

1

Základy jazyka C# 1/2

Organizácia

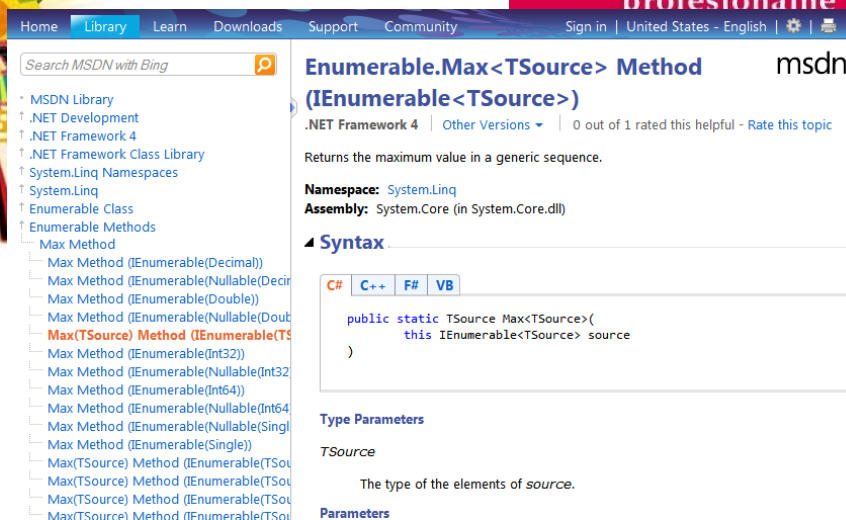
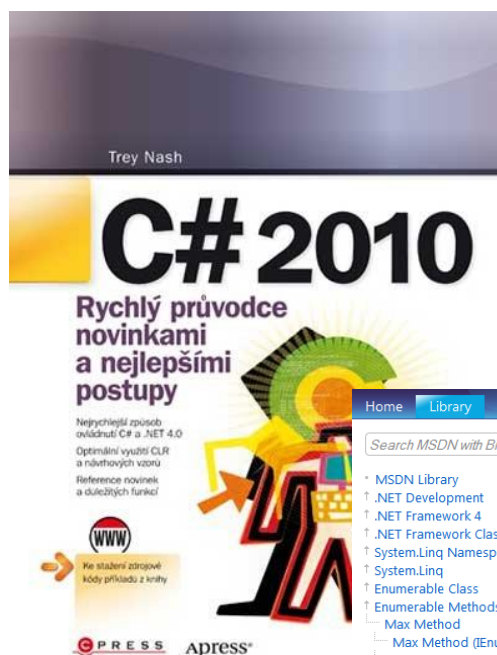
- Vyučujúci:
 - doc. Ing. Emil Kršák, PhD.
 - Ing. Marek Tavač, PhD.
 - Ing. Patrik Hrkút, PhD.
 - doc. Ing. Ján Janech, PhD.
 - Ing. Štefan Toth, PhD.
 - Ing. Michal Ďuračík
- eVzdelávanie
 - informácia o spôsobe hodnotenia (moodle)
 - zápis na cvičenia (prihlasovanie na skúšky)

Predpoklady

- Jazyky C++, Java
- Znalosť objektového programovania
- Základy systému Windows

Literatúra

- C# Programujeme profesionálne
 - Simon Robinson, K.Scott Allen, ...
- C# 2010
 - Trey Nash
- MSDN
- Internet



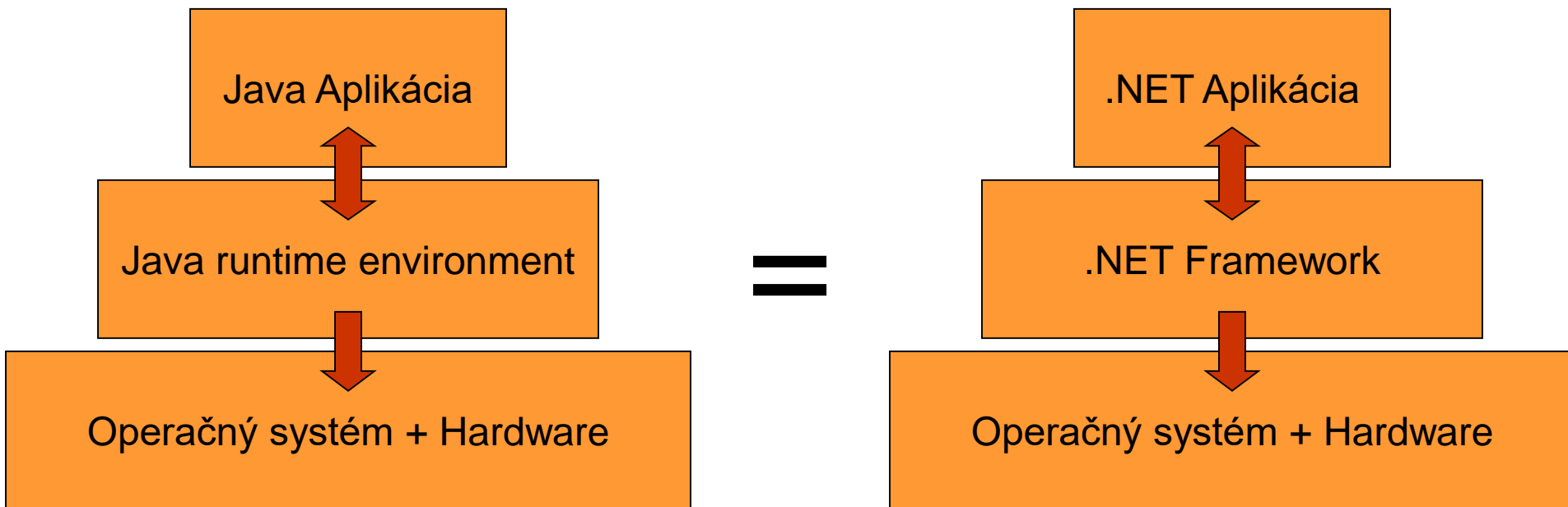
Základné témy

- Základy jazyka C#
- Objektové vlastnosti
- GUI
- Práca s dátami
- WCF a Sockety
- ASP .NET
- Reflexia

Výhody .NET Frameworku

- Podpora pre objektovo orientované programovanie
- Nezávislosť na jazyku a programovacej paradigme (F#)
- Dobrá podpora pre tvorbu dynamických webových stránok
- Jednoduchý prístup k dátam
- Podpora webových služieb
- Visual Studio .NET
- C#

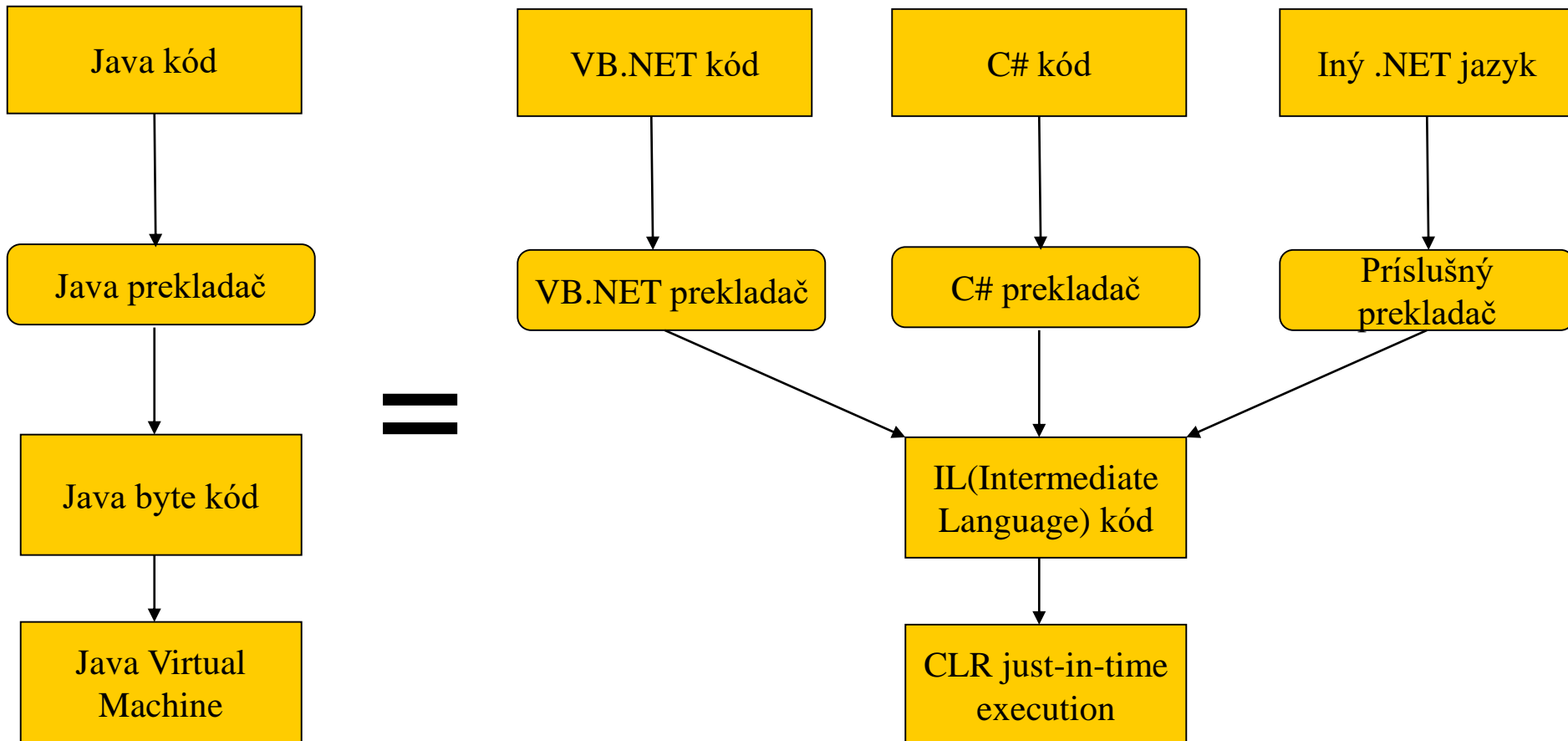
.NET Framework



Základné vlastnosti C#

- Dobrá podpora OOP
- Základné typy a typová bezpečnosť
- Dokumentácia v XML
- Garbage collector
- Atribúty (! nie atribúty z objektového programovania)
- Úplný prístup k BCL aj Windows API
- Podpora vlastností a udalostí
- ASP.NET

Preklad



Prvý program

```
using System;

namespace fri.pot.p1
{
    public class Hello
    {
        static void Main()
        {
            Console.WriteLine("Hello universe");
        }
    }
}
```

- using
- namespace
- Console
- metódy veľkým písmenom

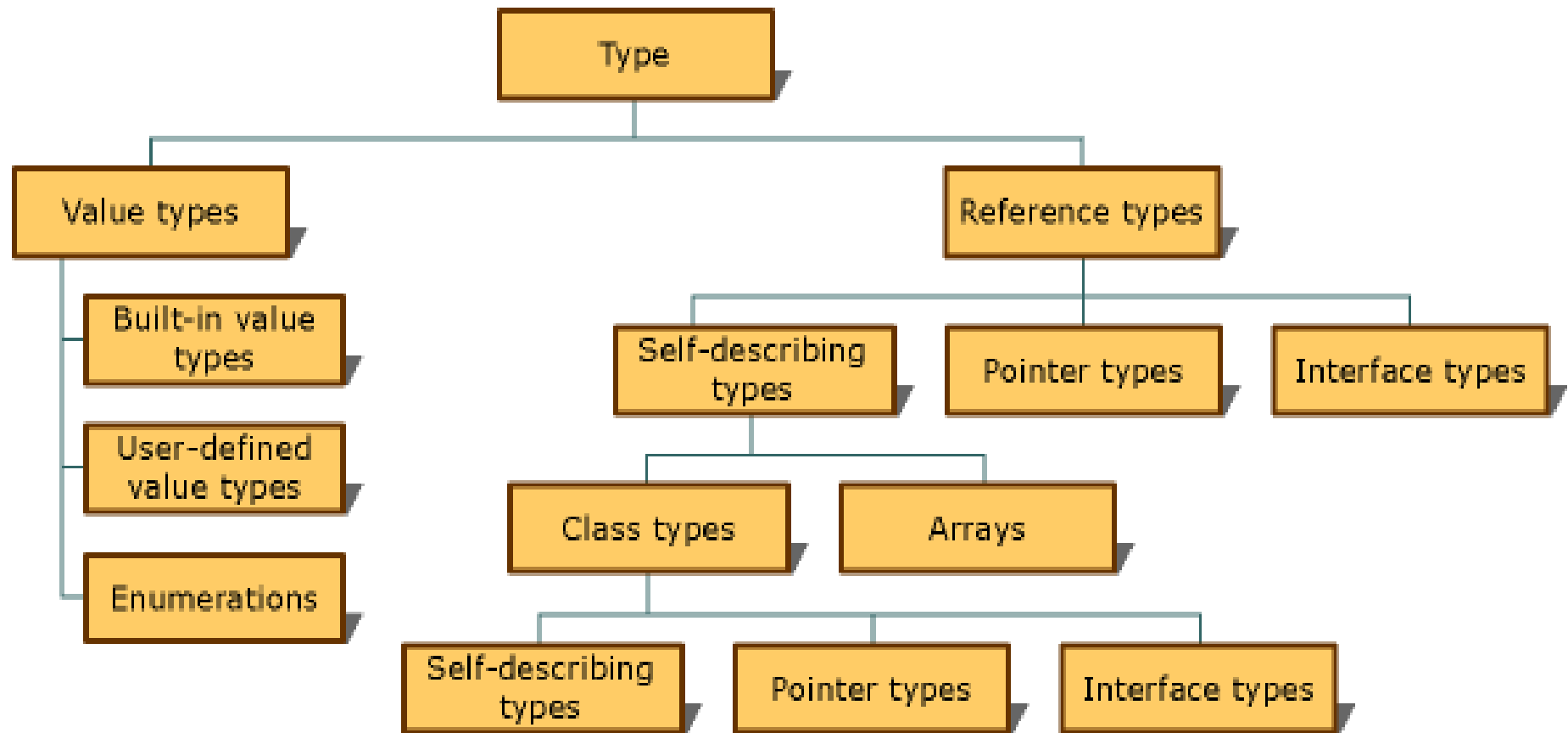
Dátové typy

- Vstavané
- Definované užívateľom
- Anonymné
- Nullable typy

Dátové typy

- Java
 - primitívne vs. objektové typy
- .NET:
 - len objektové
 - **Hodnotové** (value) vs. **odkazové** (reference)
- C#:
 - int vs. System.Int32

Hierarchia typov



Hodnotové typy

- Štruktúrové (struct)
 - Numerické
 - Celočíselné, desatinné, decimálne
 - Bool
 - Užívateľom definované

```
public struct MyStructure
{
    private int count;

    public void Increase()
    {
        count++;
    }

    // ...
}
```

- Vymenované (enums)

```
public enum Color
{
    Unknown,
    Black,
    White,
}
```

Vstavané hodnotové typy

Kategória	BCL Typ	C# typ	Popis
	System.SByte	sbyte	8-bit znamienkový integer
	System.Byte	byte	8-bit neznamienkový integer
	System.Int16	short	16-bit znamienkový integer
	System.UInt16	ushort	16-bit neznamienkový integer
	System.Int32	int	32-bit znamienkový integer
U	System.UInt32	uint	32-bit neznamienkový integer
L	System.Int64	long	64-bit znamienkový integer
UL	System.UInt64	ulong	64-bit neznamienkový integer

Vstavané hodnotové typy

Kategória	BCL Typ	C# typ	Popis
F	System.Single	float	32-bit desatinné číslo
D	System.Double	double	64-bit desatinné číslo
	System.Boolean	bool	true alebo false
	System.Char	char	UNICODE 16- bit znak
M	System.Decimal	decimal	96-bit decimálna hodnota

Odkazové typy

- Triedy
- Rozhrania
- Delegáti

Vstavané odkazové typy

C# Typ	BCL Typ	Popis
object	System.Object	Koreňový typ
string	System.String	Reťazec znakov Unicode

```
string str1 = "Ahoj " ;  
string str2 = "Svet";  
string str3 = str1 + str2;
```

```
string str1 = "Ahoj " ;  
string str2 = str1;    // str1 == str2 == "Ahoj"  
str1 = "Svet";         // str1 == "Svet", str2 == "Ahoj"
```

Operátory

- Zhodné ako v Java, C++

```
byte b = 255;  
b++;
```

Operátory

- checked, unchecked

```
byte b = 255;  
checked { b++;}  
unchecked {b++;}
```

- is, as
- sizeof
- typeof

Prevody typov

- Vysoká spoľahlivosť typov
- Implicitná konverzia
- Explicitná konverzia
 - () operátor konverzie
 - Na reťazce – ToString()
 - Z reťazca – Parse() – int.Parse(s)

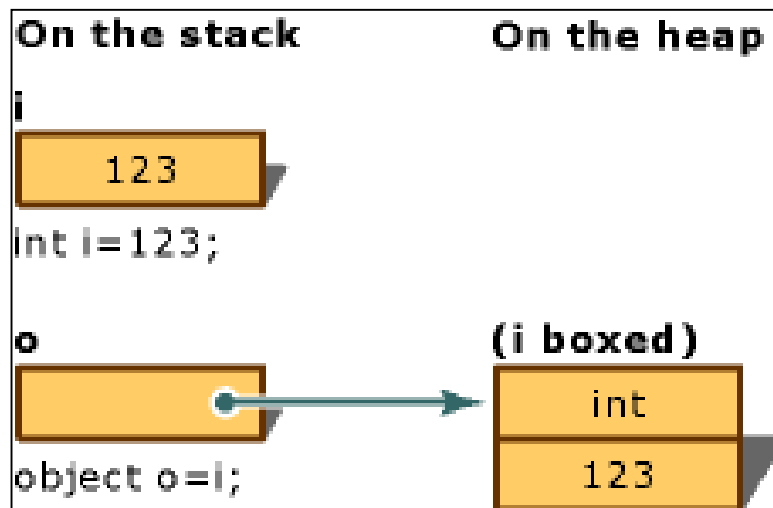
Implicitné konverzie

Z typu	Na typ
sbyte	short, int, long, float, double, decimal
byte	short, ushort, int, uint, long, ulong, float, double, decimal
short	int, long, float, double, decimal
ushort	int, uint, long, ulong, float, double, decimal
int	long, float, double, decimal
uint	long, ulong, float, double, decimal
(u)long	float, double, decimal
float	Double
char	ushort, int, uint, long, ulong, float, double, decimal

Balenie, rozbalenie

- Boxing, unboxing – rovnako ako Java
- zmena hodnotového na referenčný

```
int i = 123;  
object o = i;
```



Balenie, rozbalenie

- rozdiel od Java – boxing nie je podmienkou na použitie hodnotového typu ako objektu

```
int i = 123;  
string s = i.ToString();
```

```
string s = 10.ToString();
```


- Jednoznačný samostatný typ
- Potomok triedy System.Array
- Zhodný typ

```
int[] integers;  
int[] integers = new int[100];  
  
integers[0] = 10;  
  
int[] integers = new int[2] {1,2};  
int[] integers = new int[] {1,2};  
int[] integers = {1,2};
```

- Vlastné metódy
 - Length
 - Array.Sort()
 - Array.Reverse()

```
string[] krajiny = {"sk", "cz", "gb"};  
int pocet = krajiny.Length;  
Array.Sort(krajiny);  
foreach (string polozka in krajiny) {  
    Console.WriteLine(polozka);  
}
```

Viacrozmerné polia

- Pravouhlé – matica

```
int[,] integers = { {1,2}, {1,3}, {1,4} };

int[,] integers = new int[10, 20];
for (int i=0; i < 10; i++) {
    for (int j=0; j < 20; j++) {
        integers[i,j] = i*j;
    }
}

int rozmer = integers.GetLength(0); // == 10
```

Viacrozmerne polia

- Ortogonálne - rozoklané

```
int[][] integers = new int[2][];  
integers[0] = new int[10];  
integers[1] = new int[20];
```

Konštanty

```
const int i = 100;
```

- Iba lokálne premenné a dátové členy
- Vlastnosti
 - Inicializované v mieste deklarácie
 - Hodnota známa v dobe prekladu
 - Vždy sú statické
- Iba na čítanie

Riadenie toku

- Podmienené príkazy
- Cykly
- Príkazy skoku
- Výnimky

Príkaz if

```
if (podmienka)
    príkaz (alebo príkazy)
else
    príkaz (alebo príkazy)
```

- Aj bez else
- Podmienka vždy bool

```
string input;
input = Console.ReadLine();
if (input.Length < 5) {
    Console.WriteLine("Retazec ma menej ako 5 znakov");
} else {
    Console.WriteLine("Retazec ma aspon 5 znakov");
}
```

Príkaz switch

```
switch (výraz) {  
    case konštantný_výraz:  
        príkaz (alebo príkazy)  
        príkaz skoku  
    [default:  
        príkaz (alebo príkazy)  
        príkaz skoku]  
}
```

```
string jazyk;  
switch (krajina) {  
    case "Amerika":  
        goto case "Britan";  
    case "Britan":  
        jazyk = "English";  
        break;  
}
```


Príkaz switch

```
switch (výraz) {  
    case konštantný_výraz:  
        príkaz (alebo príkazy)  
        príkaz skoku  
    [default:  
        príkaz (alebo príkazy)  
        príkaz skoku]  
}
```

```
const string england = "uk";  
const string britain = "uk";  
switch (krajina) {  
    case england:                                // !!! chyba  
    case britain:  
        jazyk = "English";  
        break;  
}
```

Cyklus for

```
for (inicializácia; podmienka; krok)
    príkaz (alebo príkazy)
```

- Vyhodnotenie výrazu inicializácie
- Vyhodnotenie podmienky
- Vyhodnotenie výrazu kroku

```
for (int i = 0; i < 100; i+=10) {
    for (int j = i; j < i + 10; j++) {
        Console.Write(" " + j);
    }
    Console.WriteLine();
}
```

Cykly while, do ... while

```
while (podmienka)  
    príkaz (alebo príkazy)
```

```
do príkaz (alebo príkazy)  
while (podmienka)
```

Cyklus foreach

```
foreach (typ identifikátor in výraz)
    príkaz (alebo príkazy)
```

- Typ identifikátora
- Identifikátor
- Výraz
 - Kolekcia
 - Pole

```
foreach (int temp in poleintov) {
    Console.WriteLine(temp);
}
```

Príkazy skoku

- break
- continue
- return

Triedy

- Metódy
 - Inštančné, Statické
- Vlastnosti
 - Inštančné, Statické
- Konštruktory
 - Inštančné, Statický
- Deštruktory
- Operátory
- Indexery

Modifikátory prístupu

- private
- protected
- internal
- internal protected
- public

Metóda

```
[modifikátory] návratový_typ NázovMetody([argumenty])  
{  
    // telo metódy  
}
```

```
public bool IsSquare(Rectangle rect) {  
    return (rect.Height == rect.Width);  
}
```


Vstupné parametre

```
public static void Calc(int i) {  
    Console.WriteLine(„Calc - “ + i)  
    i = 10;  
}  
  
int x = 5;  
Calc(x);  
Console.WriteLine(„Vonku - “ + x);  
// vystup:  
// Calc - 5  
// Vonku - 5
```

Vstupno-výstupné parametre

- ref
- Predanie odkazom

```
public static void Calc(ref int i) {  
    Console.WriteLine(„Calc - “ + i)  
    i = 10;  
}  
  
int x = 5;  
Calc(ref x);  
Console.WriteLine(„Vonku - “ + x);  
// vystup:  
// Calc - 5  
// Vonku - 10
```

Výstupné parametre

- out
- Nepovinná inicializácia
- Predanie odkazom

```
public static void Calc(out int i) {  
    // i nie je inicializovane, nemozeme vypisat  
    i = 10;  
}  
....  
int x;  
Calc(out x);  
Console.WriteLine („Vonku - " + x);  
// vystup:  
// Vonku - 10
```

Vlastnosti

- Nový prvok – property
- Použitie ako atribút
- Funguje ako metóda
- Na čítanie alebo zápis

Pôvodne v jazyku Java

```
public class DopravnyProstriedok {  
    private int maxRychlost;  
  
    public void setMaxRychlost(int maxRychlost) {  
        this.maxRychlost = maxRychlost;  
    }  
  
    public int getMaxRychlost() {  
        return maxRychlost;  
    }  
}
```

použitie:

```
DopravnyProstriedok auto = new DopravnyProstriedok();  
auto.setMaxRychlost(160);  
System.out.println(auto.getMaxRychlost());
```

Vlastnosti v C#

```
public class DopravnyProstriedok {  
    private int maxRychlost;  
  
    public int MaxRychlost {  
        get {  
            return maxRychlost;  
        }  
        set {  
            maxRychlost = value;  
        }  
    }  
}
```

použitie:

```
DopravnyProstriedok auto = new DopravnyProstriedok();  
auto.MaxRychlost = 160;  
Console.WriteLine(auto.MaxRychlost);
```

- Nie je to premenná

```
public void Test(ref int speed)

Test(ref auto.MaxRychlost);      // !!! Chyba
```

```
public int MaxRychlost {
    get {
        return maxRychlost;
    }
    private set {
        maxRychlost = value;
    }
}
```

Automaticky implement. vlastnosti

- Netreba deklarovať dátový člen
- Musí byť get aj set časť

```
public class DopravnyProstriedok {  
    public int AktualnaRychlost { get; set; }  
  
    public int MaxRychlost { get; private set; }  
}
```


Vlastnosti vs. metódy

- Property – keď je to logický člen
- Metódy
 - Je to konverzia
 - Náročná operácia
 - Getter nečakaný vedľajší efekt
 - Volanie 2x – rôzne výsledky
 - Poradie vykonania je dôležité
 - Člen je pole
 - Nepoužívať write-only properties