

Aufgaben Multithreading

Aufgabe 1:

- Erstelle einen Thread, der eine Methode ausführt, die die Zahlen von 1 bis 10 ausgibt.
- Währenddessen soll die Hauptmethode eine andere Nachricht ausgeben.
- Überprüfe die parallele Ausgabe beider Threads.

Aufgabe 2:

- Erstelle eine Anwendung mit zwei Threads, die denselben Zähler 1000-mal inkrementieren
- Gib den Endwert des Zählers nach Abschluss beider Threads aus
- Starte die Anwendung mehrmals und beobachte die Ergebnisse

Aufgabe 3:

- Ändere Aufgabe 2, um die Race Condition mithilfe des lock-Schlüsselworts zu beheben.
- Teste die Anwendung erneut

Aufgabe 4:

- Simuliere einen Deadlock durch zwei Threads, die auf zwei Ressourcen in umgekehrter Reihenfolge zugreifen.
- Vermeide den Deadlock, indem du eine feste Sperrreihenfolge einführest

Aufgabe 5:

- Erstelle eine Anwendung, in der mehrere Threads gleichzeitig auf eine Ressource zugreifen
- Nutze einen Mutex, um den exklusiven Zugriff sicherzustellen
- Verwende anschließend ein Semaphore, um maximal zwei Threads gleichzeitig Zugriff zu gewähren

Funktion zum Zeitmessen:

```
var sw = System.Diagnostics.Stopwatch.StartNew();
```

```
sw.Stop();
```

```
Console.WriteLine(sw.ElapsedMilliseconds);
```