Cviko09 recap

Thread

Trieda reprezentujúca OS vlákna
Ich vytvorenie je časovo náročné
Obmedzená funkcionalita

```
public Thread(ThreadStart start);
public Thread(ParameterizedThreadStart start);
```

Vieme na nich spustiť len metódy bez návratovej hodnoty
s max 1 argumentom typu *object*

BAD

```
try
{
    new Thread(() => throw new NullReferenceException()).Start();
}
catch (NullReferenceException)
{
    //handle exception
}
```

Zvonku nechytíme žiadnu výnimku vyvolanú vo vlákne

Vlákno si musí handlovať výnimky samé

```
lock (locker)
{
    if (!done)
    {
        done = true;
        Console.WriteLine("I am done");
    }
}
```

Vždy pri práci s viacerými vláknami dávať pozor na kritické sekcie!
Pri ich ošetrovaní dbať nato aby program neskončil v deadlocku

Threadpool

Kolekcia predalokovaných vláken
Rovnako obmedzené ako Thread, ale bez nutnosti vytvárať nové vlákna

```
ThreadPool.QueueUserWorkItem(Console.WriteLine, 123);
ThreadPool.QueueUserWorkItem(Console.WriteLine);
```

Na pozadí to využíva queue na joby

Task

Abstrakcia nad vláknami
Viacero možností ako s nimi pracovať
Preferovaný pred Thread/ThreadPool

```
var task = Task.Run(() => Console.WriteLine(message + Task.CurrentId));
```

```
const int factorial = 3;
var startedTask = Task.Run(() => Factorial.ComputeSmallFactorial(factorial));
var factorialOfThree = startedTask.Result;
```

Tasky majú už aj návratové hodnoty
Vytiahnuť sa dá z property Result
Nezabúdať čakať na Tasky pomocou Wait()

```
try
{
    Task.Run(() => throw new NotImplementedException()).Wait();
}
catch (AggregateException)
{
    //handle exception
}
```

Chytať výnimky vieme aj zvonku
Vyhodená výnimka sa obalí do AggregateException

```
var taskWithException = Task.Run(() => throw new NotSupportedException());
var handledTask = taskWithException.ContinueWith(task => Console.WriteLine($"Thrown exception: {task?.Exception?.GetType()}"),
    TaskContinuationOptions.OnlyOnFaulted);
```

Naväzovanie Taskov, na základe stavu v akom skončili

Pri vytvárani vieme Tasku posunúť taktiež CancellationToken, pomocou, ktorého vieme zabiť daný task zvonku