

# Notatki do 1. tygodnia kursu Dotneta

Maciek Mielczarek

5 sierpnia 2020

## 1 Co ja czytam?

W tej serii dokumentów zamierzam opisywać drobne problemy pojawiające się podczas przerabiania kursu, w miarę możliwości razem z rozwiązaniami. Będą tu też opisy tego co robię inaczej na Linuxie niż na Windowsie. Staram się nie zakładać, że świetnie znasz język angielski.

## 2 Instalacja

Problem:

- Mam już zainstalowane Visual Studio Community i okazało się że to wersja demonstracyjna, tylko na 30 dni.

Rozwiązanie:

- Zaloguj się do swojego konta Microsoft w Visual Studio, aby zmienić licencję. Odpowiednie miejsce do logowania to "Help -> About Microsoft Visual Studio -> License status (w górnym prawym rogu nowego okna)". Jeśli nie masz konta Microsoft, to je stwórz.

Problem:

- Chcę przerabiać ten kurs na Linuxie (albo Macu), ale nie ma Visual Studio na Linuxa (ani Maca).

Rozwiązanie:

- Rzeczywiście, nie ma. Są jednak inne IDE, jak VS Code i Monodevelop. Mój główny system to Linux i na nim również (mam też dostęp do Windowsa) będę się starał wykonywać zadania z kursu i opisywać związane z tym problemy. Jeśli więc masz zamiar przerabiać kurs na Linuxie, to wiedz, że nie jesteś z tym sam(a). O Macach się nie wypowiadam, bo nie mam żadnego pod ręką żeby cokolwiek na nim sprawdzać.

Problem:

- To co w końcu mam zainstalować na Linuxie?

Rozwiązanie:

- VS Code,
- Monodevelop,
- Azure Data Studio (chyba mniejwięcej odpowiednik SQL Server Management Studio),
- Postman.

## 3 Hello World

### 3.1 ...bez IDE

Jak stworzyć i uruchomić swój pierwszy program w C# na Linuxie:

1. Zainstaluj ".NET Core SDK"
2. Sprawdź, czy jest zainstalowane: Wpisz w terminalu "dotnet".
3. Jeśli wyświetlą się jakieś informacje o dotnecie, to kontynuuj, w przeciwnym przypadku wróć do kroku 1.
4. Wpisz "dotnet new console -o myApp".  
To stworzy aplikację konsolową "Hello World" w folderze myApp.
5. cd myApp  
dotnet run

Trochę cierpliwości. To wprowadzie tylko "Hello World", ale i tak uruchamianie chwilę trwa.

Jak stworzyć więcej programów konsolowych:

1. `dotnet new console -o nazwaProgramu`
2. Edytuj plik `./nazwaProgramu/Program.cs`

Jak je uruchomić:

1. Przejdź do folderu "nazwaProgramu"
2. `dotnet run`

### 3.2 ...w Monodevelopie

1. Wejdź w File -> New Solution.
2. Wybierz "App" pod ".NET Core".
3. Wybierz "Console Application" i kliknij next.
4. Odznacz "Create a project directory within the solution directory", aby uzyskać strukturę folderów taką jak na kursie. Kliknij create.
5. Wybierz z menu Run -> Run without debugging.

Jeśli program nie zadziała (to jest nie zobaczysz okienka z napisem "Hello World!") i zobaczysz wiadomość podobną do tej:

```
It was not possible to find any compatible framework version
The framework 'Microsoft.NETCore.App', version '1.1.2' was not found.
- The following frameworks were found:
    3.1.6 at [/usr/share/dotnet/shared/Microsoft.NETCore.App],
```

to kliknij File -> Close Solution, następnie otwórz plik `/HelloWorld/HelloWorld.csproj` i w linii:

```
<TargetFramework>netcoreapp1.1</TargetFramework>
```

zmień numer "1.1" (bo w komunikacie o błędzie jest "wersja 1.1.coś nie została znaleziona") na "3.1" (bo w komunikacie zaraz potem jest "znaleziono (...) 3.1.coś"). Potem każ Monodevelopowi otworzyć zamknięte przed chwilą rozwiązanie i ponownie wykonaj ostatni krok (Run -> Run without debugging). Teraz powinno zadziałać.

## 4 Praca z IDE

W Monodevelopie tworzenie nowych, pustych klas przebiega tak jak w VS, natomiast w konsoli wśród komend "dotnet new..." nie ma takiej tworzącej pustą klasę. Stworzenie odpowiedniego szablonu przypisującego nowemu plikowi i klasie nazwę najwyraźniej nie jest trywialne. Jeśli będę tworzył dużo nowych, pustych klas, to może napiszę do tego jakiś skrypt w bashu.

Breakpointy w Monodevelopie nie działają i nie udało mi się ich uruchomić. Gdy będę chciał poznać wartości jakichś zmiennych w jakimś miejscu programu, to zawsze mogę je wprost wypisać.

Wracam do VS. Nie widziałem narzędzi diagnostycznych, ale kliknięcie w Debug -> Windows -> Show Diagnostic Tools podczas działania programu włącza je.

## 5 Kompilowanie

Ciekawe jak to wygląda z kompatybilnością? Sprawdzam... Pliki wykonywalne przenoszone między Windowsem a Linuxem oczywiście nie działają i Wine automatycznie tego nie zmienia. Projekty kopiowane między systemami otwierają i uruchamiają się natomiast bez problemów.

## 6 Debugowanie

Zauważyłem, że podczas debugowania widoczne są zmienne, które zostaną stworzone dopiero za kilka liniiek. Sprawdziłem, że schowanie tych kolejnych liniiek w nawiasie klamrowym chowa te zmienne przed debuggerem do momentu wejścia do środka tego nawiasu.