

Objektové programování I (C#) – 3

RNDr. Michal Pobucký

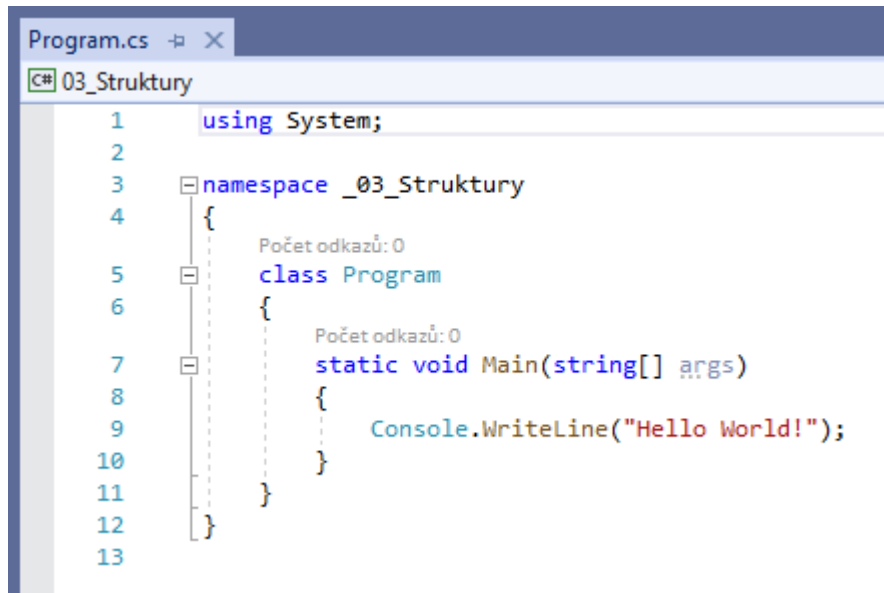
michal.pobucky@fpf.slu.cz

Ústav informatiky – zima 2020

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



- Vytvořte konzolovou aplikaci (.NET Core), název projektu 03_Struktury



```
Program.cs  ▸ ×
C# 03_Struktury
1  using System;
2
3  namespace _03_Struktury
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              Console.WriteLine("Hello World!");
10         }
11     }
12 }
13
```

Výjimky

Počet odkazů: 0

```
static void Main(string[] args)
{
    int a = 5, b = 0, c;

    c = a / b;

    Console.WriteLine(c);
}
```

Co se stane a proč?

Výjimky

```
Počet odkazů: 0
static void Main(string[] args)
{
    int a = 5, b = 0, c;

    c = a / b;

    Console.WriteLine(c);
}
```

Co se stane a proč?

```
1 using System;
2
3 namespace _03_Struktury
4 {
5     class Program
6     {
7         static void Main(string[] args)
8         {
9             int a = 5, b = 0, c;
10
11             c = a / b;
12
13             Console.WriteLine(c);
14         }
15     }
16 }
17
```

Neošetřená výjimka

System.DivideByZeroException: Attempted to divide by zero.

[Zobrazit podrobnosti](#) | [Kopírovat podrobnosti](#) | [Spustit relaci Live Share](#)

▸ [Nastavení výjimek](#)

Výjimky

```
Počet odkazů: 0
class Program
{
    Počet odkazů: 0
    static void Main(string[] args)
    {
        int a = 5, b = 0, c;

        try
        {
            c = a / b;
        }
        catch
        {
            Console.WriteLine("CHYBA!");
            c = 0;
        }
        Console.WriteLine(c);
    }
}
```

```
Počet odkazů: 0
class Program
{
    Počet odkazů: 0
    static void Main(string[] args)
    {
        int a = 5, b = 1, c;

        try
        {
            c = a / b;
        }
        catch
        {
            Console.WriteLine("CHYBA!");
            c = 0;
        }
        finally
        {
            Console.WriteLine("hotovo");
        }
        Console.WriteLine(c);
    }
}
```

Výjimky

Počet odkazů: 0

```
static void Main(string[] args)
{
    int a = 5, b = 0, c;
    string retezec = "0";

    try
    {
        int cislo = int.Parse(retezec);
        c = a / cislo;
    }
    catch (DivideByZeroException)
    {
        Console.WriteLine("Dělení nulou nepovoleno!");
        c = 0;
    }
    finally
    {
        Console.WriteLine("hotovo");
    }
    Console.WriteLine(c);
}
```

Výjimky

Co se stane a proč?

Počet odkazů: 0

```
static void Main(string[] args)
{
    int a = 5, b = 0, c;
    string retezec = "112A";

    try
    {
        int cislo = int.Parse(retezec);
        c = a / cislo;
    }
    catch (DivideByZeroException)
    {
        Console.WriteLine("Dělení nulou nepovoleno!");
        c = 0;
    }
    finally
    {
        Console.WriteLine("hotovo");
    }
    Console.WriteLine(c);
}
```

Výjimky

Co se stane a proč?

Počet odkazů: 0

```
static void Main(string[] args)
{
    int a = 5, b = 0, c;
    string retezec = "112A";

    try
    {
        int cislo = int.Parse(retezec);
        c = a / cislo;
    }
    catch (DivideByZeroException)
    {
        Console.WriteLine("Dělení nulou nepovoleno!");
        c = 0;
    }
    finally
    {
        Console.WriteLine("hotovo");
    }
    Console.WriteLine(c);
}
```

Počet odkazů: 0

```
static void Main(string[] args)
{
    int a = 5, b = 0, c;
    string retezec = "112A";

    try
    {
        int cislo = int.Parse(retezec);
        c = a / cislo;
    }
    catch (DivideByZeroException)
    {
        Console.WriteLine("Dělení nulou");
        c = 0;
    }
    finally
    {
    }
}
```

Neošetřená výjimka

System.FormatException: Input string was not in a correct format.

[Zobrazit podrobnosti](#) | [Kopírovat podrobnosti](#) | [Spustit relaci Live Share](#)

▸ [Nastavení výjimek](#)

Výjimky

```
Počet odkazů: 0
class Program
{
    Počet odkazů: 0
    static void Main(string[] args)
    {
        int a = 5, b = 0, c = 0;
        string retezec = "112A";

        try
        {
            int cislo = int.Parse(retezec);
            c = a / cislo;
        }
        catch (DivideByZeroException)
        {
            Console.WriteLine("Dělení nulou nepovoleno!");
        }
        catch (FormatException)
        {
            Console.WriteLine("Řetězec není číslo!");
        }
        finally
        {
            Console.WriteLine("hotovo");
        }
        Console.WriteLine(c);
    }
}
```



Výčet

```
namespace _03_Struktury
{
    Počet odkazů: 2
    enum RocniObdobi { Jaro, Leto, Podzim, Zima };

    Počet odkazů: 0
    class Program
    {
```

```
RocniObdobi aktualni = RocniObdobi.Podzim;
Console.WriteLine("Období je " + aktualni);
Console.WriteLine("Období je " + (int)aktualni);
```

Výčet

```
namespace _03_Struktury
{
    Počet odkazů: 2
    enum RocniObdobi { Jaro, Leto, Podzim, Zima };

    Počet odkazů: 0
    class Program
    {

```

```
RocniObdobi aktualni = RocniObdobi.Podzim;
Console.WriteLine("Období je " + aktualni);
Console.WriteLine("Období je " + (int)aktualni);
```

```
aktualni++;
Console.WriteLine(aktualni);
```

Co vypíše?

Struktury

Hodnotový typ x referenční typ



```
Počet odkazů: 1
struct Cas
{
    private int hodiny, minuty, sekundy;

    Počet odkazů: 0
    public Cas(int hh, int mm, int sec)
    {
        hodiny = hh;
        minuty = mm;
        sekundy = sec;
    }

    Počet odkazů: 0
    public int Hodiny()
    {
        return hodiny;
    }
}
```

Rozdíl oproti třídám:

- Není možné deklarovat implicitní konstruktor (bez parametrů)

```
Počet odkazů: 0
public Cas()
{
    ...
}
```

- Není možné inicializovat datové složky přímo v deklaraci

```
private int hodiny = 0;
private int minuty, sekundy;
```

Struktury

Počet odkazů: 1

```
struct Cas
{
    private int hodiny, minuty, sekundy;

    Počet odkazů: 0
    public Cas(int hh, int mm, int sec)
    {
        hodiny = hh;
        minuty = mm;
        sekundy = sec;
    }

    Počet odkazů: 0
    public int Hodiny()
    {
        return hodiny;
    }
}
```

```
Cas nyní = new Cas(12, 3, 5);
Console.WriteLine(nyni.Hodiny());
```

Struktury

Počet odkazů: 1

```
struct Cas
{
    private int hodiny, minuty, sekundy;

    Počet odkazů: 0
    public Cas(int hh, int mm, int sec)
    {
        hodiny = hh;
        minuty = mm;
        sekundy = sec;
    }

    Počet odkazů: 0
    public int Hodiny()
    {
        return hodiny;
    }
}
```

```
Cas nyni = new Cas(12, 3, 5);
Console.WriteLine(nyni.Hodiny());
```

```
nyni = new Cas();
Console.WriteLine(nyni.Hodiny());
```

Co se stane a proč?

Struktury

```
Počet odkazů: 5
struct Cas
{
    private int hodiny, minuty, sekundy;

    Počet odkazů: 1
    public Cas(int hh, int mm, int sec)
    {
        hodiny = hh;
        minuty = mm;
        sekundy = sec;
    }

    Počet odkazů: 3
    public int Hodiny()
    {
        return hodiny;
    }

    Počet odkazů: 0
    public void PridejHodinu()
    {
        hodiny++;
    }
}
```

```
Cas kopie = nyní;
Console.WriteLine("Kopie :" + kopie.Hodiny());
kopie.PridejHodinu();
Console.WriteLine("Kopie :" + kopie.Hodiny());
Console.WriteLine("Originál :" + nyní.Hodiny());
```

Co se stane a proč?

Pole

```
int[] císla;           // deklarace proměnné typu pole
císla = new int[3];   // vytvoření instance pole o velikosti 3 prvků (statické)

císla = new int[4] { 1, 2, 3, 4 }; // statické

Console.WriteLine("Zadej velikost pole: ");
int velikost = int.Parse(Console.ReadLine());
císla = new int[velikost]; // dynamické

Random rand = new Random();
for (int i = 0; i < velikost; i++)
{
    císla[i] = rand.Next();
}
foreach(int hodnota in císla)
{
    Console.WriteLine(hodnota);
}
```


Pole

```
int[][] matice = new int[3][];
for (int i = 0; i < matice.Length; i++)
{
    matice[i] = new int[3];
    for (int j = 0; j < matice[i].Length; j++)
    {
        matice[i][j] = i + j;
    }
}

foreach(int[] radek in matice)
{
    foreach(int cislo in radek)
    {
        Console.Write(cislo);
    }
    Console.WriteLine();
}
```

Co vypíše?

Pole

```
int[][] matice = new int[3][];  
for (int i = 0; i < matice.Length; i++)  
{  
    matice[i] = new int[3];  
    for (int j = 0; j < matice[i].Length; j++)  
    {  
        matice[i][j] = i + j;  
    }  
}  
  
foreach(int[] radek in matice)  
{  
    foreach(int cislo in radek)  
    {  
        Console.Write(cislo);  
    }  
    Console.WriteLine();  
}
```

Co vypíše?

Jak můžeme vytvořit kopii pole?

Pole

```
int[][] matice = new int[3][];
for (int i = 0; i < matice.Length; i++)
{
    matice[i] = new int[3];
    for (int j = 0; j < matice[i].Length; j++)
    {
        matice[i][j] = i + j;
    }
}

foreach(int[] radek in matice)
{
    foreach(int cislo in radek)
    {
        Console.Write(cislo);
    }
    Console.WriteLine();
}
```

Co vypíše?

Jak můžeme vytvořit kopii pole?

Další možnosti...

```
int[] kopiePole = new int[cisla.Length];

cisla.CopyTo(kopiePole, 0);
Array.Copy(cisla, kopiePole, cisla.Length);
kopiePole = (int[])cisla.Clone();
```

Kolekce

```
ArrayList seznam = new ArrayList();
seznam.Add(5);
seznam.Add("Nic");
seznam.Add(3.1415);
seznam.Insert(2, "Vloženo");
seznam.RemoveAt(1);

foreach (object o in seznam) {
    Console.WriteLine(o);
}
```

```
Queue fifo = new Queue();
fifo.Enqueue(1);
fifo.Enqueue("Dva");
fifo.Enqueue(155);
Console.WriteLine(fifo.Dequeue());
Console.WriteLine(fifo.Dequeue());
Console.WriteLine(fifo.Dequeue());
Console.WriteLine(fifo.Count);
```

```
Stack lifo = new Stack();
lifo.Push(1);
lifo.Push("Dva");
lifo.Push(155);
Console.WriteLine(lifo.Pop());
Console.WriteLine(lifo.Pop());
Console.WriteLine(lifo.Pop());
Console.WriteLine(lifo.Count);
```

Rozhraní

- Chceme-li seřadit obsahy kolekcí podle jejich obsahu, potřebujeme nástroj pro porovnání jejich obsahu
- Například když budeme mít metodu `int Porovnej(object obj)`, můžeme pomocí ní seřadit obsah kolekce
- Můžeme definovat rozhraní pro objekty, které obsahují metodu `Porovnej`

```
29 |  
30 | Počet odkazů: 0  
31 | interface IPorovnatelny  
32 | {  
33 | Počet odkazů: 0  
34 |     int Porovnej(object obj);  
   | }
```

Rozhraní

```
Počet odkazů: 1
interface ISuchozemsky
{
    Počet odkazů: 0
    int PocetNohou();
}

Počet odkazů: 1
class Savec
{
}

Počet odkazů: 0
class Kun : Savec, ISuchozemsky
{
}
```

Kód	Popis
CS0535	'Kun neimplementuje člen rozhraní ISuchozemsky.PocetNohou().

```
Počet odkazů: 0
class Kun : Savec, ISuchozemsky
{
    Počet odkazů: 1
    public int PocetNohou()
    {
        return 4;
    }
}
```

Rozhraní

Počet odkazů: 1

```
interface ISuchozemsky
{
    Počet odkazů: 2
    int PocetNohou();
}
```

Počet odkazů: 1

```
interface ICesta
{
    Počet odkazů: 2
    int PocetNohou();
}
```

Počet odkazů: 1

```
class Savec
{
}
```

Počet odkazů: 0

```
class Kun : Savec, ISuchozemsky, ICesta
{
    Počet odkazů: 2
    public int PocetNohou()
    {
        return 4;
    }
}
```

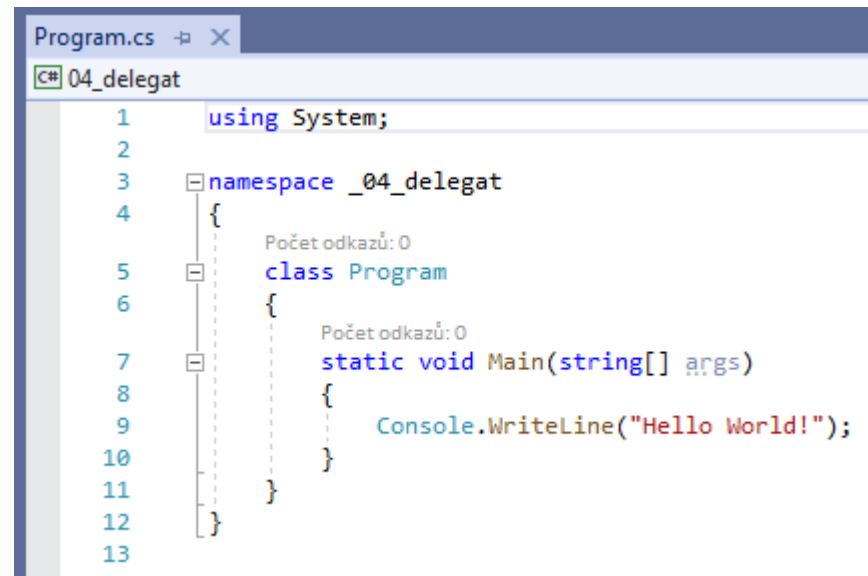
Počet odkazů: 0

```
class Kun : Savec, ISuchozemsky, ICesta
{
    Počet odkazů: 1
    int ISuchozemsky.PocetNohou()
    {
        return 4;
    }

    Počet odkazů: 1
    int ICesta.PocetNohou()
    {
        return 10;
    }
}
```

Delegát

- Delegát odkazuje na metodu
- Uvozen klíčovým slovem delegate
- Vytvořte novou konzolovou aplikaci 04_delegat



```
Program.cs [X]
C# 04_delegat
1 using System;
2
3 namespace _04_delegat
4 {
5     class Program
6     {
7         static void Main(string[] args)
8         {
9             Console.WriteLine("Hello World!");
10        }
11    }
12 }
13
```


Delegát

```
Program.cs + X
C# 04_delegat
1 using System;
2
3 namespace _04_delegat
4 {
5     public delegate void Funkce();
6     class Program
7     {
8         public static void MojeFunkce()
9         {
10             Console.WriteLine("MojeFunkce je spuštěna.");
11         }
12         static void Main(string[] args)
13         {
14             Funkce funkce = new Funkce(MojeFunkce);
15             funkce();
16         }
17     }
18 }
19
```

Spustit

Delegát

```
Program.cs  + X
C# 04_delegat
1  using System;
2
3  namespace _04_delegat
4  {
5      public delegate void Funkce();
6      class Program
7      {
8          public static void MojeFunkce()
9          {
10             Console.WriteLine("MojeFunkce je spuštěna.");
11         }
12
13         public static void MojeFunkce2()
14         {
15             Console.WriteLine("MojeFunkce2 je taky spuštěna.");
16         }
17
18         static void Main(string[] args)
19         {
20             Funkce funkce = new Funkce(MojeFunkce);
21             funkce();
22
23             funkce = null;
24             funkce += new Funkce(MojeFunkce);
25             funkce += new Funkce(MojeFunkce2);
26             funkce();
27         }
28     }
29 }
30
```

Spustit

Delegát

```
Program.cs  + X
C# 04_delegat  _04_del

1  using System;
2
3  namespace _04_delegat
4  {
5      public delegate void Funkce();
6      class Program
7      {
8          public static void MojeFunkce()
9          {
10             Console.WriteLine("MojeFunkce je spuštěna.");
11         }
12
13         public static void MojeFunkce2()
14         {
15             Console.WriteLine("MojeFunkce2 je taky spuštěna.");
16         }
17
18         static void Main(string[] args)
19         {
20             Funkce funkce = new Funkce(MojeFunkce);
21             funkce();
22
23             funkce = null;
24             funkce += new Funkce(MojeFunkce);
25             funkce += new Funkce(MojeFunkce2);
26             funkce -= new Funkce(MojeFunkce);
27             funkce();
28         }
29     }
30 }
31
```

Co se stane?

Spustit