

Objektové programování I (C#)

RNDr. Michal Pobucký

michal.pobucky@fpf.slu.cz

Ústav informatiky – zima 2020

**SLEZSKÁ
UNIVERZITA**
FILOZOFICKO-
PŘÍRODOVĚDECKÁ
FAKULTA V OPAVĚ

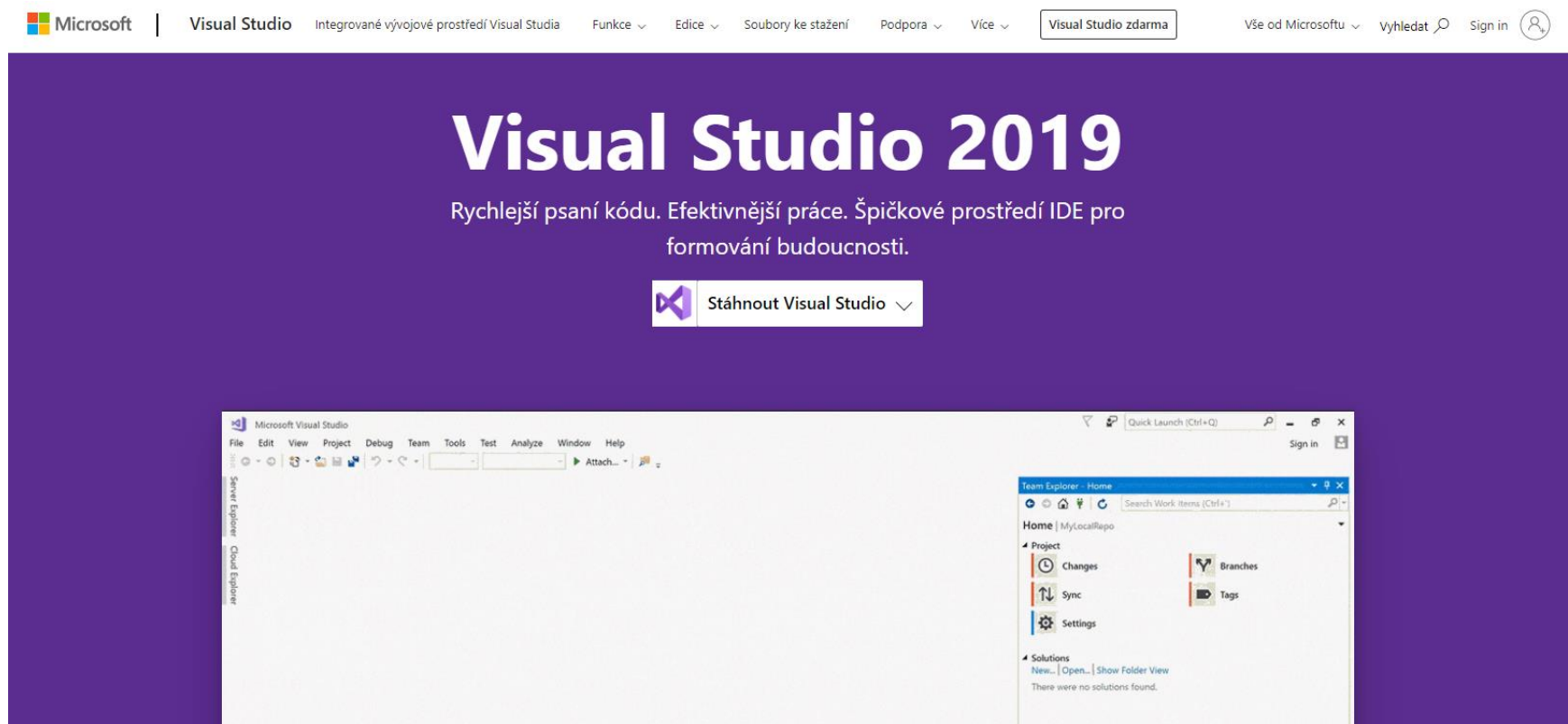


Podmínky absolvování předmětu

- Vytvoření aplikace – libovolný výběr
- Samostatná práce studenta – žádné kopírování z internetu

Visual Studio 2019


- <https://visualstudio.microsoft.com/cs/vs/>



Microsoft | Visual Studio Integrované vývojové prostředí Visual Studio Funkce ▾ Edice ▾ Soubory ke stažení Podpora ▾ Více ▾ Visual Studio zdarma Vše od Microsoftu ▾ Vyhledat 🔍 Sign in 👤

Visual Studio 2019

Rychlejší psaní kódu. Efektivnější práce. Špičkové prostředí IDE pro formování budoucnosti.

 Stáhnout Visual Studio ▾

Microsoft Visual Studio

File Edit View Project Debug Team Tools Test Analyze Window Help

Server Explorer Cloud Explorer

Team Explorer - Home

Search Work Items (Ctrl+I)

Home | MyLocalRepo

Project

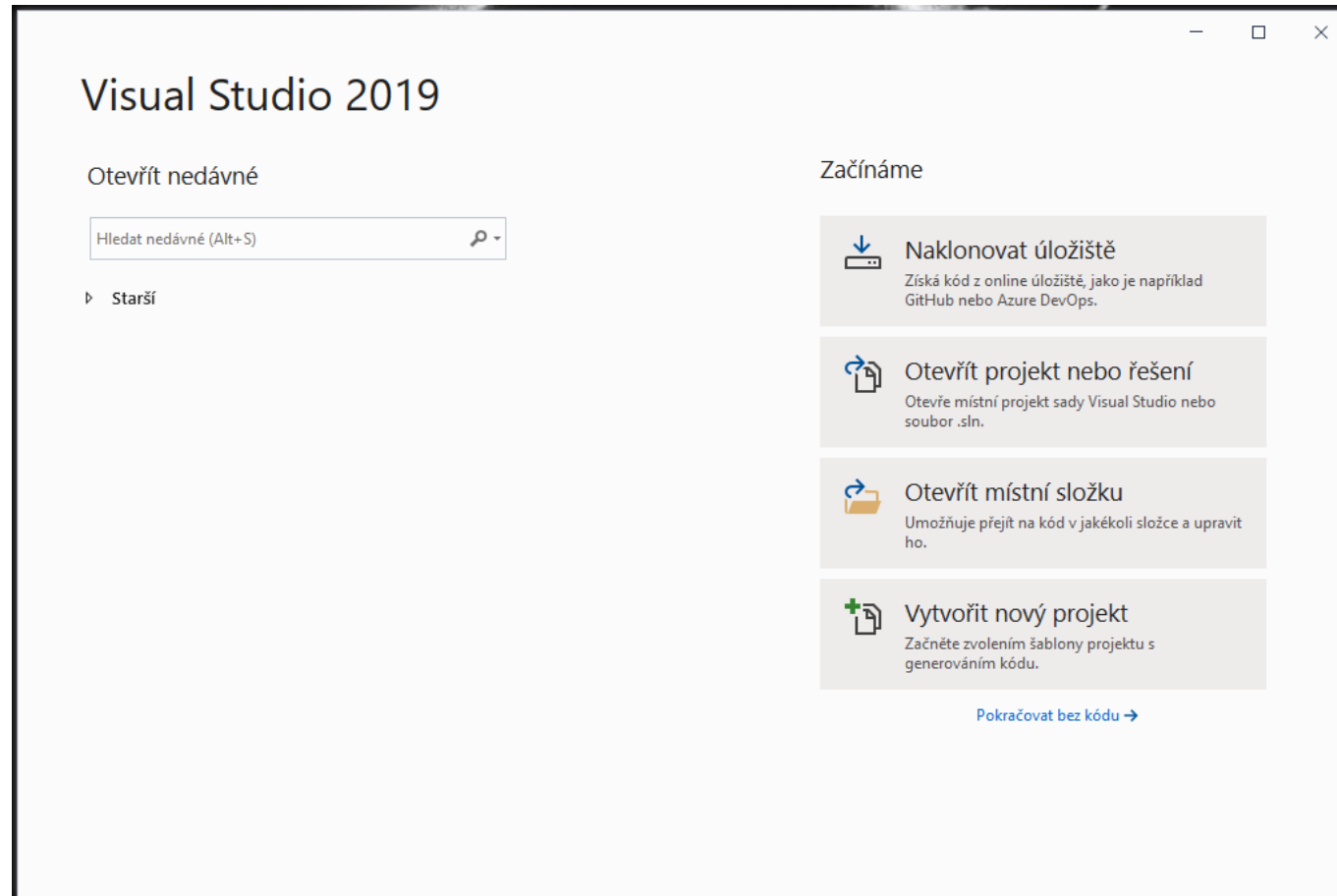
- Changes
- Sync
- Settings
- Branches
- Tags

Solutions

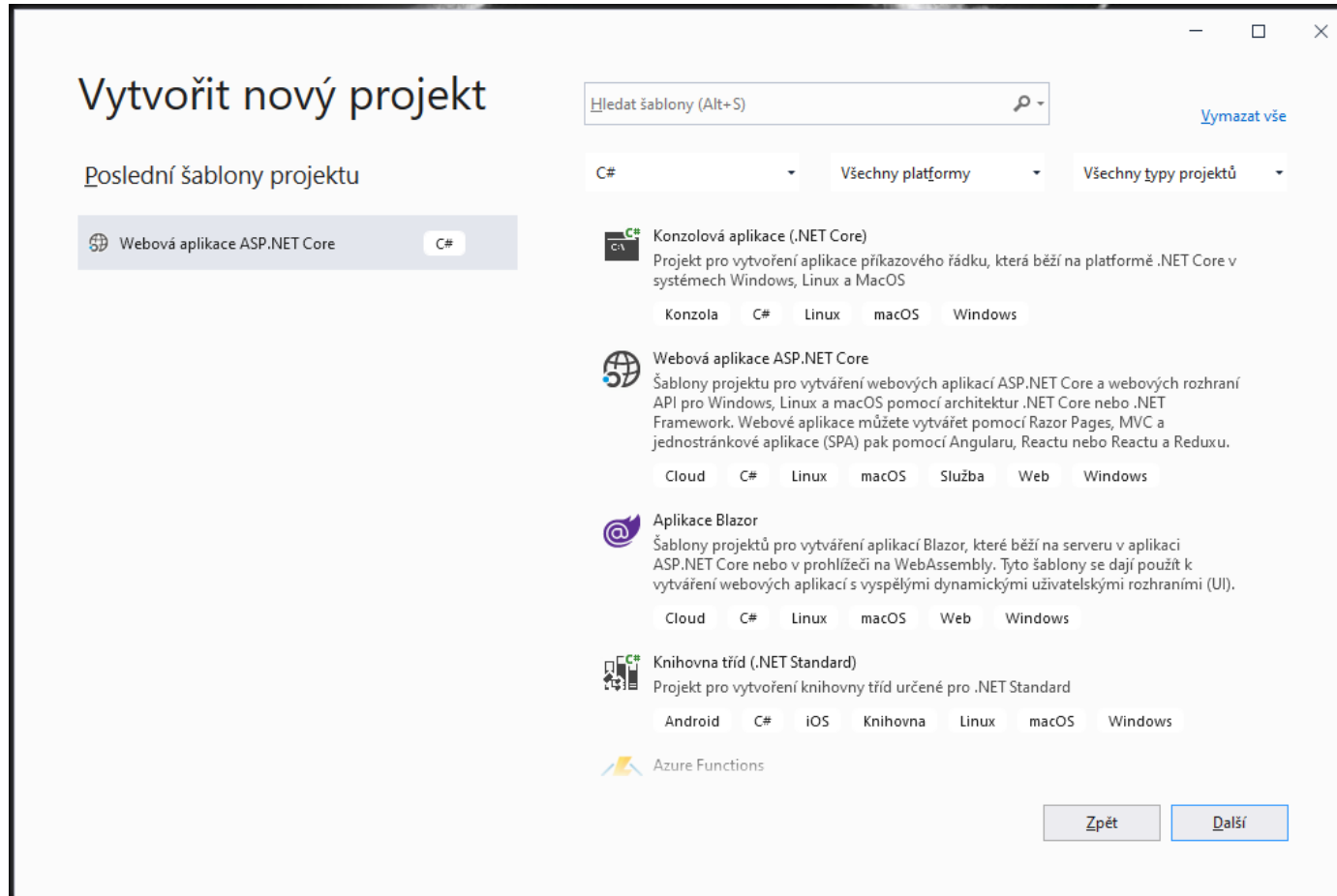
New... | Open... | Show Folder View

There were no solutions found.

Visual Studio 2019



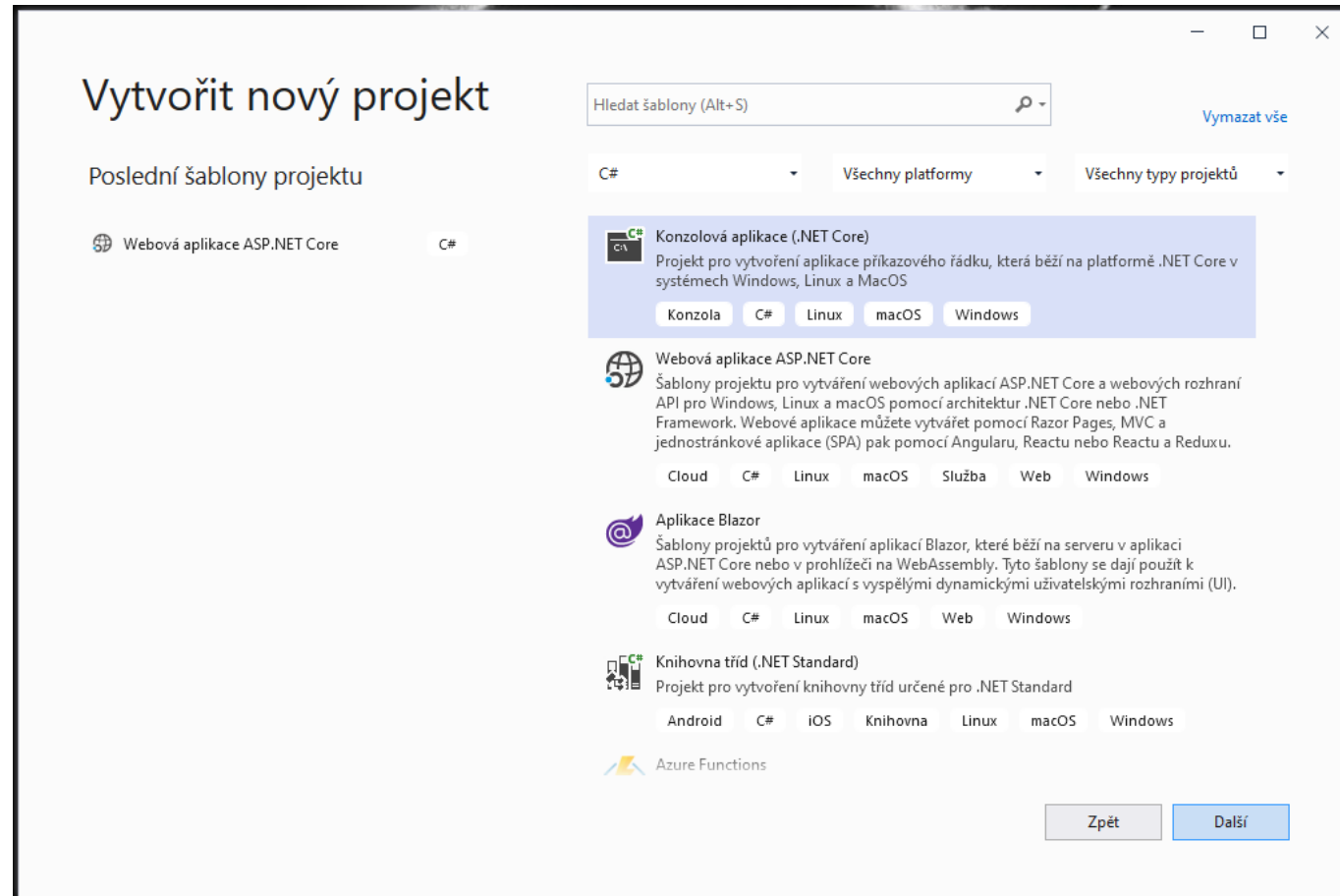
Visual Studio 2019



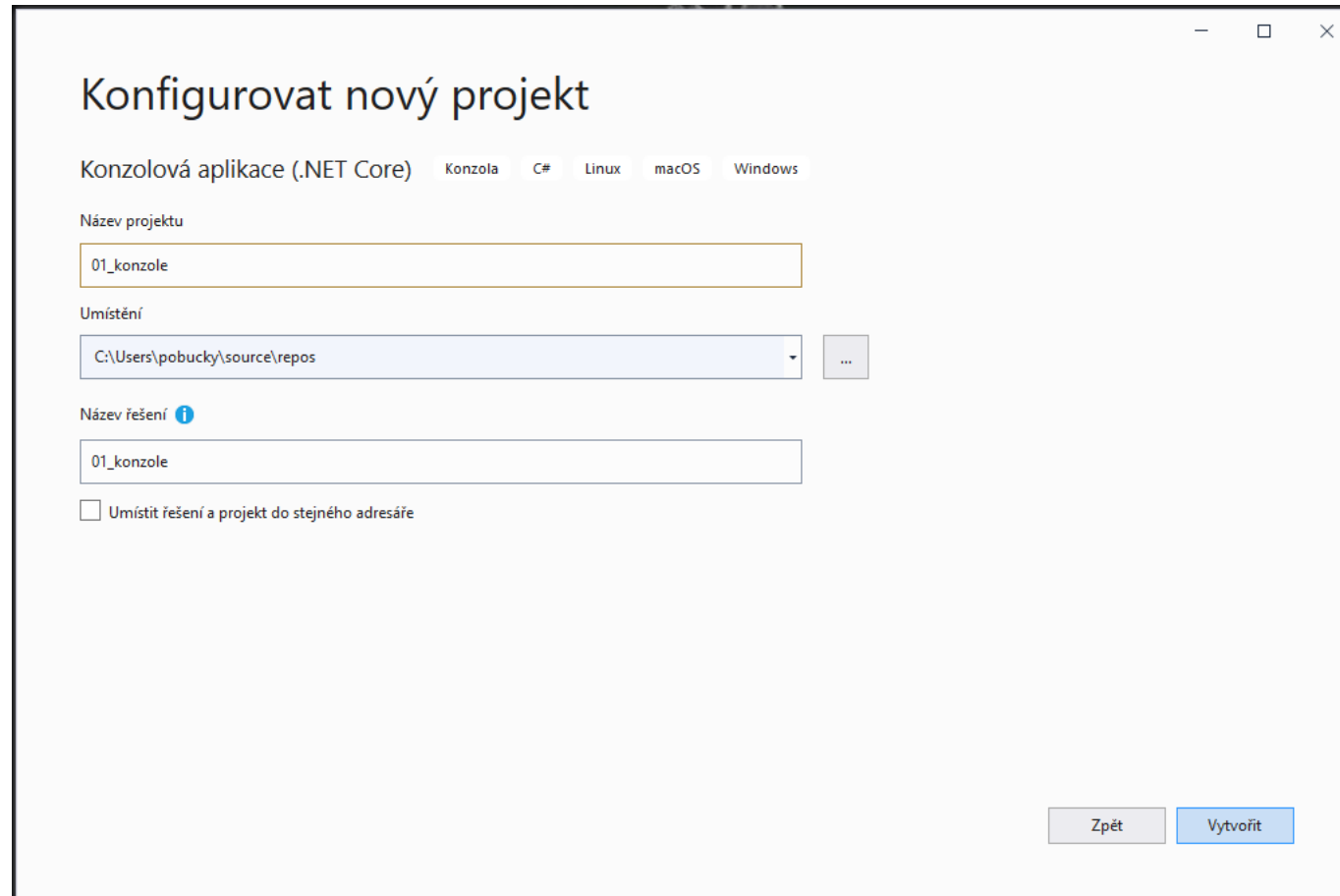
Visual Studio 2019

- Vytvořit nový projekt
 - Výběr podle jazyka (C#, C++, Java, Python, Visual Basic, ...)
 - Výběr podle platformy (Android, iOS, Linux, macOS, Windows, ...)
 - Výběr podle typu projektu (hry, konzole, test, web, ...)
- Vybereme C#
 - .NET Core x .NET Framework x .NET Standard
 - Konzolová aplikace, knihovna tříd, aplikace Win Forms, aplikace WPF, ...

Visual Studio 2019



Visual Studio 2019



Konfigurovat nový projekt

Konzolová aplikace (.NET Core) Konzola C# Linux macOS Windows

Název projektu

01_konzole

Umístění

C:\Users\pobucky\source\repos ...

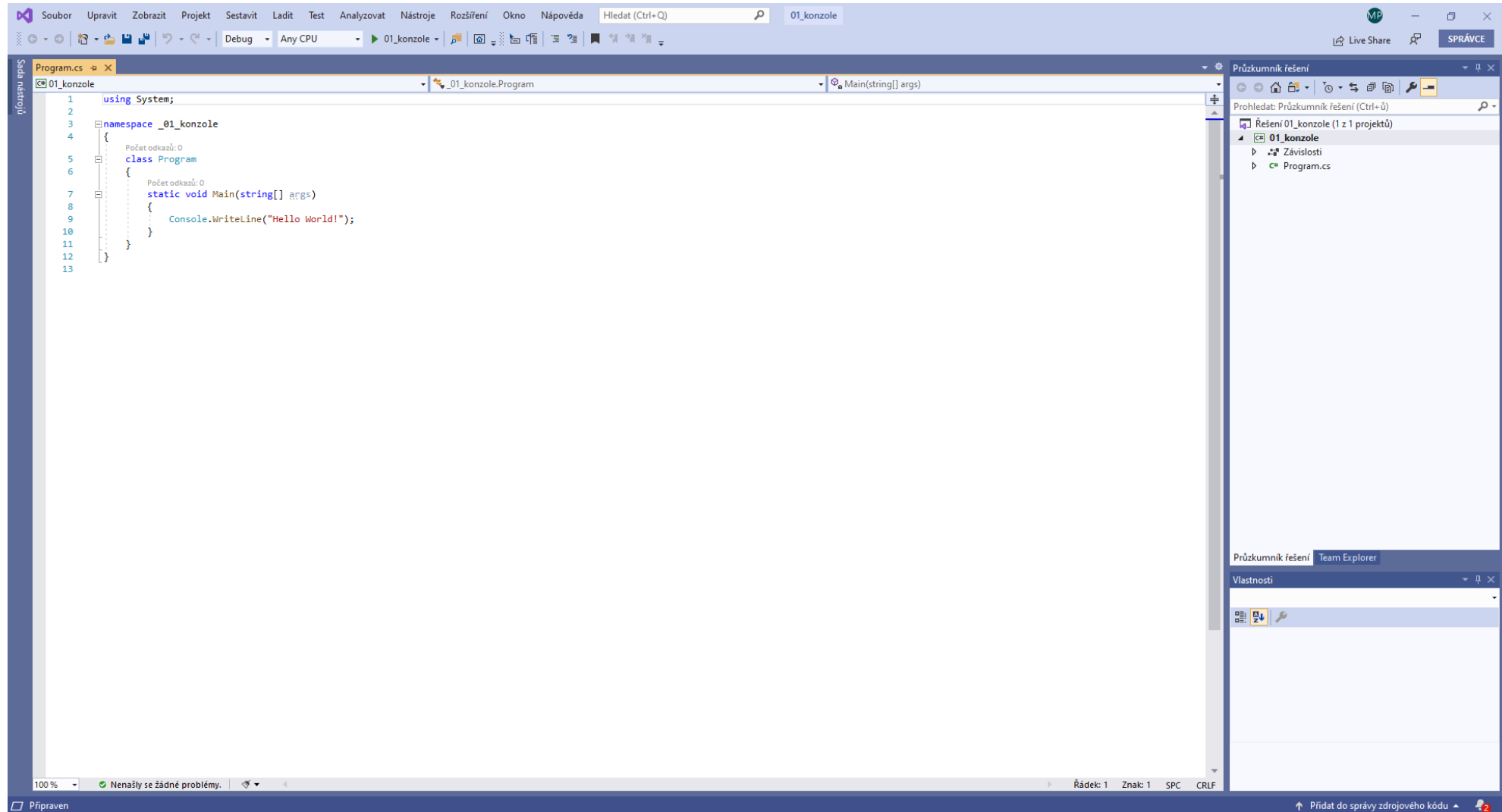
Název řešení ⓘ

01_konzole

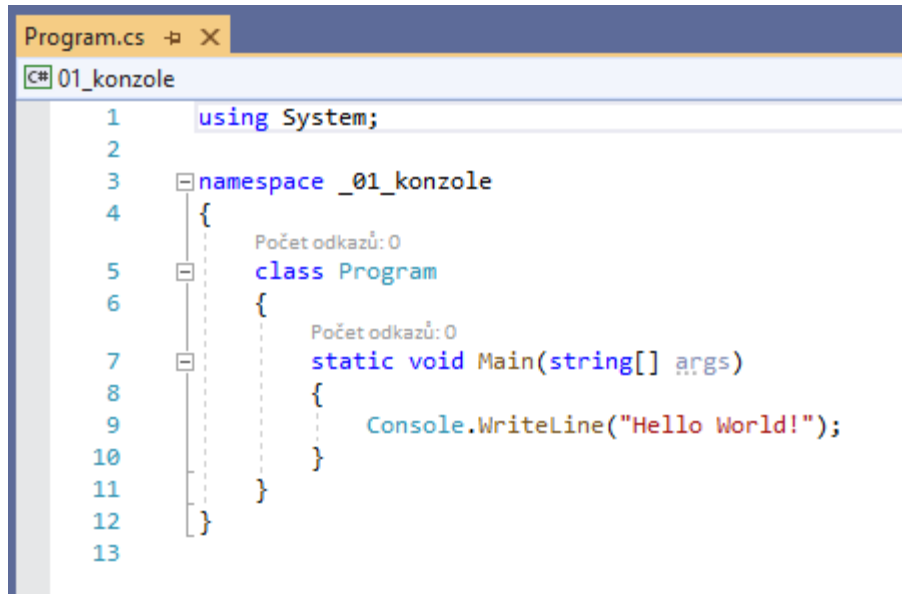
Umístit řešení a projekt do stejného adresáře

Zpět Vytvořit

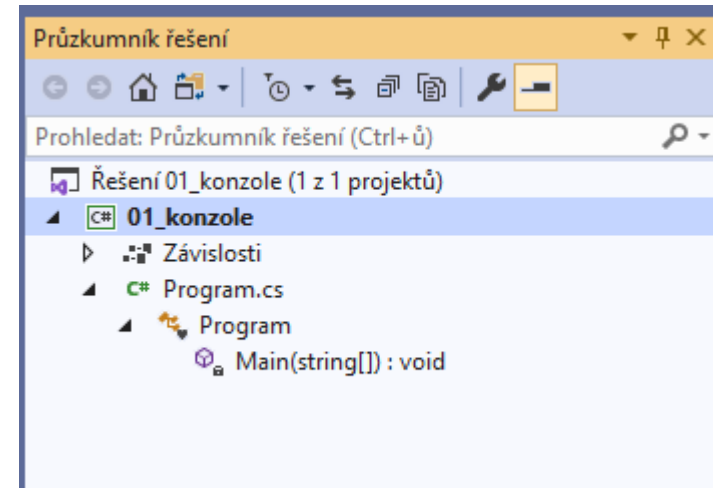
Visual Studio 2019



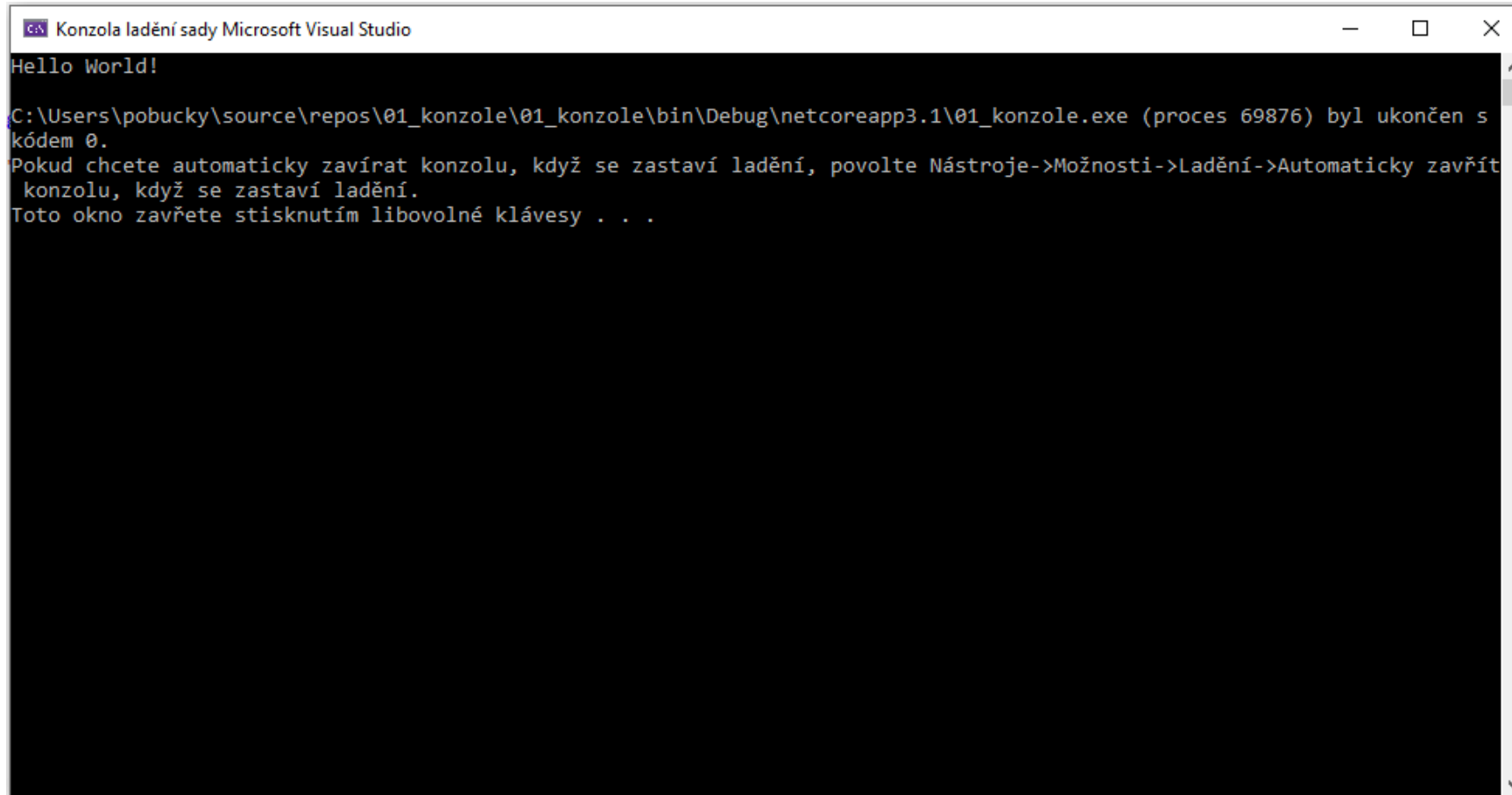
Visual Studio 2019



```
Program.cs [X]
C# 01_konzole
1 using System;
2
3 namespace _01_konzole
4 {
5     class Program
6     {
7         static void Main(string[] args)
8         {
9             Console.WriteLine("Hello World!");
10        }
11    }
12 }
13
```



Visual Studio 2019



```
Konzola ladění sady Microsoft Visual Studio
Hello World!
C:\Users\pobucky\source\repos\01_konzole\01_konzole\bin\Debug\netcoreapp3.1\01_konzole.exe (proces 69876) byl ukončen s kódem 0.
Pokud chcete automaticky zavírat konzolu, když se zastaví ladění, povolte Nástroje->Možnosti->Ladění->Automaticky zavřít konzolu, když se zastaví ladění.
Toto okno zavřete stisknutím libovolné klávesy . . .
```

Základy - proměnné

```
string cast1 = "Otázka života";  
string cast2 = "a dál";  
int kolik = 42;  
double a = 1, b = 2.5, c = 11.3;  
  
cast2 = ", vesmíru a vůbec.";  
a += 1.23;  
  
var otazka = ''' + cast1 +  
    cast2 + "\\n Jaká je odpověď na tuto otázku?";  
var odpoved = "Odpověď je " + kolik + ".";  
  
Console.WriteLine(otazka);  
Console.WriteLine(odpoved);  
Console.WriteLine(a + b + c);
```

Základy - proměnné

Characteristics of the integral types

C# supports the following predefined integral types:

C# type/keyword	Range	Size	.NET type
<code>sbyte</code>	-128 to 127	Signed 8-bit integer	System.SByte
<code>byte</code>	0 to 255	Unsigned 8-bit integer	System.Byte
<code>short</code>	-32,768 to 32,767	Signed 16-bit integer	System.Int16
<code>ushort</code>	0 to 65,535	Unsigned 16-bit integer	System.UInt16
<code>int</code>	-2,147,483,648 to 2,147,483,647	Signed 32-bit integer	System.Int32
<code>uint</code>	0 to 4,294,967,295	Unsigned 32-bit integer	System.UInt32
<code>long</code>	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	Signed 64-bit integer	System.Int64
<code>ulong</code>	0 to 18,446,744,073,709,551,615	Unsigned 64-bit integer	System.UInt64

Základy - proměnné

Characteristics of the floating-point types

C# supports the following predefined floating-point types:

C# type/keyword	Approximate range	Precision	Size	.NET type
<code>float</code>	$\pm 1.5 \times 10^{-45}$ to $\pm 3.4 \times 10^{38}$	~6-9 digits	4 bytes	System.Single
<code>double</code>	$\pm 5.0 \times 10^{-324}$ to $\pm 1.7 \times 10^{308}$	~15-17 digits	8 bytes	System.Double
<code>decimal</code>	$\pm 1.0 \times 10^{-28}$ to $\pm 7.9228 \times 10^{28}$	28-29 digits	16 bytes	System.Decimal

char (C# reference)

11. 05. 2020 • 2 min ke čtení • 🌐 📄

The `char` type keyword is an alias for the .NET [System.Char](#) structure type that represents a Unicode UTF-16 character.

Type	Range	Size	.NET type
<code>char</code>	U+0000 to U+FFFF	16 bit	System.Char

bool (C# reference)

26. 11. 2019 • 2 min ke čtení • 🌐 📄

The `bool` type keyword is an alias for the .NET [System.Boolean](#) structure type that represents a Boolean value, which can be either `true` or `false`.

Základy – proměnné – konverze a formátování

```
int cislo;  
double desetinneCislo = 6.0;  
cislo = (int)desetinneCislo;  
  
char[] znaky = { 'a', 'u', (char) 0x074, 'o' };  
string slovo = new string(znaky);  
Console.WriteLine(slovo);
```

```
string jmeno = "Karel";  
int vek = 6;  
string zprava = string.Format("{0} má {1} let.", jmeno, vek);  
Console.WriteLine(zprava);
```

```
zprava = $"{jmeno} nemá {vek} let.";  
Console.WriteLine(zprava);
```

```
zprava = $"Vážilo to přesně {vek:f2} kilogramů.";  
Console.WriteLine(zprava);
```

Základy - výrazy

```
c *= -1;  
double x = (-b + Math.Sqrt(b * b - 4 * a * c)) / (2 * a);  
Console.WriteLine(x);
```

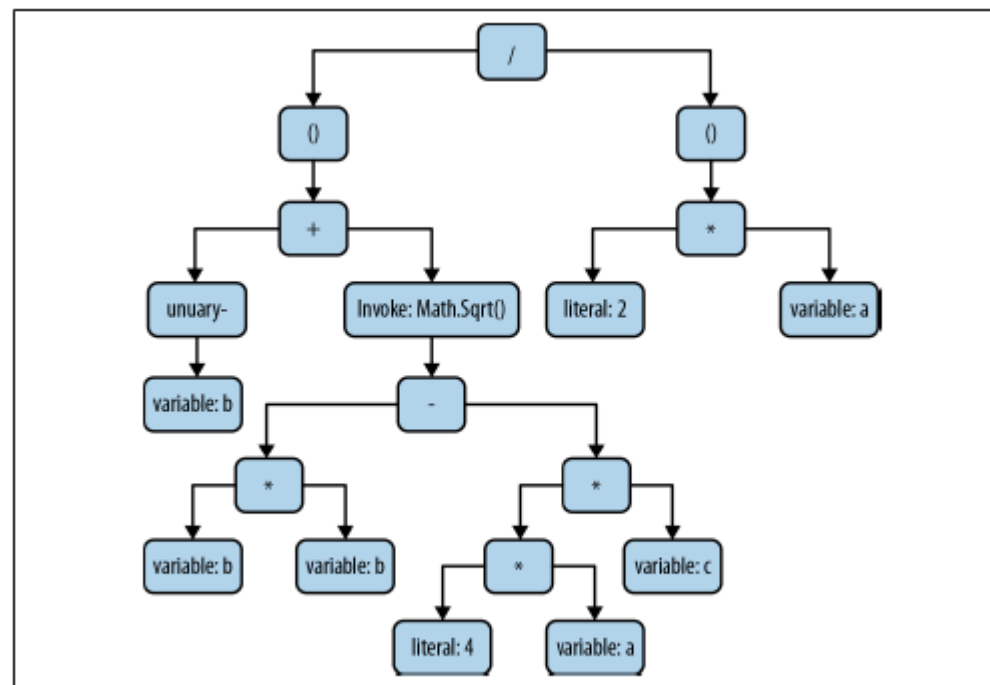


Figure 2-1. The structure of an expression

Griffiths 2019

Základy - komentáře

```
// komentář  
Console.WriteLine("první"); // komentář  
Console.WriteLine("druhý"); /*  
Console.WriteLine("třetí"); /*  
Console.WriteLine("čtvrtý"); */  
Console.WriteLine("pátý");
```

```
#region ČÁST KÓDU  
int q = 15;  
q += 17;  
#endregion
```



Základy - tuples

```
(int X, int Y) point = (10, 5);  
Console.WriteLine($"Bod má souřadnice X: {point.X} a Y: {point.Y}");  
  
var point2 = (X: 10, Y: 5);  
  
int X = 10, Y = 5;  
var point3 = (X, Y);  
Console.WriteLine($"Bod má souřadnice X: {point2.X} a Y: {point2.Y}");  
Console.WriteLine($"Bod má souřadnice X: {point3.X} a Y: {point3.Y}");  
  
(int, int) point4 = (10, 5);  
Console.WriteLine($"Bod má souřadnice X: {point4.Item1} a Y: {point4.Item2}");
```

Základy - operátory

Table 2-3. Basic arithmetic operators

Name	Example
Unary plus (does nothing)	+x
Negation (unary minus)	-x
Postincrement	x++
Postdecrement	x--
Preincrement	++x
Predecrement	--x
Addition	x + y
Subtraction	x - y
Multiplication	x * y
Division	x / y
Remainder	x % y

Table 2-4. Binary integer operators

Name	Example
Bitwise negation	~x
Bitwise AND	x & y
Bitwise OR	x y
Bitwise XOR	x ^ y
Shift left	x << y
Shift right	x >> y

Table 2-5. Operators for bool

Name	Example
Logical negation (also known as NOT)	!x
Conditional AND	x && y
Conditional OR	x y

Table 2-6. Relational operators

Name	Example
Less than	x < y
Greater than	x > y
Less than or equal	x <= y
Greater than or equal	x >= y
Equal	x == y
Not equal	x != y

Griffiths 2019

```
int maximum = (X > Y) ? X : Y;  
string s = "";  
int pocetZnaku = s == null ? 0 : s.Length;  
Console.WriteLine(pocetZnaku);
```

Základy – podmíněné příkazy

```
vek = 25;
if (vek > 18)
{
    Console.WriteLine("Jž jsi dávno plnoletý.");
}
else if (vek == 18)
{
    Console.WriteLine("Gratuluji, právě jsi dosáhl plnoletosti.");
}
else
{
    Console.WriteLine("Ještě ti není 18 let.");
}
```

```
if (point.X > 5)
{
    if (point2.X > 5)
    {
        if (point3.X > 5)
        {
            Console.WriteLine("Všechny tři body mají x-ovou sořadnici větší než 5.");
        }
    }
}
```

```
if (point.X > 5 && point2.X > 5 && point3.X > 5)
{
    Console.WriteLine("Všechny tři body mají x-ovou sořadnici větší než 5.");
}
```

Základy – podmíněné příkazy – switch – case

```
Console.Write("Zadej materiál: ");
string vstup = Console.ReadLine();
switch (vstup)
{
    case "dřevo":
        Console.WriteLine("Dřeva máme dostatek.");
        break;
    case "voda":
    case "led":
        Console.WriteLine("Vody máme dostatek v jakémkoliv skuceství.");
        break;
    case "železo":
        Console.WriteLine("Nic železného nemáme");
        break;
    default:
        Console.WriteLine("Musíš zadat jeden z našich materiálů.");
        break;
}
```

Základy – cykly – for

```
for (int i = 0; i < 10; i++)  
{  
    Console.WriteLine(i);  
}
```

```
for (int i = 0; i <= 10; i++)  
{  
    Console.WriteLine(i);  
}
```

```
for (int i = 0; i == 10; i++)  
{  
    Console.WriteLine(i);  
}
```

```
for (; ; )  
{  
    Console.WriteLine("?");  
}
```

```
for (int i = 0; i < 10; i++)  
{  
    for (int j = 0; j < 10; j++)  
    {  
        Console.Write(i);  
        Console.Write(j);  
        Console.Write(" ");  
    }  
    Console.WriteLine();  
}
```

Základy – cykly – do – while

```
char klavesa;  
do  
{  
    Console.WriteLine("Stiskni klávesu x, aby program pokračoval...");  
    klavesa = Console.ReadKey().KeyChar;  
} while (klavesa != 'x');
```

```
while (vek > 1)  
{  
    Console.WriteLine(vek);  
    vek /= 2;  
}  
Console.WriteLine(vek);
```

```
while (true)  
{  
    Console.WriteLine(vek);  
    vek *= 2;  
}  
Console.WriteLine(vek);
```

```
while (true)  
{  
    Console.WriteLine(vek);  
    vek *= 2;  
    if (vek == 0) break;  
}  
Console.WriteLine(vek);
```

Základy – cykly – foreach

```
foreach (char znak in cast1)
{
    Console.WriteLine(znak);
}
```


Zdroje

- <https://docs.microsoft.com/cs-cz/dotnet/csharp/language-reference/builtin-types/integral-numeric-types>
- <https://docs.microsoft.com/cs-cz/dotnet/csharp/language-reference/builtin-types/floating-point-numeric-types>
- <https://docs.microsoft.com/cs-cz/dotnet/csharp/language-reference/builtin-types/char>
- <https://docs.microsoft.com/cs-cz/dotnet/csharp/language-reference/builtin-types/bool>
- Griffiths, Ian. *Programming C# 8.0. Build Cloud, Web, and Desktop Applications*. O'Reilly Media, Inc. 2019, ISBN: 9781492056812