

MS Visual Studio jest narzędziem do pisania programów nie tylko w C#. W Visual Studio można pisać programy w językach jak C++, Visual Basic i F# oraz wielu innych po dodaniu wtyczek (IronPython, IronRuby). Platforma .NET nie ogranicza się do C#. Lista wszystkich języków w platformie .NET jest dosyć spora.

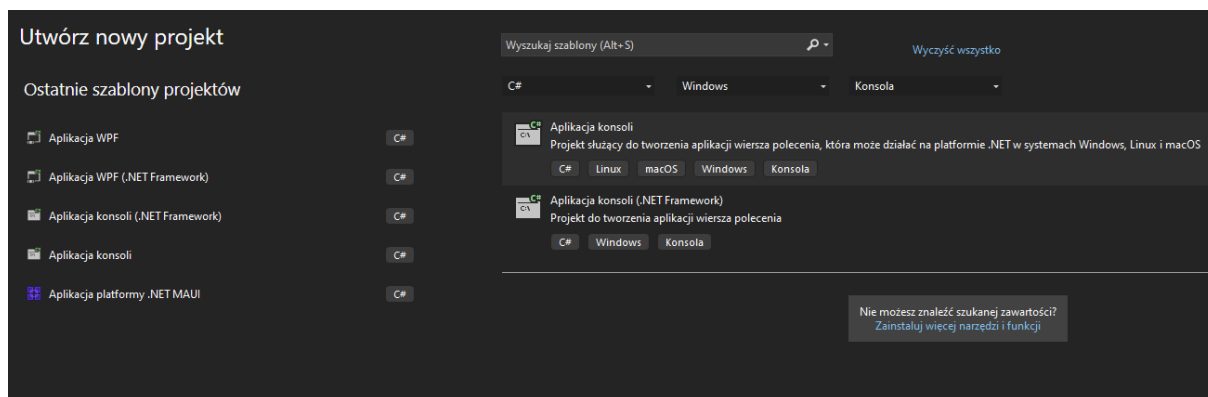
Za pomocą C# i Visual Studio można napisać:

- **Aplikacje konsolowe** – przydatne do pisanie prostych aplikacji gdzie kolorowy interface nas nie interesuje.
- **Aplikacje Windows Form** – Aplikacje okienkowe, które powoli odchodzą w niepamięć z powodu starej mechaniki i grafiki rodem z Windows 3.11. Jednak wciąż są stosowane.
- **Aplikacje WPF** – Aplikacje okienkowe z lepszą grafiką (wektorową) i mechaniką. Początkującego mogą wprowadzić w małe zamieszanie z powodu kodu XAML, który zarządza wyglądem aplikacji.
- **Aplikacje internetowe ASP.NET** – Pod samą nazwą ASP.NET w sumie kryją się aż 3 sposoby pisania technologii internetowych.
- **Gry na XNA** – W C# też można pisać gry i na pewno jest to dużo prostsze niż w innych językach ,platformach. Gra napisana na XNA może działać na telefonie Windows Phone 7, Windows oraz na XBOX. Poważny interes ,ale jeśli jesteś ambitny może to jest właśnie dla ciebie.
- **Aplikacje na telefon Windows Phone 7** – W C# można pisać aplikacje na ten telefon. W zależności od potrzeby może być to gra XNA , bądź trochę zmodyfikowany Silverlight na urządzenia mobilne.
- **Usługa sieciowa WCF, Chmura , Office , SharePoint...**

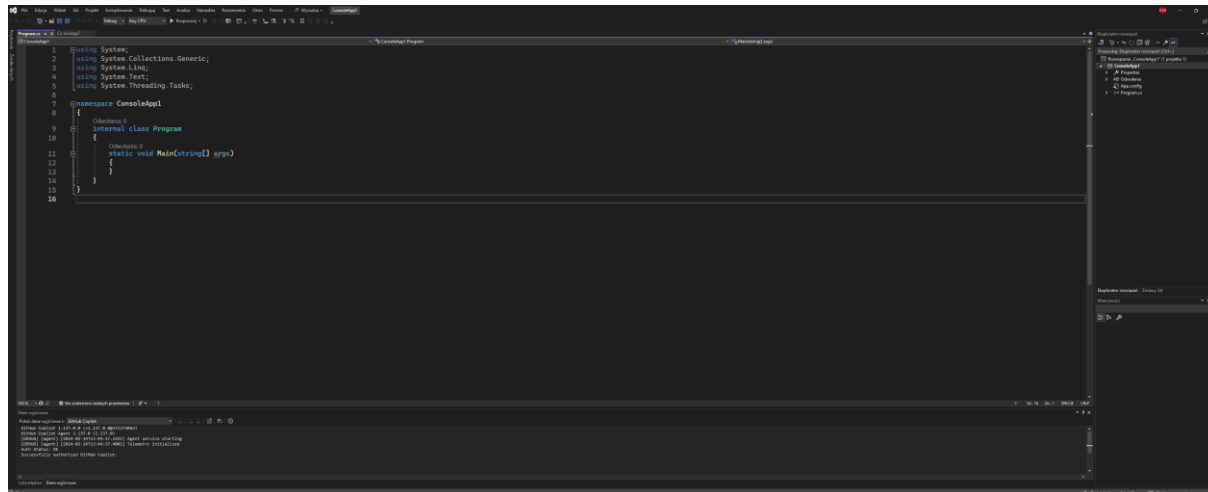
Pierwszy program C#

Przy pierwszym uruchomieniu trzeba wybrać styl środowiska, w którym chcemy pracować. Wybierz „C#”.

Kliknij na „Utwórz nowy projekt”, a wyświetli się okno z wieloma typami możliwych projektów. Należy wybrać z listy „C#” (nie VB!), „Windows” i „Konsola”, a następnie Aplikacja konsoli (.NET Framework).



Visual Studio automatycznie tworzy folder z nazwą projektu więc po wciśnięciu „OK” zostanie utworzony projekt w ścieżce „C:\Users\Nazwa_użytkownika\source\repos\ConsoleApplication1\”. Nazwę projektu można oczywiście zmienić.



Nasze okno startowe wygląda mniej więcej tak, jak na obrazku powyżej. Okno „Eksplorator rozwiązań” pokazuje jakie pliki są w naszym projekcie. Visual Studio stworzył m.in. następujące pliki do projektu:

- **Solucja „ConsoleApplication1”** - Na samej górze;jeden na aplikacje. Jeśli otworzy folder z projektem zobaczysz plik

“ConsoleApplication1.sln”, który jest plikiem referującym się do tego projektu. Jednak każda solucja może zawierać więcej takich plików projektowych. Do aplikacji konsolowej mamy tylko jeden projekt więc nie ma podziału na solucje.

- **Program.cs** - To jest plik źródłowy, w którym obecnie piszemy aplikację C#. W aplikacji konsolowej jest to główny plik z gotowym kodem stworzonym przez Visual Studio.

W pliku Program.cs jest zdefiniowana klasa “Program” która zawiera w sobie metodę Main. Ta metoda jest stworzona specjalnie do aplikacji konsolowej. Możesz teraz uruchomić aplikację za pomocą skrótu klawiszowego F5 (bądź F5 z ctrl-em) albo z menu wybrać “Debugowanie/Rozpocznij Debugowanie” albo “ Debugowanie/Uruchom bez debugowania”.

Kliknij na plik “Program.cs” i wewnątrz niego między nawiasami w metodzie Main {} postaw Enter. Zaczynaj pisać słowo Console po to, by pierwszy raz zobaczyć pomocnika Visual Studio, który nazywa się “IntelliSense”.

Wyświetla ona pełen opis tej klasy jak i inne rzeczy, które zaczynają się od słowa “Console”. Po napisaniu słowa “Console” i po dodaniu kropki zobaczysz wszystkie metody i właściwości, które ta klasa oferuje.

Przydałoby się , aby nasza konsola coś do nas napisała i tak będzie to symboliczny “Hello World!”.

Napisz "Console.WriteLine()" i zobacz jak Visual Studio sam cię nakierowuje tym razem na tę metodę.

Metoda “WriteLine” napisze nam w konsoli w jednej linii dane słowo.

Ostatecznie kod aplikacji wygląda następująco. Do metody “Writeline” w nawiasach podaliśmy tekst, który zostanie wyświetlony w oknie konsoli.

```
Console.WriteLine("Hello World!");
```

Uwaga: po kodzie który kończy operację trzeba do kodu dodać znak “;”.

Uruchom aplikację bez debugowania (ctrl-F5) ,a zobaczysz następujący rezultat:

A screenshot of a Windows command prompt window. The title bar shows 'C:\WINDOWS\system32\cmd.exe'. The command prompt displays 'Hello World!' on the first line and 'Press any key to continue . . .' on the second line. The background is black, and the text is white.

Zauważ ,że gdy uruchomisz aplikację z debugowaniem, zobaczysz tylko na chwilę napis stworzony przez aplikację, po czym okno się zamknie. Musimy zatrzymać aplikację konsolową, abyśmy mogli zobaczyć rezultat.

W tym celu najlepiej np. spróbować odczytać klawisz użytkownika. W ten sposób aplikacja nie zostanie zamknięta dopóki użytkownik nie wciśnie klawisza. Zrobimy to poprzez dodanie następującego kodu:

```
Console.ReadKey();
```

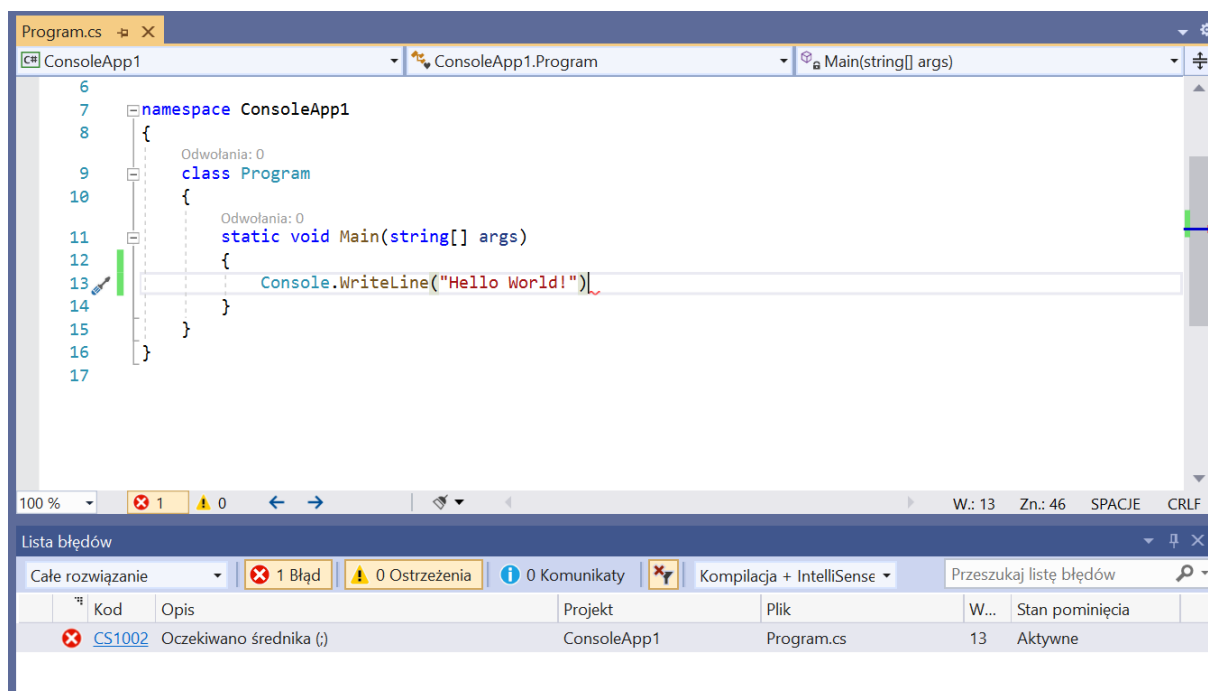
```
//Program czeka aż użytkownik wciśnie klawisz
```

Zielony tekst to komentarz. Rozpoczynamy go od dwóch znaków „//”.

Komentarz nie jest odczytywany przez kompilator i służy wyłącznie pomocą programistom.

Zwykle pierwsze pisanie programów wiąże się z drobnymi błędami np. zapominaniem znaku “;” po napisaniu metody. Na szczęście Visual Studio zaznacza nasze błędy i dokładnie je opisuje. Gdyby aplikacja nie mogła się skompilować wyskoczy nam okno, czy chcemy uruchomić ostatnią dobrą kompilację. Jeżeli wybierzemy nie, to pod oknem edytora kodu Visual Studio

wyświetli nam się lista błędów. Klikając dwa razy na dany błąd możemy od razu przejść do miejsca w kodzie, w którym on występuje.



Program, który napisaliśmy jest dostępny w formie pliku exe w podfolderze projektu **“bin/Debug/ConsoleApplication1.exe”**. Możemy go uruchomić kiedy chcemy już poza środowiskiem Visual Studio.