TPL

Vincent Uhlmann
IT-Akademie Dr. Heuer GmbH

PARALLEL

- Die Klasse Parallel bietet eine Reihe von Methoden, um parallele Ausführung zu ermöglichen
- Die zu verarbeitenden Daten werden aufgeteilt
- Anzahl der Teilmengen wird automatisch festgelegt, kann über die ParallelOptions jedoch überschrieben werden
- Sorgt für eine optimale Ausnutzung der Hardware (z.B. Anzahl der Prozessoren)
- Die Threads bezieht die Klasse aus dem Threadpool

PARALLEL FOR

- Eine for-Schleife bei der die Iterationen parallel ausgeführt werden können
- Gibt ein ParallelLoopResult zurück, das Informationen über die abgeschlossenen Iterationen enthält

```
Parallel.For(0, 100, i => {
    Console.WriteLine(i);
});
```

PARALLEL FOREACH

- Eine foreach-Schleife bei der die Iterationen parallel ausgeführt werden können
- Gibt ein ParallelLoopResult zurück, das Informationen über die abgeschlossenen Iterationen enthält

```
var directoryInfo = new
DirectoryInfo("C:\\Windows\\System32").GetFiles();

Parallel.ForEach(directoryInfo, i =>
{
    Console.WriteLine(i.Name);
});
```

PARALLEL INVOKE

- Parallel.Invoke wird verwendet, um mehrere Methoden gleichzeitig auszuführen
- Akzeptiert wird eine Liste von Delegaten die parallel ausgeführt werden sollen
- Erst wenn alle übergebenen Methoden abgearbeitet sind, wird die auf Invoke folgende Anweisung fortgesetzt

```
static void Main(string[] args)
{
    Parallel.Invoke(DoSomething, DoSomething2);
    Console.WriteLine("Fertig");
}

public static void DoSomething()
{
    Thread.Sleep(5000);
    Console.WriteLine("Thread1 fertig");
}

public static void DoSomething2()
{
    Thread.Sleep(7000);
    Console.WriteLine("Thread2 fertig");
}
```