

# C# Nyelvi Elemei

Tóth Zsolt

Miskolci Egyetem

2013

# Tartalomjegyzék

- 1 Object
- 2 Típusok
- 3 String
- 4 RegEx

- System névtér
- Osztály hierarchia gyökere
- Alacsony szint metódusok
- Öröklődés
- Metódus felüldefiniálás

## Metódusok

- Equals
- Finalize
- GetHashCode
- GetType
- MemberwiseClone
- ReferenceEquals
- ToString

# Objektumok összehasonlítása

- Equals(Object obj)
- bool
- Példány szintű
- statikus Equals(Object, Object)
- Saját egyezőség vizsgálat
- ReferenceEquals alapértelmezetten
- is operátor

```
override
bool Equals(Object obj) {
    if ((obj is MyClass) ==
        false)
    {
        return false;
    }
    ...
}
```

- Ellenőrzött megszüntetés
- Garbage collector
- Mikor?
- Alkalmazás
  - ▶ Nyitott adatfolyamok
  - ▶ Adatbázis kapcsolatok
  - ▶ Log
- Nem lehet közvetlenül implementálni
- `~MyClass`

- GetType()
- System.Type
- Típus deklarációt reprezentál
  - ▶ Osztály
  - ▶ Interface
  - ▶ Tömb
  - ▶ Érték
  - ▶ Enumeráció
  - ▶ ...

## Reflection

- Adatok lekérdezése
- Típus
- Interface-ek
- Ősosztály típusa
- Adattagok
- Konstruktorok
- Metódusok

## MemberwiseClone

- Shallow copy

## ToString

- String
- Objektumot reprezentálja

## GetHashCode

- Hash függvény
- Numerikus érték
- Objektum alapján
- Számos felhasználás
- Egyezőség tesztelése

# Tartalomjegyzék

1 Object

**2 Típusok**

3 String

4 RegEx



# Boxing, Unboxing

- Minden objektum
- Érték típus → objektum típus konverzió
- Cast-olás
- Implicit boxing  
object o = 5;
- Explicit boxing  
int i = 5;  
object o = (object)i;
- Objektum típus → érték típus konverzió
- Típus ellenőrzés
- Érték másolás
- InvalidCastException

# Érték és Objektum Típusok

## 1. táblázat. Numerikus Típusok

Érték	Objektum
byte	Byte
double	Double
float	Single
int	Int32
long	Int64

## 2. táblázat. Logikai és Szöveges Típusok

Érték	Objektum
bool	Boolean
char	Char
string	String

# Tartalomjegyzék

- 1 Object
- 2 Típusok
- 3 String**
- 4 RegEx

- Referencia típus
- Érték alapú összehasonlítás
- @ operátor
- kvótázás
- Escape szekvenciák

```
string path1 =  
    @"C:\myFile.txt";  
string path2 =  
    "C:\\myFile.txt";
```

- Copy
- Remove
- Replace
- Split
- Trim
- StartsWith
- EndsWith
- Substring
- ToUpper
- ToLower

# Tartalomjegyzék

- 1 Object
- 2 Típusok
- 3 String
- 4 RegEx**

# Regular Expression

- Szövegfeldolgozás
  - Minta illesztés
  - Egyszerű struktúra
  - Szöveg = karakterlánc
  - Szövegrészek
  - Számosság
- `[A-Z]{3}-[0-9]{3}`
  - `.*\.txt$`

# Wildcards

- Előre nem ismert karakterek
- Speciális karakterek
  - . tetszőleges karakter
  - \n újsor
  - \d számjegy
  - \w szó
  - \s fehér karakter
  - ^ string eleje
  - \$ string vége



## Blokk

- Karakter halmaz []
- Egységként kezelt
- Tetszőleges választás
- Intervallum is megadható –

## Számosság

### Egyszer

? 0 vagy 1

+ 1 vagy több

\* 0 vagy több

{n} pontosan n

{n,m} legalább n, maximum m

# Regex Osztály

- Text.RegularExpressions névtér
- Regex osztály
- Fix reguláris kifejezést reprezentál
- IsMatch(String)
- Replace(String,String)

```
string input = "a
    string";
string pattern = "\\s+";
string replacement = "
    ";
Regex rgx = new
    Regex(pattern);
string result =
    rgx.Replace(input,
        replacement);
```