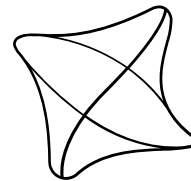


# 1



## Úvod do jazyka C# a platformy .NET

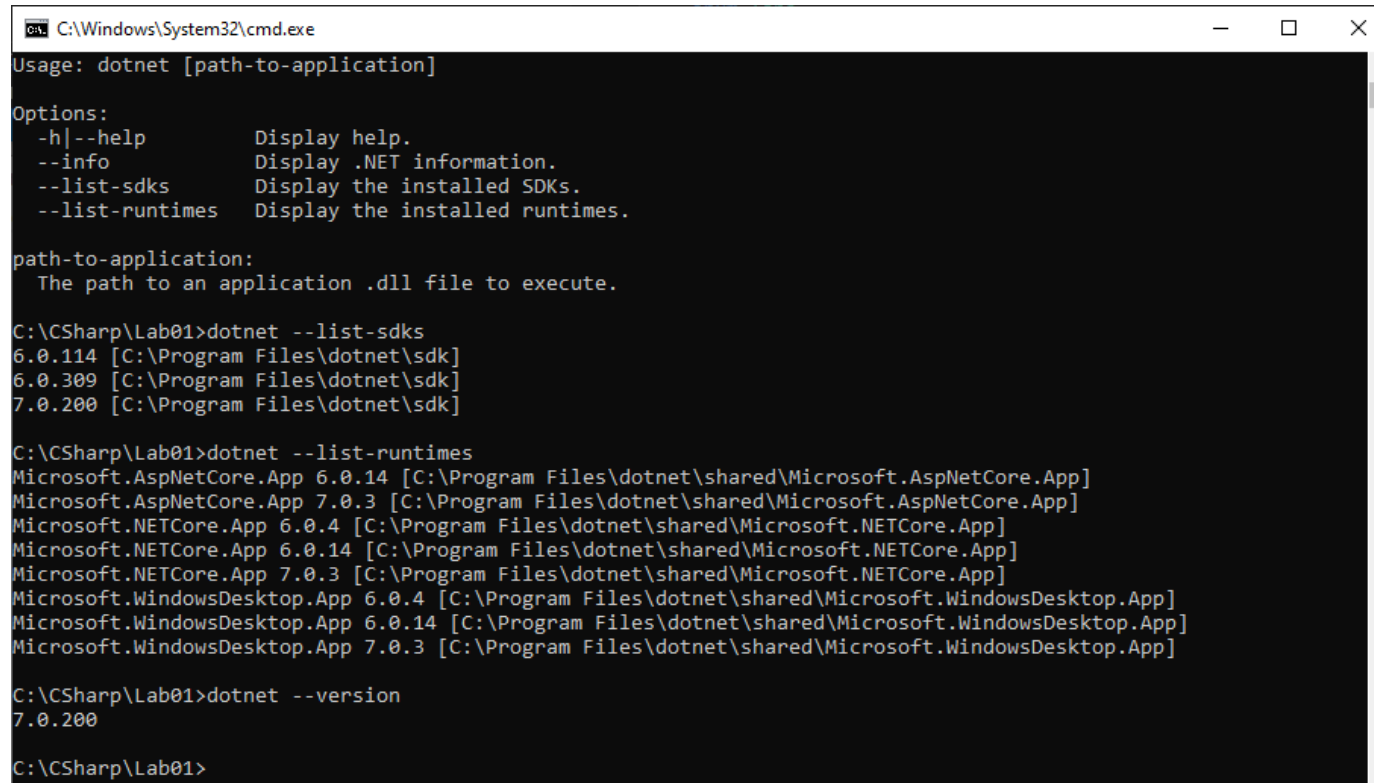
Ing. Štefan Toth, PhD.

# Obsah cvičenia

- Základy práce s .NET CLI a VS Code
- Vytvorenie projektu konzolovej aplikácie
- Práca s Jupyter notebookmi
- Práca s triedou Console, štruktúrou DateTime, formátovanie výstupu
- Použitie a práca s dokumentáciou (Microsoft Learn)

# Úloha 0 (.NET CLI) – zoznámte sa s príkazovým riadkom

- **Cez .NET CLI zistíte, aké máte nainštalované verzie .NET SDK a runtimov**



```
C:\Windows\System32\cmd.exe
Usage: dotnet [path-to-application]

Options:
  -h|--help          Display help.
  --info             Display .NET information.
  --list-sdks        Display the installed SDKs.
  --list-runtimes    Display the installed runtimes.

path-to-application:
  The path to an application .dll file to execute.

C:\CSharp\Lab01>dotnet --list-sdks
6.0.114 [C:\Program Files\dotnet\sdk]
6.0.309 [C:\Program Files\dotnet\sdk]
7.0.200 [C:\Program Files\dotnet\sdk]

C:\CSharp\Lab01>dotnet --list-runtimes
Microsoft.AspNetCore.App 6.0.14 [C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App]
Microsoft.AspNetCore.App 7.0.3 [C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App]
Microsoft.NETCore.App 6.0.4 [C:\Program Files\dotnet\shared\Microsoft.NETCore.App]
Microsoft.NETCore.App 6.0.14 [C:\Program Files\dotnet\shared\Microsoft.NETCore.App]
Microsoft.NETCore.App 7.0.3 [C:\Program Files\dotnet\shared\Microsoft.NETCore.App]
Microsoft.WindowsDesktop.App 6.0.4 [C:\Program Files\dotnet\shared\Microsoft.WindowsDesktop.App]
Microsoft.WindowsDesktop.App 6.0.14 [C:\Program Files\dotnet\shared\Microsoft.WindowsDesktop.App]
Microsoft.WindowsDesktop.App 7.0.3 [C:\Program Files\dotnet\shared\Microsoft.WindowsDesktop.App]

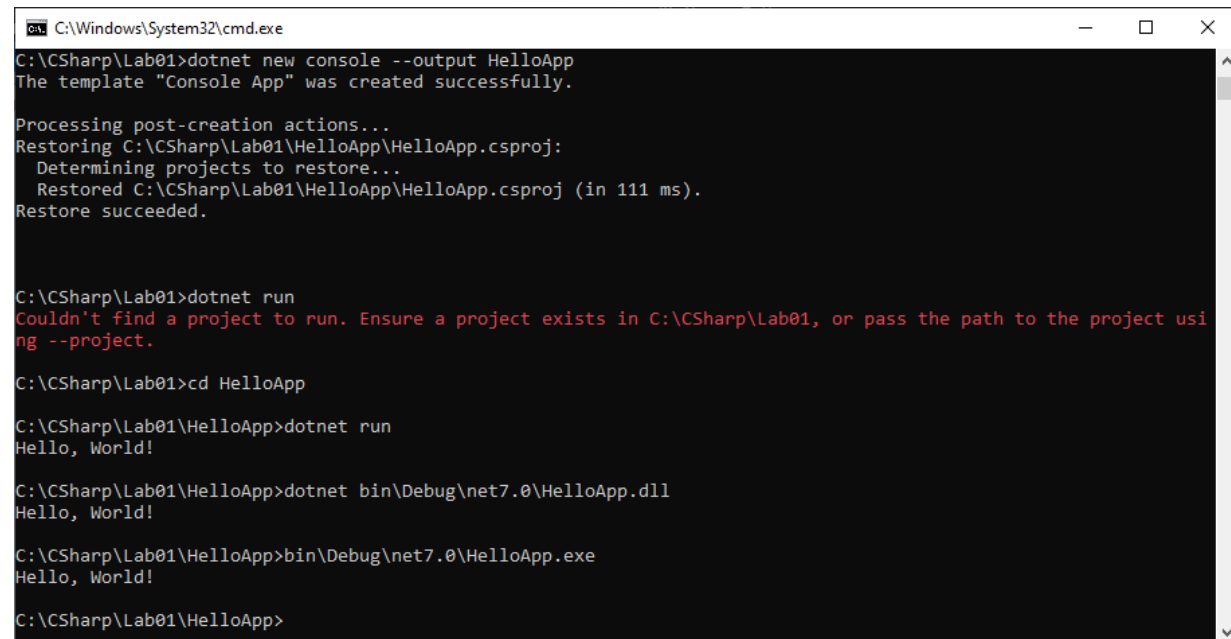
C:\CSharp\Lab01>dotnet --version
7.0.200

C:\CSharp\Lab01>
```

- **Vyskúšajte si aj pomocníka cez dotnet help a dotnet --help**

# Úloha 1 (HelloApp) – vytvorenie projektu

- Cez príkazový riadok **vytvorte** nový projekt **HelloApp**
- Preskúmajte vytvorené **priečinky a súbory**
- Spustite program cez príkazový riadok (**dotnet run**) a opäť preskúmajte vytvorené **priečinky a súbory** (napr. cez *dir /b /s* alebo Prieskumníkom)
- Vyskúšajte si spustenie programu aj cez:
  - **dotnet HelloApp.dll**
  - **HelloApp.exe** (alebo **.\HelloApp** na iných OS ako Windows)



```
C:\Windows\System32\cmd.exe
C:\CSharp\Lab01>dotnet new console --output HelloApp
The template "Console App" was created successfully.

Processing post-creation actions...
Restoring C:\CSharp\Lab01\HelloApp\HelloApp.csproj:
  Determining projects to restore...
  Restored C:\CSharp\Lab01\HelloApp\HelloApp.csproj (in 111 ms).
Restore succeeded.

C:\CSharp\Lab01>dotnet run
Couldn't find a project to run. Ensure a project exists in C:\CSharp\Lab01, or pass the path to the project using --project.

C:\CSharp\Lab01>cd HelloApp

C:\CSharp\Lab01\HelloApp>dotnet run
Hello, World!

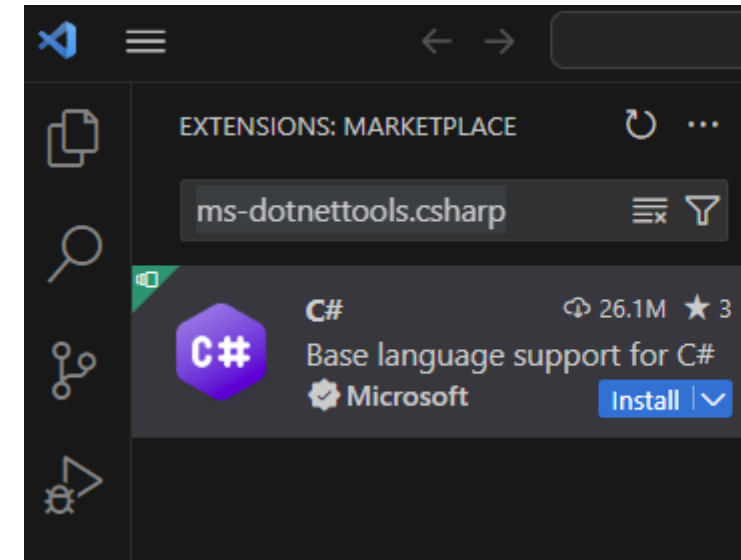
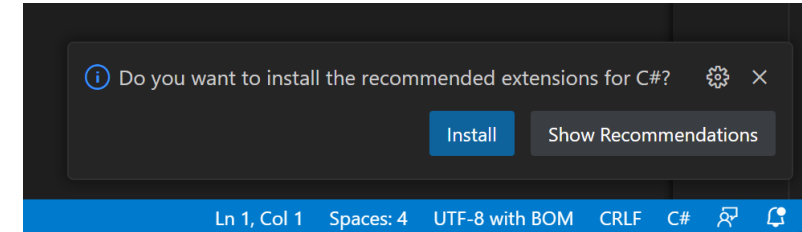
C:\CSharp\Lab01\HelloApp>dotnet bin\Debug\net7.0\HelloApp.dll
Hello, World!

C:\CSharp\Lab01\HelloApp>bin\Debug\net7.0\HelloApp.exe
Hello, World!

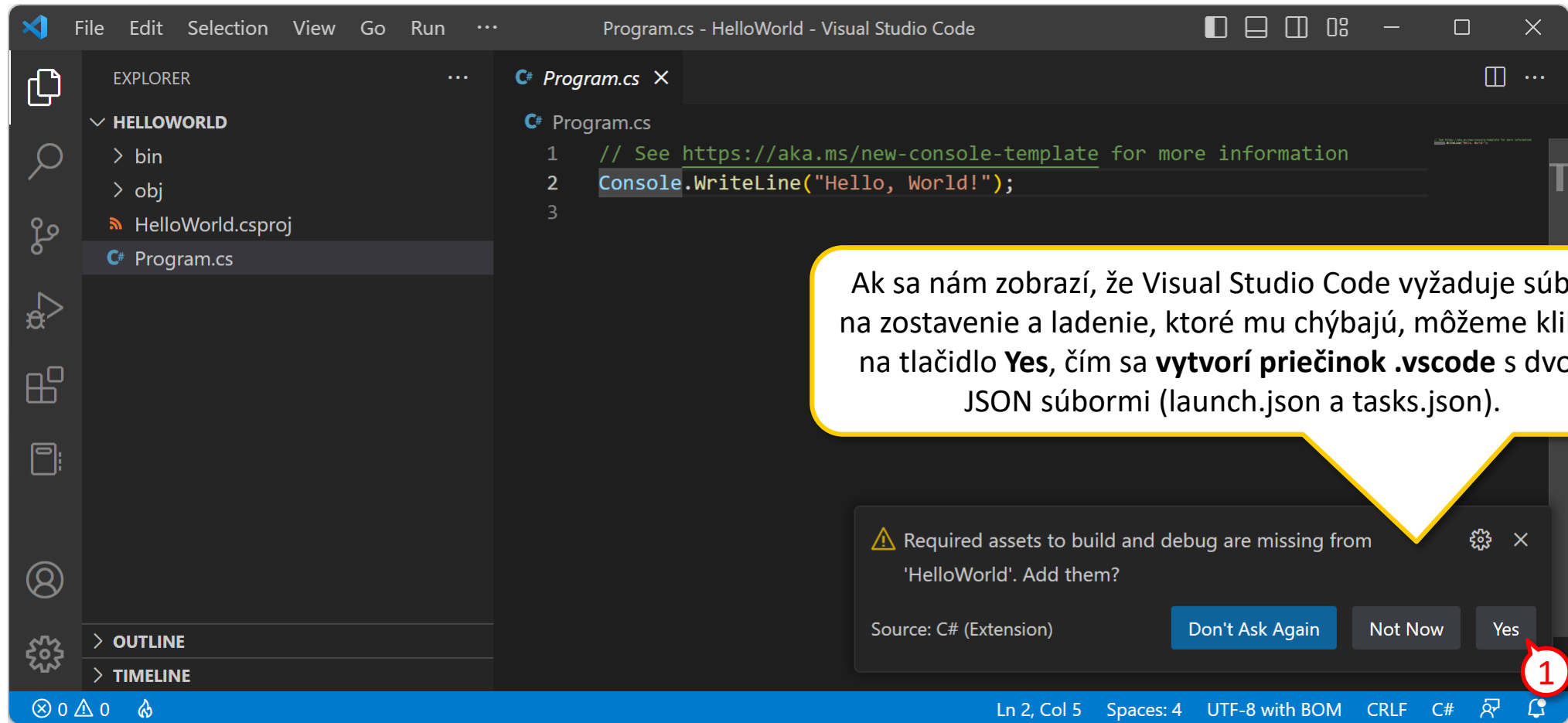
C:\CSharp\Lab01\HelloApp>
```

# Úloha 1.1 (HelloApp) – VS Code

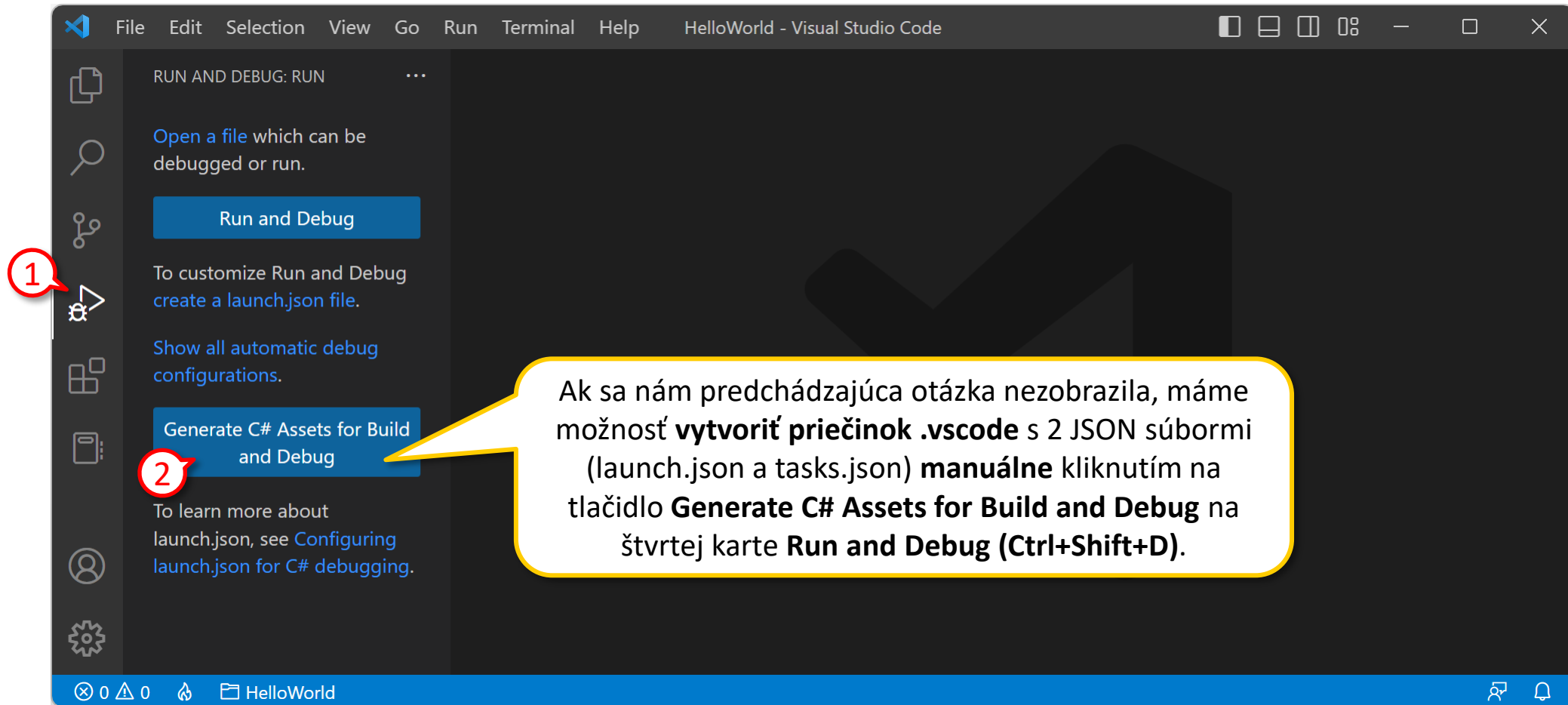
- **Otvorte** priečinok projektu cez **Visual Studio Code (VS Code)**
- Ak ešte nemáte nainštalované rozšírenie **C#** alebo **C# Dev Kit**, tak si ho vo VS Code nainštalujte
  - ak si otvoríte zdrojový kód Program.cs, mal by Vám ho VS Code ponúknuť nainštalovať
  - ak zmizne alebo sa vôbec nezobrazí, nainštalujte si rozšírenie (extension) manuálne vyhľadáním
- **Spustite program** v režime ladenia **vo VS Code**, vyskúšajte si aj zarážku (breakpoint) a ladenie
  - Budete potrebovať vygenerovať súbor launch.json



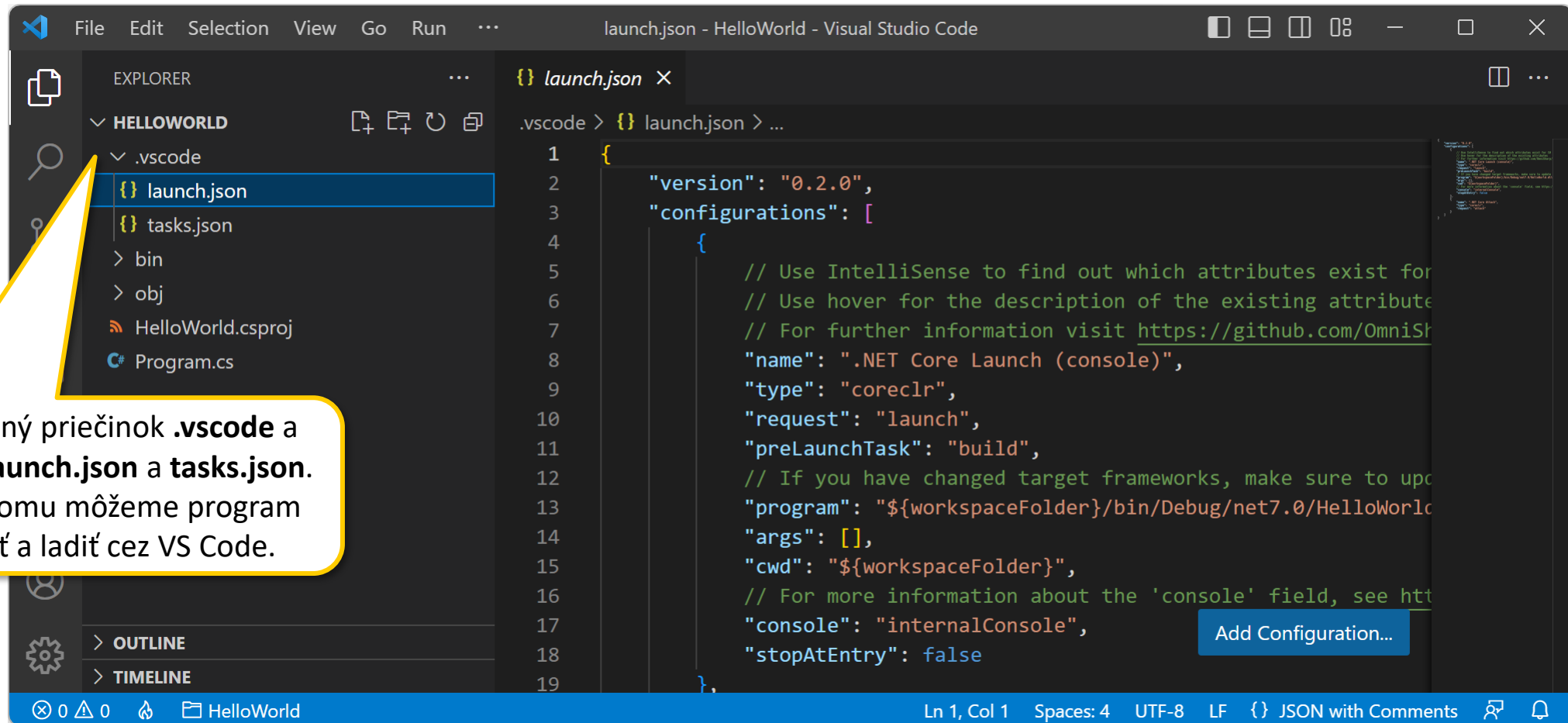
# VS Code – vygenerovanie súborov na ladenie (1)



# VS Code – vygenerovanie súborov na ladenie (2)

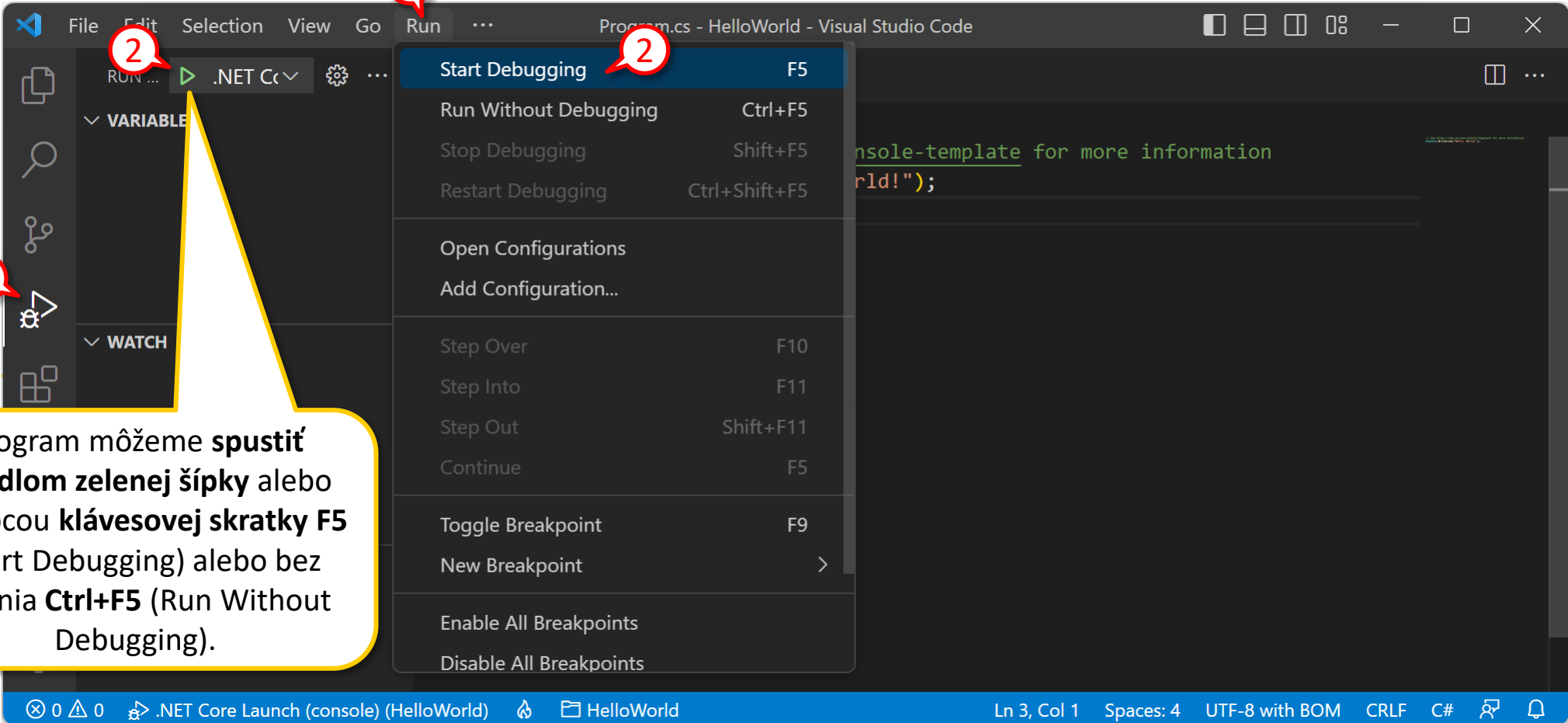


# VS Code – vygenerovanie súborov na ladenie (3)





# VS Code – spustenie programu



The screenshot shows the Visual Studio Code interface with the 'Run' menu open. The menu options are:

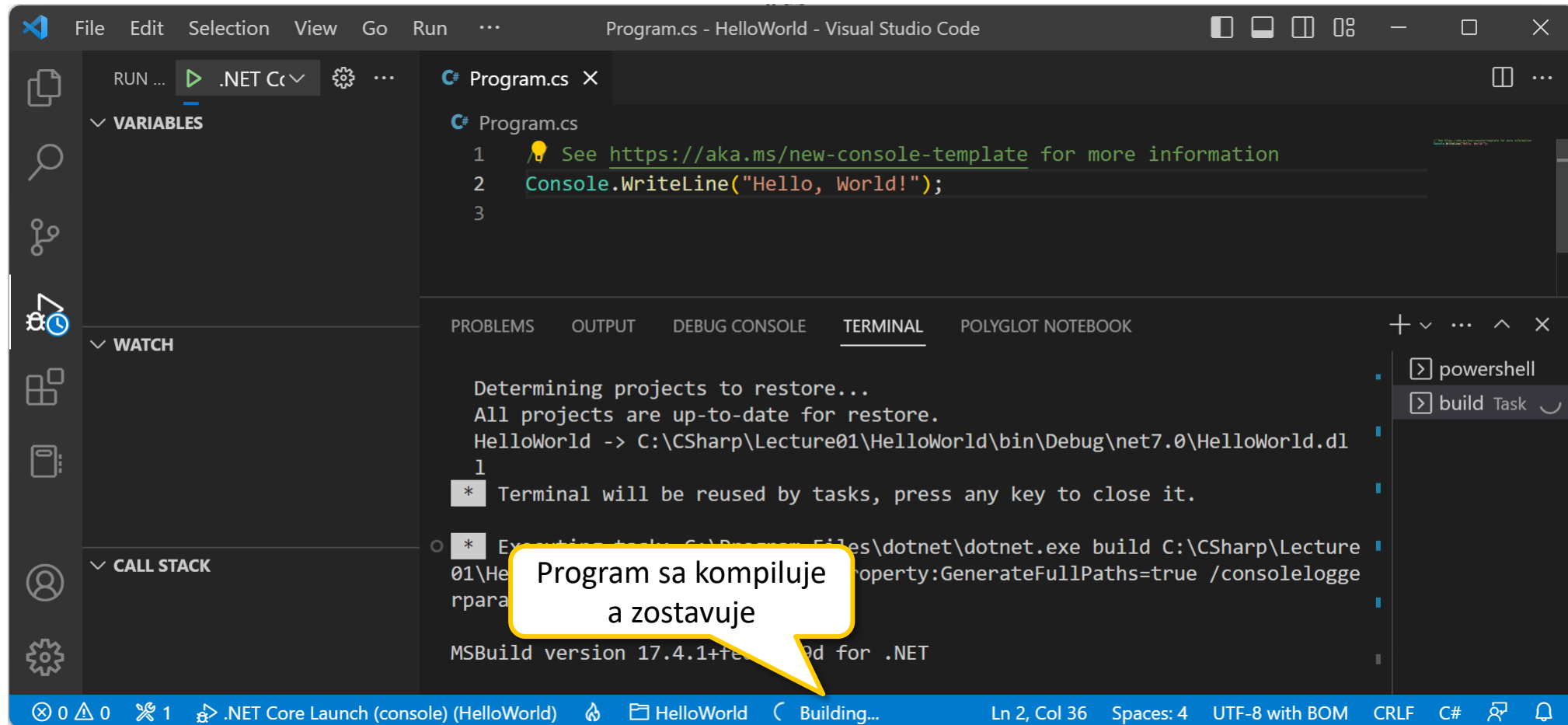
- Start Debugging (F5)
- Run Without Debugging (Ctrl+F5)
- Stop Debugging (Shift+F5)
- Restart Debugging (Ctrl+Shift+F5)
- Open Configurations
- Add Configuration...
- Step Over (F10)
- Step Into (F11)
- Step Out (Shift+F11)
- Continue (F5)
- Toggle Breakpoint (F9)
- New Breakpoint (>)
- Enable All Breakpoints
- Disable All Breakpoints

Annotations in the image:

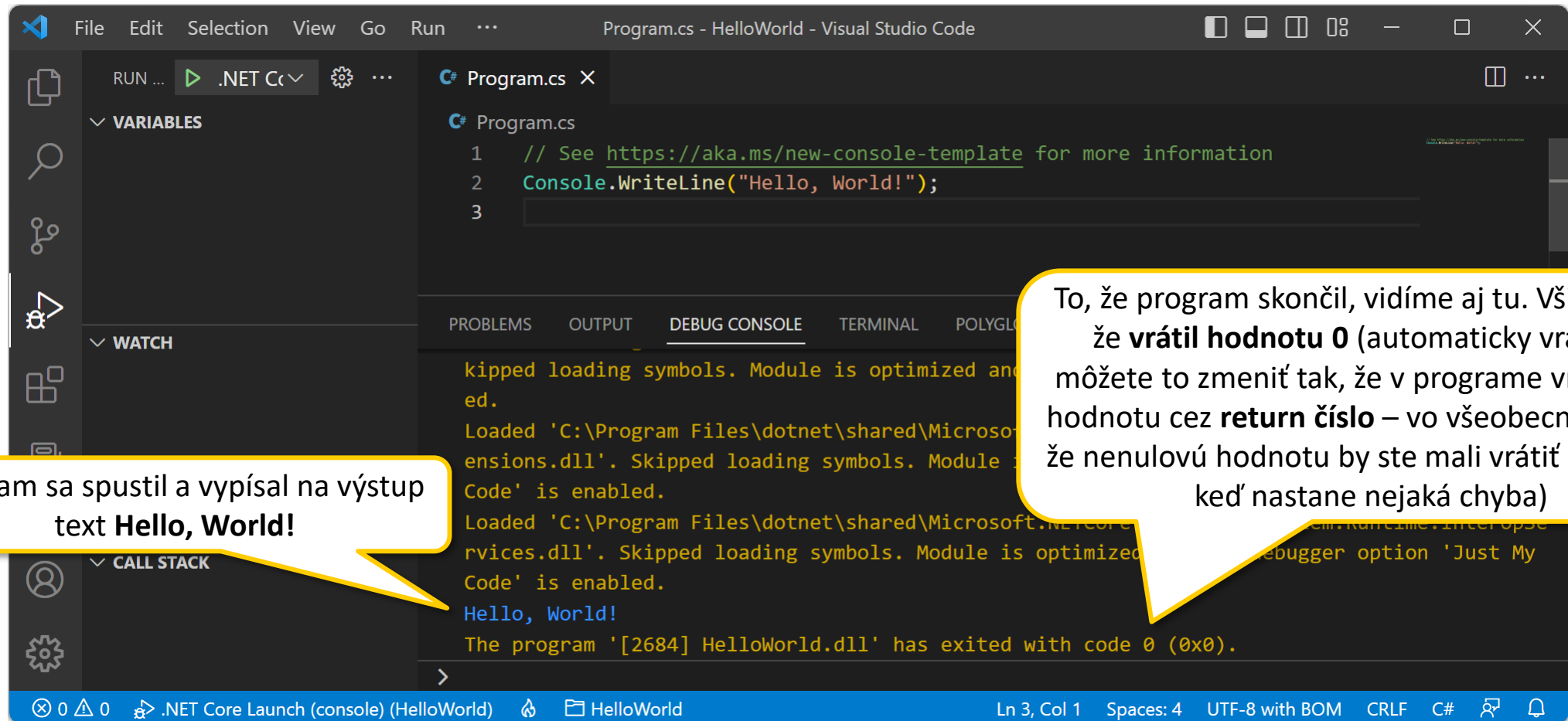
- Red circle '1' points to the 'Run' menu button in the top toolbar.
- Red circle '2' points to the 'Start Debugging' option in the Run menu.
- Red circle '1' points to the Run and Debug icon in the left sidebar.

Program môžeme **spustiť tlačidlom zelenej šípky** alebo pomocou **klávesovej skratky F5** (Start Debugging) alebo bez ladenia **Ctrl+F5** (Run Without Debugging).

# VS – spustenie programu – kompilácia

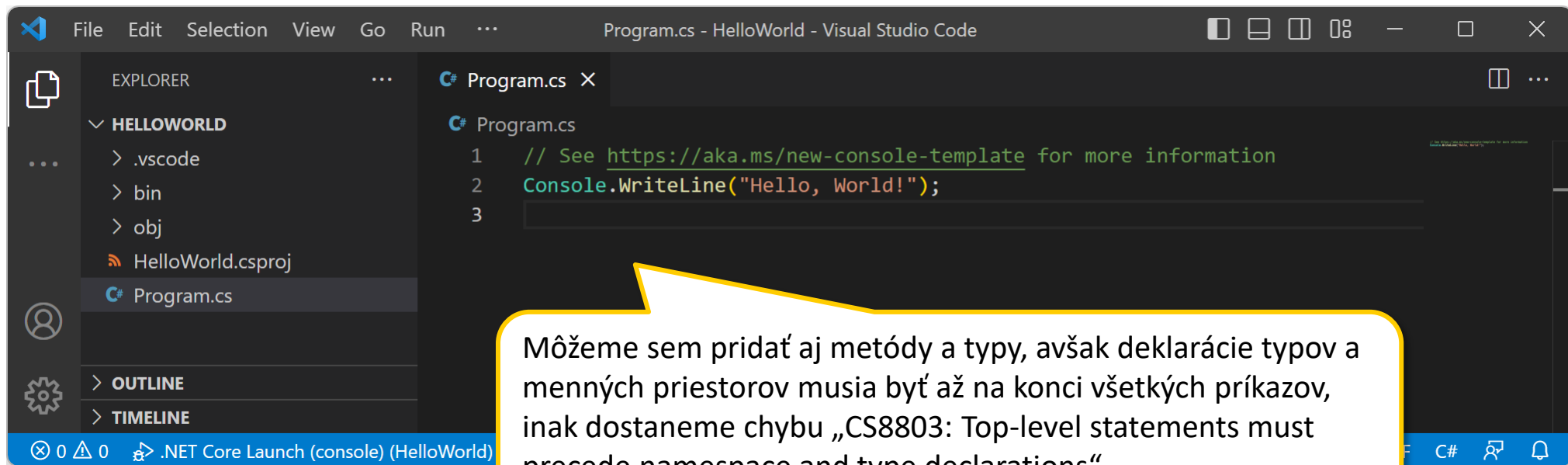


# VS Code – spustený program



# Hello World – od C# 10

- **Štandardná štruktúra programu od C# 10 (.NET 6) je s využitím príkazov na najvyššej úrovni („top-level statements“)**
  - nie sú potrebné menné priestory, triedy, metóda Main
  - **kompilátor sám vygeneruje triedu Program a metódu Main**, do ktorej vloží obsah zdrojového kódu:



# Hello World – od C# 10 (voľba --use-program-main)

```
C:\Windows\System32\cmd.e x + v
C:\CSharp\Lecture01>dotnet new console --output HelloWorldMain --use-program-main
The template "Console App" was created successfully.

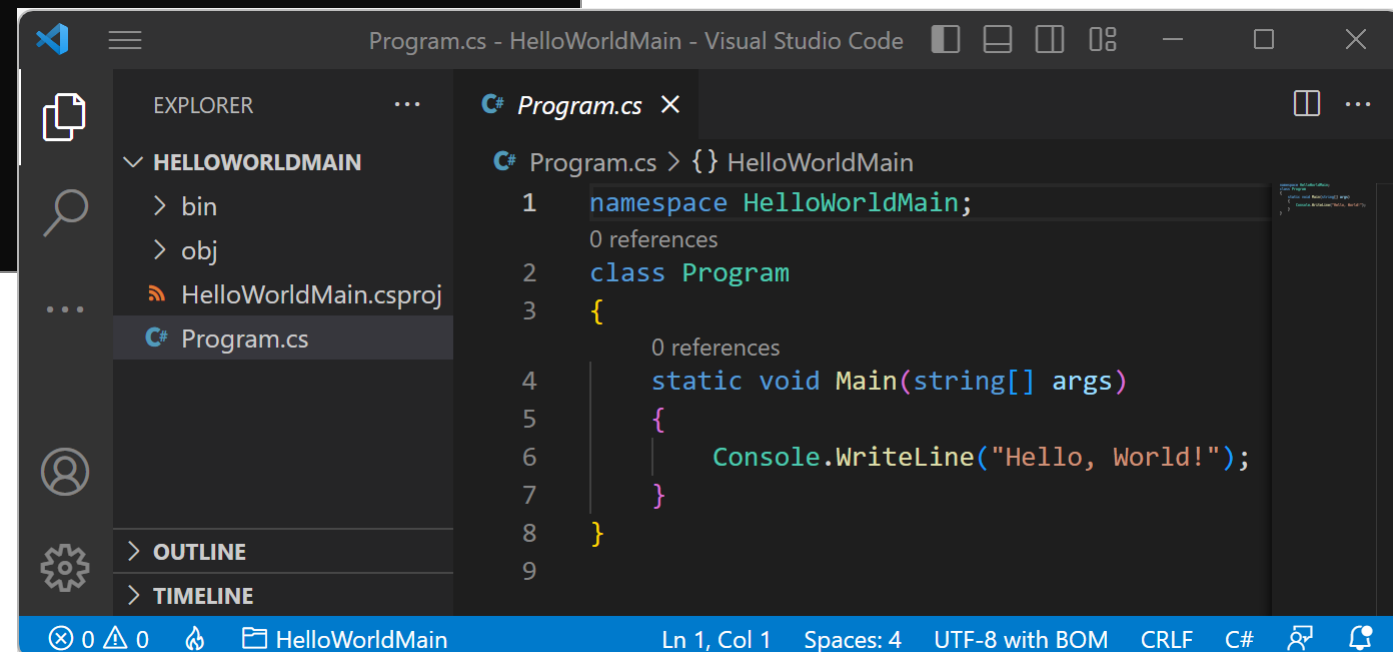
Processing post-creation actions...
Restoring C:\CSharp\Lecture01\HelloWorldMain\HelloWorldMain.csproj:
  Determining projects to restore...
  Restored C:\CSharp\Lecture01\HelloWorldMain\HelloWorldMain.csproj (in 77 ms).
Restore succeeded.

C:\CSharp\Lecture01>cd HelloWorldMain

C:\CSharp\Lecture01\HelloWorldMain>code .

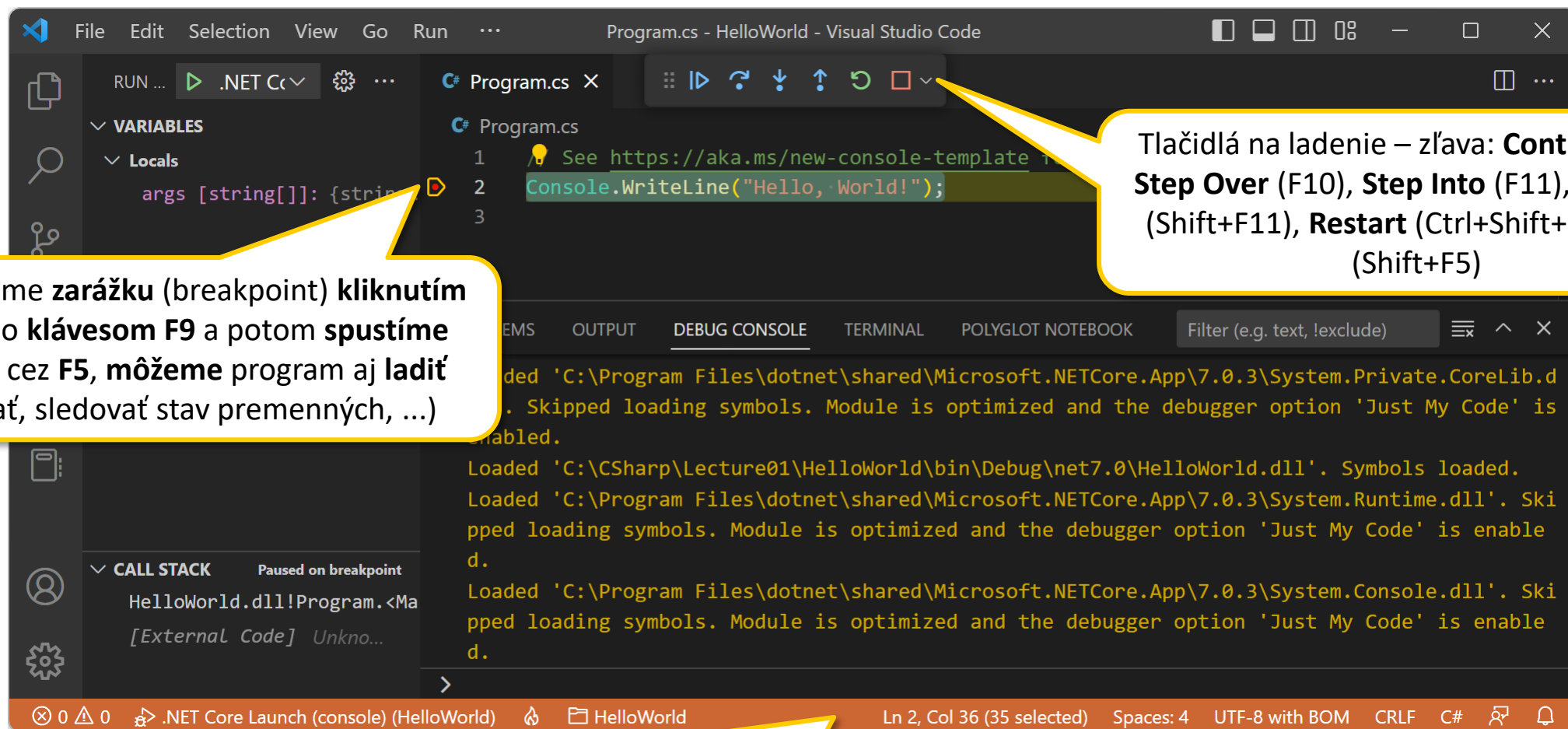
C:\CSharp\Lecture01\HelloWorldMain>
```

Od .NET SDK 6.0.300 môžeme pri vytváraní nového projektu použiť voľbu: **--use-program-main**, čím sa vytvorí štandardná trieda **Program** s metódou **Main()**, t. j. vygeneruje sa nám program bez použitia „top-level statements“



```
Program.cs - HelloWorldMain - Visual Studio Code
C# Program.cs x
C# Program.cs > {} HelloWorldMain
1 namespace HelloWorldMain;
  0 references
2 class Program
3 {
  0 references
4     static void Main(string[] args)
5     {
6         Console.WriteLine("Hello, World!");
7     }
8 }
9
```

# VS Code – zarážka a ladenie

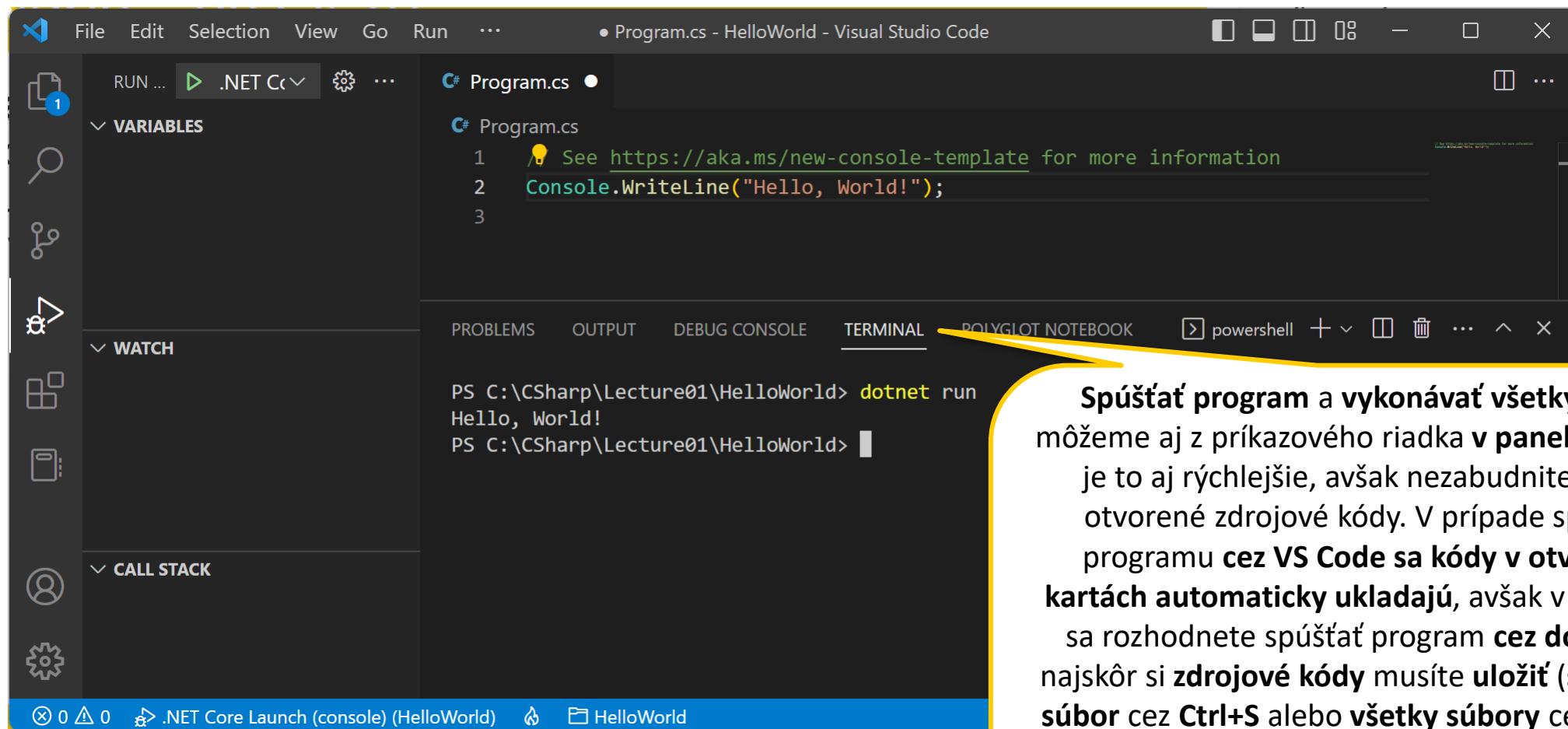


Ak si pridáme **zarážku** (breakpoint) **kliknutím sem** alebo **klávesom F9** a potom **spustíme program cez F5**, **môžeme program aj ladiť** (krokovat', sledovať stav premenných, ...)

Tlačidlá na ladenie – zľava: **Continue** (F5), **Step Over** (F10), **Step Into** (F11), **Step Out** (Shift+F11), **Restart** (Ctrl+Shift+F5), **Stop** (Shift+F5)

Môžete si všimnúť, že **pri ladení sa „status bar“ zmení na oranžovú farbu.**

# VS Code – príkazový riadok cez Terminal



**Spúšťať program a vykonávať všetky príkazy** môžeme aj z príkazového riadka v paneli **Terminal** – je to aj rýchlejšie, avšak nezabudnite si uložiť otvorené zdrojové kódy. V prípade spúšťania programu **cez VS Code sa kódy v otvorených kartách automaticky ukladajú**, avšak v prípade, že sa rozhodnete spúšťať program **cez dotnet run**, najskôr si **zdrojové kódy** musíte **uložiť (samostatný súbor cez Ctrl+S alebo všetky súbory cez Ctrl+K S)**. Alternatívne je možné použiť príkaz **dotnet watch**, ktorý sleduje zmeny a automaticky ich aktualizuje.

# Úloha 1.2 (HelloApp) – pozdrav podľa času

- **Upravte program** tak, aby **vypísal** nasledujúci **pozdrav podľa aktuálneho času**:
  - „Dobré ráno“, ak je čas od 4:00 do 8:00
  - „Dobrý deň“, ak je čas od 8:00 do 18:00
  - „Dobrý večer“, ak je čas od 18:00 do 22:00
  - „Dobrá noc“, ak je čas od 22:00 do 4:00
- **Poznámka:** Na získanie aktuálneho dátumu a času môžete použiť statickú vlastnosť **Now** štruktúry **DateTime**. Tá vráti naplnenú štruktúru **DateTime** s aktuálnym časom, následne môžete použiť inštančnú vlastnosť **Hour** (jej detailné použitie si pozrite v dokumentácii – learn – použite napr. Google: „c# datetime“, prípadne „c# datetime learn“)

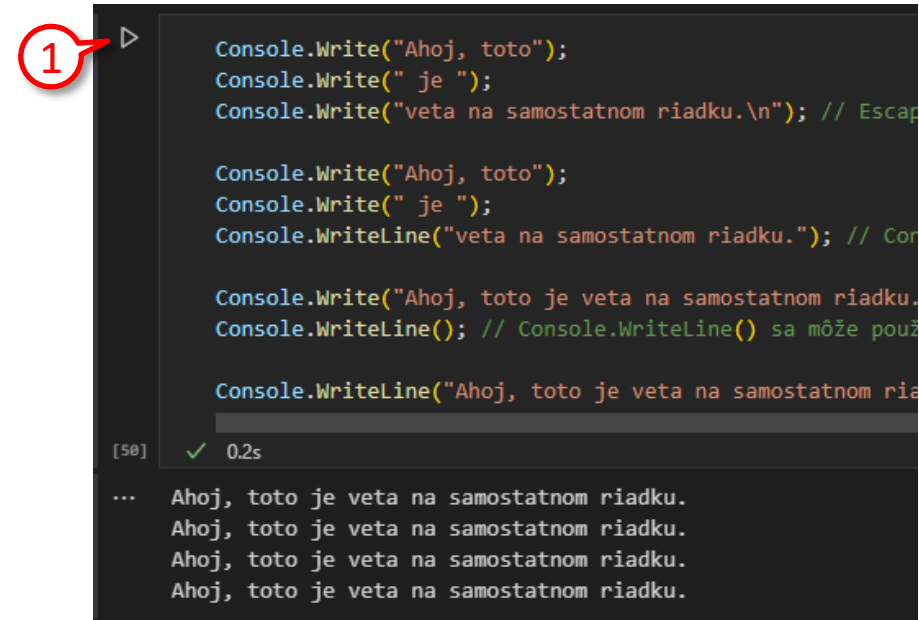


# Úloha 1.3 (HelloApp) – hádanie čísla

- Následne vytvorte „hru na hádanie čísla“ – používateľ musí uhádnuť číslo, pričom dostáva reakciu, či číslo je väčšie alebo menšie
- Program napíše: „Zahráme si hru na uhádnutie čísla. Myslím na číslo od 1..100, uhádneš na aké myslím?“
- Potom používateľ môže zadávať číslo, až dokiaľ ho neuhádne. Potom program vypíše: „Uhádol si! Celkový čas hľadania v sekundách: XX. Počet hádaní: YY. Chceš si zahrať hru ešte raz?“

# Úloha 2 (Notebooky)

- **Stiahnite si priložené notebooky, rozbaľte ich do priečinka a následne otvorte priečinok vo VS Code** (pozor: vo Visual Studiu 2022 sa spúšťať nedajú)
  - Aby Vám notebooky fungovali, budete potrebovať prvýkrát nainštalovať rozšírenie **Polyglot Notebooks** (ms-dotnettools.dotnet-interactive-vscode)
- Pozrite si notebooky a vyskúšajte prvý z nich: **00 Trieda Console.ipynb**
  - **Spustiť bunku** môžete pomocou šípky vľavo alebo klávesovou skratkou **Ctrl+Alt+Enter**



The screenshot shows a Jupyter Notebook cell in VS Code. A red circle with the number '1' and a red arrow points to the play button (a right-pointing triangle) in the top right corner of the cell. The cell contains C# code for writing to the console. Below the code, the output is displayed, showing the text 'Ahoj, toto je veta na samostatnom riadku.' repeated four times on separate lines. The output is preceded by '[50]' and a green checkmark, with a timing of '0.2s'.

```
Console.WriteLine("Ahoj, toto");
Console.WriteLine(" je ");
Console.WriteLine("veta na samostatnom riadku.\n"); // Escap

Console.WriteLine("Ahoj, toto");
Console.WriteLine(" je ");
Console.WriteLine("veta na samostatnom riadku."); // Cor

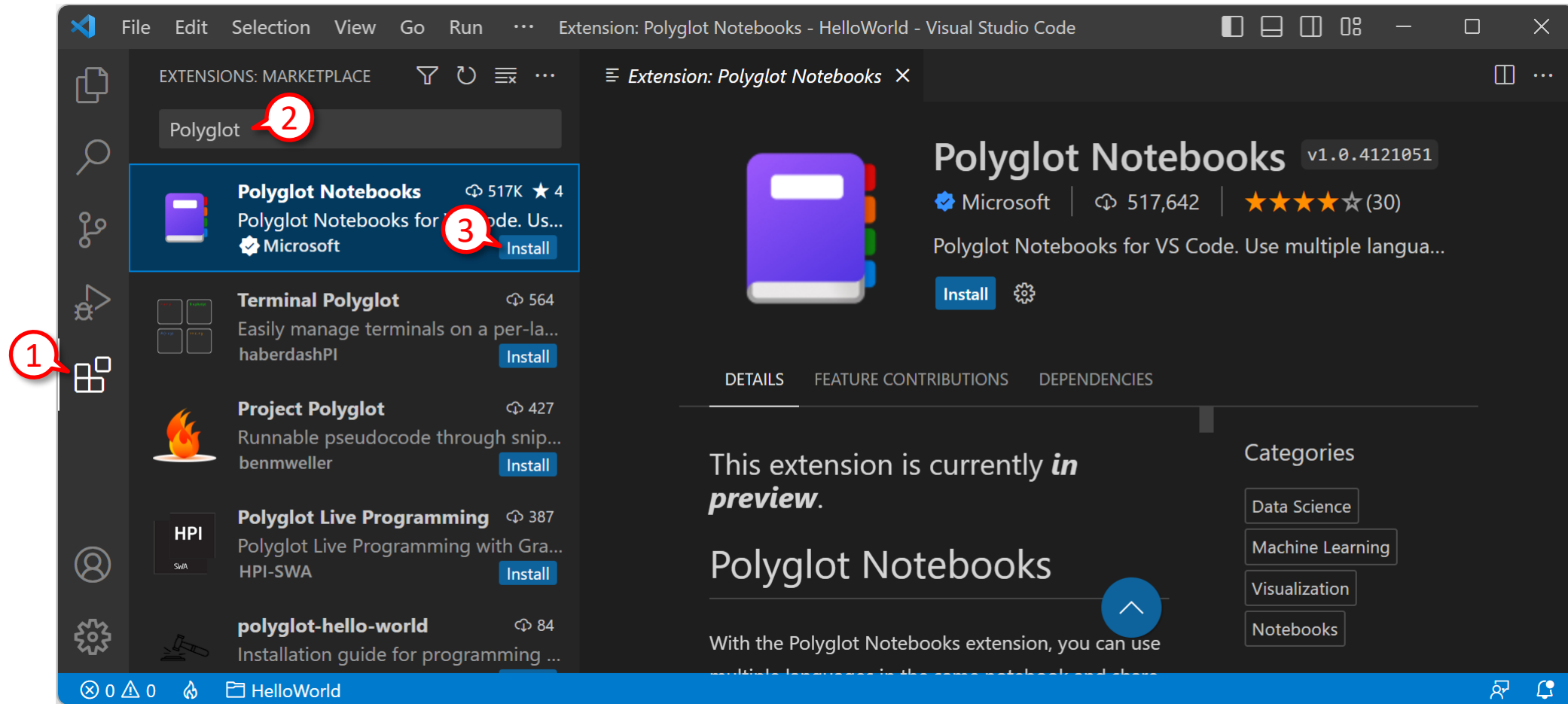
Console.WriteLine("Ahoj, toto je veta na samostatnom riadku.");
Console.WriteLine(); // Console.WriteLine() sa môže použiť

Console.WriteLine("Ahoj, toto je veta na samostatnom riadku.");
```

[50] ✓ 0.2s

... Ahoj, toto je veta na samostatnom riadku.  
Ahoj, toto je veta na samostatnom riadku.  
Ahoj, toto je veta na samostatnom riadku.  
Ahoj, toto je veta na samostatnom riadku.

# VS Code – Polyglot Notebooks



# Funkčnosť notebooku

- To, že Vám notebook funguje, zistíte tak, že sa správne deteguje kernel a zobrazí sa „.NET Interactive“
- Druhým znakom je to, že sa syntax C# kódu zvýrazní farbou

# Úloha 2.1 (Formátovanie)

- Otvorte si notebook **01 Úloha Formátovanie** a doprogramujte do neho postupne 3 naformátované výstupy

# Úloha 2.2 (AsciiTable)

- Otvorte si notebook **02 Úloha AsciiTable** a doprogramujte do neho **výpis ASCII tabuľky od znaku s decimálnou hodnotou 32 po znak s decimálnou hodnotou 126**, pričom **na každom riadku zobrazte reprezentáciu jedného znaku** ako:
  - **decimálne (desiatkové) číslo** na tri miesta (ak bude 1-ciferné číslo, doplnia sa 2 medzery, ak bude 2-ciferné číslo, doplní sa 1 medzera pred číslo)
  - **binárne (dvojkové) číslo** s naformátovaním na 7 miest
  - **oktálové (osmičkové) číslo** s naformátovaním na 3 miesta
  - **hexadecimálne (šestnástkové) číslo** začínajúce na 0x
  - **znak**

```
32 0100000 040 0x20
33 0100001 041 0x21 !
34 0100010 042 0x22 "
35 0100011 043 0x23 #
36 0100100 044 0x24 $
37 0100101 045 0x25 %
38 0100110 046 0x26 &
39 0100111 047 0x27 '
40 0101000 050 0x28 (
41 0101001 051 0x29 )
42 0101010 052 0x2A *
43 0101011 053 0x2B +
44 0101100 054 0x2C ,
45 0101101 055 0x2D -
46 0101110 056 0x2E .
47 0101111 057 0x2F /
48 0110000 060 0x30 0
49 0110001 061 0x31 1
50 0110010 062 0x32 2
51 0110011 063 0x33 3
52 0110100 064 0x34 4
53 0110101 065 0x35 5
54 0110110 066 0x36 6
55 0110111 067 0x37 7
56 0111000 070 0x38 8
...
123 1111011 173 0x7B {
124 1111100 174 0x7C |
125 1111101 175 0x7D }
126 1111110 176 0x7E ~
```

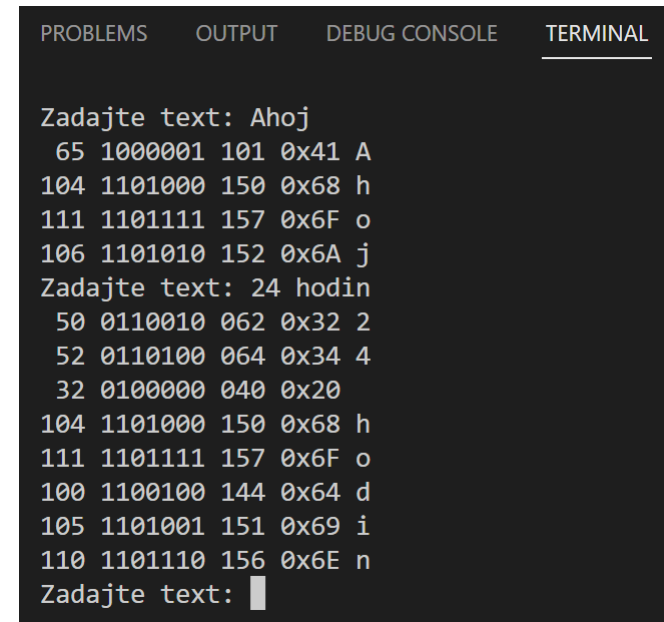
# Úloha 2.3 (Úloha MonthCalendar)

- Otvorte si notebook **03 Úloha MonthCalendar.ipynb** a **naprogramujte algoritmus**, ktorým vypíšete na výstup **kalendár** zobrazujúci dni v **zadanom mesiaci**, napr.:

```
marec 2023
po ut st št pi so ne
27 28  1  2  3  4  5
  6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31  1  2
```

# Úloha 3 (String2AsciiTable)

- Vytvorte novú konzolovú **aplikáciu** cez .NET CLI a VS Code, ktorá bude **na vstupe získavať text od používateľa** (vždy ako reťazec – riadok textu), pričom po jeho zadaní **pre každé písmeno vypíše naformátovaný riadok** ako v predchádzajúcej úlohe
- Program bude bežať neustále, až dokiaľ ho používateľ neukončí stlačením Ctrl+C



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Zadajte text: Ahoj
 65 1000001 101 0x41 A
104 1101000 150 0x68 h
111 1101111 157 0x6F o
106 1101010 152 0x6A j
Zadajte text: 24 hodin
 50 0110010 062 0x32 2
 52 0110100 064 0x34 4
 32 0100000 040 0x20
104 1101000 150 0x68 h
111 1101111 157 0x6F o
100 1100100 144 0x64 d
105 1101001 151 0x69 i
110 1101110 156 0x6E n
Zadajte text: █
```



# Zdroje

- .NET Documentation  
<https://learn.microsoft.com/en-us/dotnet/>
- C# documentation  
<https://learn.microsoft.com/en-us/dotnet/csharp/>
- Visual Studio product family documentation  
<https://learn.microsoft.com/en-us/visualstudio/>

# Upozornenie

- Tieto študijné materiály sú určené výhradne pre študentov predmetu **Jazyk C# a .NET** na Fakulte riadenia a informatiky Žilinskej univerzity v Žiline
- Reprodukovanie, šírenie (i častí) materiálov bez písomného súhlasu autora nie je dovolené



**ŽILINSKÁ UNIVERZITA V ŽILINE**  
Fakulta riadenia  
a informatiky

Ing. **Štefan Toth**, PhD.  
[stefan.toth@uniza.sk](mailto:stefan.toth@uniza.sk)