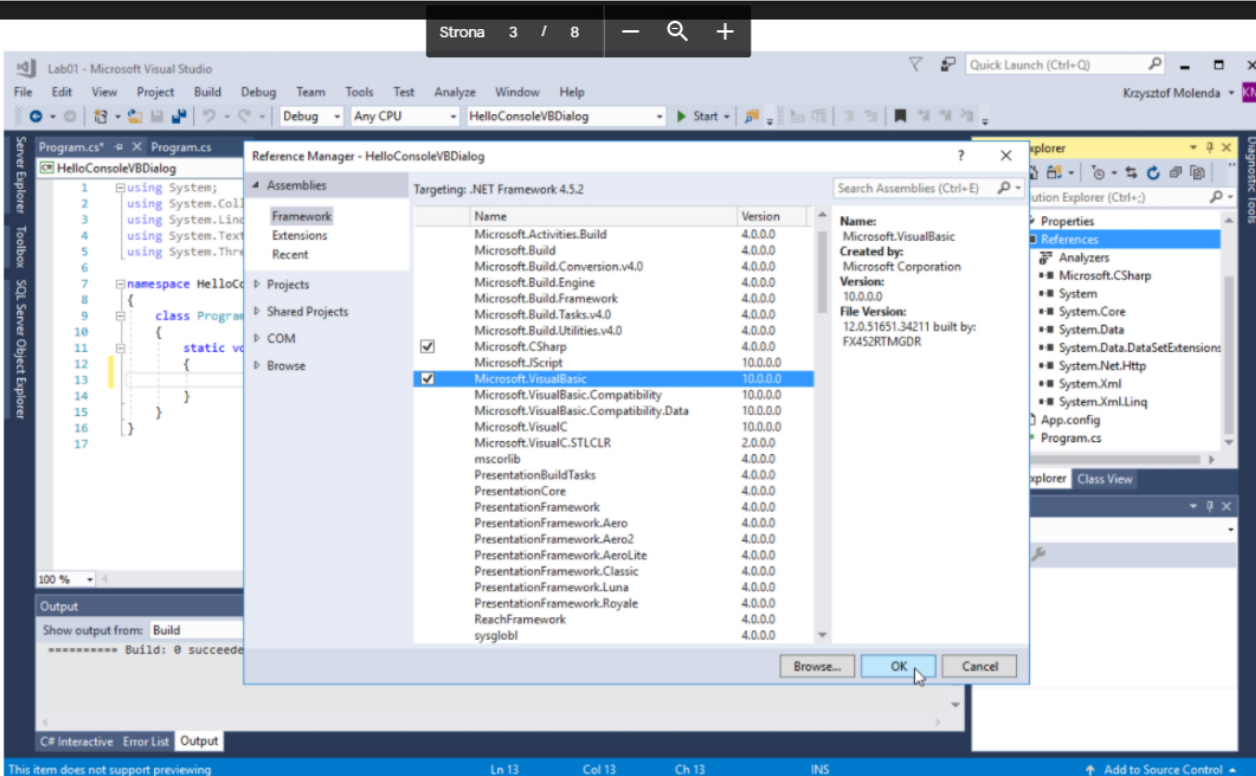
- Utwórz projekt według szablonu **Visual C# → Console App (.NET Framework)** nadając mu nazwę **HelloConsoleVBDiag**
  - (ważne: szablon nie **.Net core**)
  - Uczyń ten projekt aktywnym (**StartUp**)
- Ponieważ będziemy pracowali z zewnętrznymi bibliotekami (w stosunku do C#) języka Visual Basic .NET, musimy do naszego projektu podpiąć stosowną referencję:
- W drzewie struktury projektu prawokliknij **References** i wybierz **Add Reference...**
  - W kategorii **Assemblies** wyszukaj **Microsoft.VisualBasic** i zaznacz go do dołączenia do Twojego projektu
  - Na liście referencji projektu pojawi się wpis **Microsoft.VisualBasic**
  - Do kodu Twojego programu, w miejscu klauzul **using** dopisz: **using Microsoft.VisualBasic;**



podgląd

C#

Aplikacja konsolowa wsadowa (.NET framework)



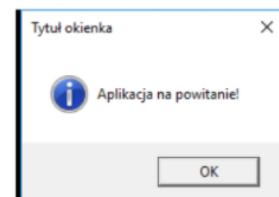
C#

## Okna dialogowe VB.NET dostępne z poziomu C#

```
Interaction.MessageBox("Aplikacja na powitanie!",
    MessageBoxStyle.OkOnly | MessageBoxStyle.Information,
    "Tytuł okienka");
```

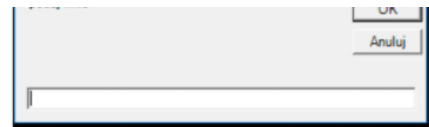
Dokumentacja: [https://msdn.microsoft.com/en-us/library/microsoft.visualbasic.interaction.msgbox\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/microsoft.visualbasic.interaction.msgbox(v=vs.110).aspx)

```
string imie = Interaction.InputBox("podaj imie",
    "Tytuł okienka");
```



nami dialogowymi

Dokumentacja: [https://msdn.microsoft.com/en-us/library/microsoft.visualbasic.interaction.inputbox\(v=vs.10\).aspx](https://msdn.microsoft.com/en-us/library/microsoft.visualbasic.interaction.inputbox(v=vs.10).aspx)



Strona 5 / 8



## C#

## Kod aplikacji

- Przekopiuj program z Lab. 1.1. Zastąp niektóre wywołania `Console.WriteLine()` na `MsgBox` oraz wczytywanie danych `Console.ReadLine()` na `InputBox`.
- Oczywiście, w swoim programie możesz mieszać kanały komunikacji – część informacji możesz wypisywać na konsolę (logi), zaś te najistotniejsze pobierać i wyświetlać w oknach dialogowych.
- Biblioteki języka C# nie przewidują funkcjonalności okienka dialogowego typu `InputBox`. Jest jednak odpowiednik okienka `MsgBox` w bibliotece Windows Form dla C#.

## &gt; Uwagi:

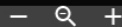
- Gdy linie kodu wychodzą poza obszar okna edytora, można włączyć zawijanie wierszy (`Edit → Advanced → Word Wrap`)
- Można również dzielić zbyt długie wiersze „enterem”. Dla kompilatora C# nie ma znaczenia wygląd kodu.
- `string.Format()` – buduje napis z użyciem symboli zastępczych.



podgląd



Strona 6 / 8



## C#

```
using System;
using Microsoft.VisualBasic;

namespace HelloConsoleVBDialo
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Program na powitanie.");

            string imie = Interaction.InputBox("Podaj imię", "Okienko imię");
            Console.WriteLine("podano imię " + imie);

            string nazwisko = Interaction.InputBox("Podaj nazwisko", "Okienko nazwisko");
            Console.WriteLine("podano nazwisko " + imie);

            Interaction.MsgBox("Witaj " + imie + " " + nazwisko);
            Interaction.MsgBox(string.Format("Witaj {0} {1}. Czy Pan {0} rzeczywiście nazywa się {1}?", imie, nazwisko));

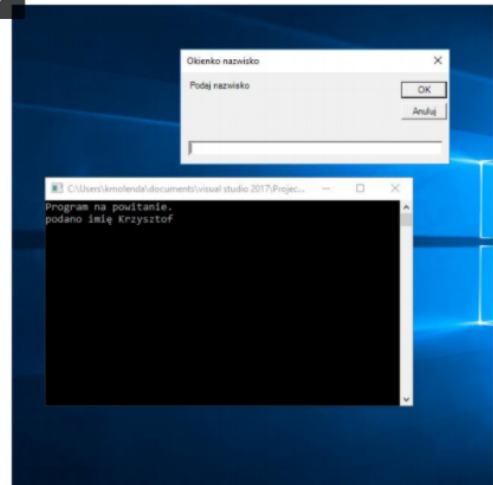
            int wiek = Convert.ToInt32(Interaction.InputBox("Podaj wiek", "Okienko wiek")); //możliwe zgłoszenie
            wyjątku

            if (wiek < 67)
            {
                Interaction.MsgBox("do emerytury zostało Ci " + (67 - wiek) + " lat", MsgBoxStyle.OkOnly |
                MsgBoxStyle.Information, "Okienko emerytura");
            }
            else
            {
                Interaction.MsgBox("jesteś emerytem", MsgBoxStyle.OkOnly | MsgBoxStyle.Information, "Okienko
                emerytura");
            }
        }
    }
}
```



# Testowanie aplikacji

- › Aplikacja (podobnie jak te z poprzednich lab.) jest jednopliskowa. Może być uruchomiona w systemie Windows z zainstalowanym środowiskiem .NET Framework 4.5.2.
- › Zamknięcie konsoli powoduje zamknięcie skojarzonych z nią okien dialogowych.
- › Błędy:
  - W przypadku pytania o wiek, podanie wartości niekonwertowalnej na typ całkowity (`int`) spowoduje zgłoszenie wyjątku `FormatException`.
  - Wprowadzenie zbyt dużej wartości całkowitej (np. 123456778987654321), przekraczającej zakres typu `int` spowoduje zgłoszenie wyjątku `OverflowException`.
  - W wyskakujących okienkach dialogowych `InputBox` kliknięcie w przycisk `Anuluj` lub zamknięcie okienka powoduje zwrócenie pustego napisu ("" ) → tzw. pusty `string`. Ta wartość jest zapamiętana w odpowiedniej zmiennej (`imie`, `nazwisko` lub `wiek`). W przypadku zmiennej `wiek` zgłaszany jest wyjątek `FormatException` (nieudana próba przekonwertowania pustego `string`-u do liczby całkowitej).



**Zadanie:** Korzystając z instrukcji warunkowego wyboru rozbuduj aplikację w taki sposób, aby kliknięcie w przycisk `Anuluj` (w dowolnym momencie jej działania) spowodowało jej zakończenie z komunikatem na konsoli o treści `"Koniec"`.

Należy sprawdzać wynik zwracany przez `InputBox`:

```
if( wynik == "" ) ...
```