

# Cviko08 recap

## IDisposable

Interface, ktorým označujeme triedy, ktoré pracujú so zdrojmi (garbage collector nevie kedy a ako ich môže uvoľniť)  
napr. súbory, rôzne sieťové srandy, databázové kontexty...

Potrebujeme zaručiť, že sa metóda Dispose zavolá vždy aby sme uvoľnili zdroje

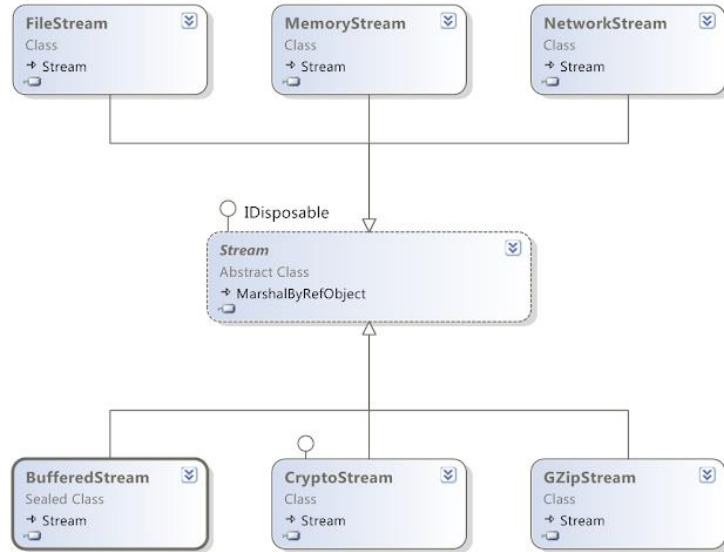
```
var metla = new MetlaInternetova();
try
{
    ...// hranie sa s metlou
}
finally
{
    ...metla.Dispose();
}
```

Korektná práca s triedami implementujúcimi tento interface

```
using (var metla = new MetlaInternetova())
{
    ...// hranie sa s metlou
}
```

Prosím používajte len toto

# Streamy



## Základné streamy

Decorator streamy (decorator pattern ukážka v solutione v triede CaesarStream)

```
using (var streamWriter = new StreamWriter("filePath"))
{
    ...
}
```

Pre prácu so súbormi stačí používať triedy StreamReader, StreamWriter (v prípade binárnych súborov BinaryReader a BinaryWriter)

## Regexy

Cheatsheet nájdete v študijných materiáloch

Regulárne výrazy na hľadanie patternov v texte  
(nesnažte sa nimi parsovať bezkontextové jazyky ako napr. html)

Vďaka tomuto nemusíme escapovať \ (vhodné v paths a regexoch)  
Vieme to kombinovať aj s \$ pre interpoláciu

```
var pattern = new Regex(@"\dx (?<oopTerm>metla)");  
var match = pattern.Match(oop);  
var oopTerm = match.Groups["oopTerm"].Value;
```

Pomenovaná groupa

K pomenovaným groupám prístupuje ako k slovníku, inak pomocou indexu

[Regex tester do Visual Studio](#)

[Slušný webový regex tester](#)