



POLITECHNIKA POZNAŃSKA
Poznan University of Technology

Skrypt do zajęć: **Modelowanie i Programowanie Obiektowe**
Visual Studio 2010: Ultimate

Andrzej Stroiński

14.10.2011

Zajęcia laboratoryjne przeprowadzane są na komputerach z systemem operacyjnym Windows 7 z wykorzystaniem oprogramowania Visual Studio 2010 w wersji Ultimate. Poniżej omówiono kilka często wykorzystywanych skrótów:

VS – Visual Studio

LPM – Lewy przycisk myszy

PPM – Prawy przycisk myszy

Zagadnienie 1: Tworzenie aplikacji oraz jej uruchomienie.

1. **Solucja** to kontener, który może zawierać projekty oraz pewne specjalistyczne obiekty dla solucji. Głównym jej zadaniem jest grupowanie oraz ułatwienie zarządzania powiązanych ze sobą projektów. Przykładem, jest realizacja bardzo dużego programu, który korzysta z implementowanej przez tych samych programistów biblioteki. W celu ułatwienia zarządzania rozwojem oprogramowania zarówno główny projekt jak i wspomniana biblioteka, mogą być elementami tej samej solucji (ułatwia to nawigowanie pomiędzy nimi podczas ich równoczesnego rozwoju). [1]
Drugim kontenerem ułatwiającym zarządzanie procesem wytwarzania oprogramowania są **projekty**. Projekty grupują pliki z kodami źródłowymi oraz pozwalają na zarządzanie bibliotekami oraz pewnymi właściwościami (debugowanie, kompilacja, architektura, docelowa wersja platformy itd.) pisanej aplikacji.[2]
2. Utwórz nowy projekt. Z menu wybierz *File->New->Project...*
3. Otworzy się nowe okno. Po lewej stronie znajdują się zainstalowane grupy szablonów umożliwiające nam utworzenie projektów w jednym z dostępnych języków. Na środku wypisane są dostępne szablony, które definiują pewne standardowe właściwości tworzonego projektu oraz generują kod. Po prawej można znaleźć opis do czego należy wykorzystywać poszczególne szablony.
4. Wybierz *Visual C# Console Application*.
5. Wpisz nazwę projektu: *MyApp* oraz zaznacz opcję *Create directory for solution*.
6. Otworzy się nowy widok w oknie głównym VS, z domyślnie utworzoną klasą o nazwie *Program.cs*, będącą pierwszym plikiem stworzonego projektu.
7. Wygenerowany kod klasy:

// Sekcja informująca środowisko z jakich bibliotek dołączonych do projektu

definiowana klasa będzie korzystał (przestrzeń nazw w C#), które będą wykorzystywane w projekcie.

// Należy pamiętać, aby zbiór bibliotek był zawsze minimalny.

```
using System; // zawiera podstawowe klasy, często używane
wartości // oraz typy
using System.Collections.Generic; // odpowiedzialna za klasy tzw. kolekcji
using System.Linq; // klasy i interfejsy umożliwiające
korzystanie z // Language-Integrated Query (LINQ)
using System.Text; // manipulacje znakami oraz ciągami znaków

namespace MyApp // deklaracja przestrzeni nazw (ułatwia
organizację // kodu oraz tworzenie globalnie
// unikalnych typów
{
    class Program // definicja klasy
    {
        static void Main(string[] args) // statyczna metoda "główna" programu
        pozwalająca na // pobranie parametrów z linii
        poleczeń
        {
            // kod aplikacji
        }
    }
}
```

8. W celu uruchomienia projektu należy z menu wybrać: *Debug->Start Debugging* (jeśli chcemy przejść do widoku debuggowania) lub *Debug->Start Without Debugging*
9. W celu zbudowania solucji (re-kompilacji kodu) wybrać należy *Build->Build Solution*
10. Zmodyfikuj wygenerowany kod w taki sposób, aby wypisywał na standardowym wyjściu parametry przekazane w konsoli. Wykorzystaj do tego metodę:
`System.Console.WriteLine("");`
 oraz konstrukcję pętli:
`for (int i = 0; i < args.Length; i++)`
11. Argumenty linii poleceń podaje się we właściwościach projektu: *PPM na projekcie->Properties->Debug->Start Options->Command line arguments*

Zagadnienie 2: Dostępne widoki

W VS 2010 w menu *View* znajduje się kilka tzw. widoków(*ang. views*) które ułatwiają proces wytwarzania oprogramowania:

- *Solution Explorer* – widok służący do przeglądania oraz modyfikowania elementów projektu oraz solucji. Pozwala na ustalenie właściwości projektu (PPM na nazwie projektu w drzewie *Solution Explorer->Properties*)
- *Project Properties* – składa się z następujących podkategorii:
 - *Application* – nazwa namespace, docelowa wersja framework'a, typ aplikacji, ustalenie ikony itd.
 - *Build* – opcje jak "budować" projekt, architektura, poziom zgłaszanych ostrzeżeń
 - *Build Events* – polecenia wykonywane przed / po zbudowaniu projektu
 - *Debug* – opcje uruchomienia testowego, parametry podawane z linii poleceń
 - ... - mniej istotne opcje
- *Properties* – widok, pozwalający na zmianę właściwości zaznaczonego w *Solution*

Explorer elementu.

- *Team Explorer* – rozbudowany klient pozwalający na korzystanie z TFS (Team Foundation Server): wersjonowanie, śledzenie zmian, kolejkovanie zadań itd.
- *Server Explorer* – zarządzanie połączeniami do baz danych, logowanie do nich oraz zarządzanie nimi.
- *Output* – wyświetla wiadomości ze statusem wielu zintegrowanych narzędzi IDE np. informacje dotyczące kompilacji kodu.
- *Error List* – wyświetla błędy oraz ostrzeżenia kompilacji
- *Object Browser* – pozwalana przeglądanie hierarchicznej struktury aktualnego zakresu wykorzystywanych klas. Udostępnia informacje na temat: obiektów, właściwości, metod, pól itd.
- *Class View* – widok klasy, pozwalający podglądać elementy klasy, wyszukiwać wystąpienia (*PPM->Find All References*), przechodzić do definicji (*PPM->Go to Definition*), itd.

Zagadnienie 3: Debugowanie

VS 2010 oferuje bardzo rozbudowany tryb debugingu. Pozwala on na podglądanie aktualnego stanu zmiennych, stosu wykonania oraz przechowuje historię wykonanych kroków testowanego programu.

- *Locals* – widok przedstawiający aktualnie dostępne zmienne, ponadto klikając PPM na zmiennej i wybierając *Edit Value* można zmienić jej aktualną wartość
- *Watch* – widok, na którym znajdują się informacje o wartościach "śledzonych" zmiennych (dodanie zmiennej do widoku: *PPM->Add Watch*), ponadto można za pomocą tego widoku edytować wartości zmiennych.
- *Call Stack* – Stos aktualnych wywołań, widać przekazane do funkcji parametry,
- *IntelliTrace* – Zapisuje historię przetwarzania aplikacji umożliwiając powrót do zapamiętanych punktów z historii.
- *Immediate Window* – wpisane zmienne natychmiast są wypisywane, wykonywanie „w locie” kodu.

Breakpoint: W celu wstawienia „*breakpoint'a*” (punktu przerwania) należy kliknąć LPM na szarym pasku obok widoku pliku źródłowego. Punkt ten powoduje w trybie debugingu zatrzymanie się wykonania programu zaraz przed wykonaniem linii nim oznaczonej. Pozwala to na przeanalizowanie wartości poszczególnych zmiennych (np. „najechanie” na nie myszą) dzięki czemu dość łatwo można wyeliminować błędy.

Nawigacja w trybie debugingu:

Step-over: F10 – następna linia

Step-into: F11 – następna linia („wejdz” do wywoływanej funkcji)

Step-out: Shift+F11 – powrót do poprzedniego kontekstu

1. Dodaj do wcześniej wykorzystywanego programu funkcję pobierającą jako parametr zmienną typu całkowitego, zwiększającą jej wartość o 1 i zwracającą wynik. Niech w funkcji Main w pętli znajduje się zmienna inkrementowania dodaną metodą:

```
static void Main(string[] args)
{
    int z = 0;
    for (int i = 0; i < args.Length; i++ )
    {
        ...
        z=Inc(z);
    }
    ...
}
public static int Inc(int param)
{
    ...
}
```

2. Sprawdź działanie wyżej wymienionych widoków.
3. Dodaj zmienną "z" oraz "param" do zmiennych "śledzonych". Zauważ co się dzieje ze zmienną "param" jak program kończy wykonanie metody "Inc" (UWAGA! Może być wymagane odświeżenie wartości zmiennej).

Zagadnienie 4: Inne IDE dla języka C#:

- Sharp Develop - <http://www.icsharpcode.net/OpenSource/SD/>
- Mono Develop - <http://monodevelop.com/>

Bibliografia:

- [1] Solutions as Containers: [http://msdn.microsoft.com/en-us/library/df8st53z\(v=VS.80\).aspx](http://msdn.microsoft.com/en-us/library/df8st53z(v=VS.80).aspx)
- [2] Solutions, Projects and Items: [http://msdn.microsoft.com/en-us/library/b142f8e7\(v=VS.80\).aspx](http://msdn.microsoft.com/en-us/library/b142f8e7(v=VS.80).aspx)
- [3] MSDN: <http://msdn.microsoft.com/>
- [4] C# Practical Learning: <http://www.functionx.com/csharp/>
- [5] C# for beginners: <http://www.csharp-help.com/2006/12/c-tutorial-for-beginners/>
- [6] C# tutorial: <http://csharpcomputing.com/Tutorials/>
- [7] C# tutorials: <http://csharp.net-tutorials.com/>