PV178 – Lab07

Karel Jiránek


Kahoot!

Agenda

- Proudy (streamy)
- Soubory, Adresáře, Cesty
- Atributy
- Tuple
- Record

Proudy (streamy)

- Proud – prostředník mezi programem a zdrojem dat (např.: souborem)
- Je třeba používat **Dispose**
- Čtení – StreamReader, zápis – StreamWriter
- Příklady
  - FileStream
  - NetworkStream
  - MemoryStream

Proudy

Samostatná práce

- Streams

Soubory, Adresáře, Cesty

- Používáme statické třídy `File`, `Directory`, `Path`
- Často používané metody:
  - `File.Create`, `File.Delete`, `File.ReadAllText`
  - `Directory.GetFiles`, `Directory.GetDirectories`
  - **`Path.Combine`**
- Nikdy nepoužívejte cestu, kde jsou „na trvdo“ specifikovaná lomítka a celá cesta
  - `"C:\user\thmoas\myFile.txt" //cesta kterou umí číst jen Windows!`
  - `"../application/myFile.txt" //cesta kterou umí číst jen Linux!`

Samostatná práce

- FilesAndFolders

Atributy

- Značka
- Metadata
- Zapisuje se v hranatých závorkách nad metodami nebo třídami
- Můžeme mít vlastní atributy

```
[Serializable]
[DebuggerDisplay("Customer name: {FullName}, age: {Age}")]
[Obsolete("Will be removed in next version.")]
public class Customer
{
        ...
}
```

Atributy

Samostatná práce

- Atributy (na doma)

Tuple

- Datová struktura spojuje různá data
- Lze nahradit třídou (v 95% případech)
- Jednotlivé položky nemají své jméno
- Doporučuji nepoužívat – nepřehledné!
- Použití často svědčí o špatném návrhu

  - jak vrátit 2+ hodnoty z metody? **Tuple**

  - jak spojit 2+ hodnot? **Tuple**

  -> řešením je třída

Tuple

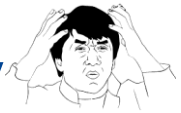
Init only setters (od C# 9)

- Možnost nastavit vlastnost mimo konstruktor, a následně již používat jako readonly

```
public class Person
{
    public string Name { get; init; }

    public int Salary { get; }

    public Person()
    {
        Name = "N/A";
        Salary = 0;
    }
}


// Použití
var person = new Person { Name = "John" };
```

Records (od C# 9)

- Podobné třídě
- Referenční typ, ale value-based equality
- Používají init-only settery
- **Nedestruktivní mutace**
- Využití na místech, kdy by třída neměla metody
- Mohou
  - navzájem od sebe dědit
  - být abstraktní
- Paradoxně mohou obsahovat vlastnosti, metody



Records