

1.

Unser Gehirn verarbeitet Informationen parallel. Die einzelnen Nervenzellen können unabhängig von anderen Nervenzellen arbeiten. Würde das Gehirn concurrent arbeiten, müsste jede Zellen pausieren, damit eine andere arbeiten kann.

2.

OO-Kapselung soll dabei helfen, den internen Status eines Objektes vor äußerer Manipulation zu schützen und eine Stimmigkeit der Daten garantieren. Dies kann in Konflikt mit Prozessen kommen, die gleichzeitig den Zustand eines Objektes verändern. Wenn Prozess A anfängt einen Variable zu verändern und während diesem Prozess unterbrochen wird, um Prozess B auszuführen, arbeitet Prozess noch mit dem alten Zustand von der Variable, da Prozess A noch nicht zu ende geführt wurde. Vor diesem Problem schützen auch keine Kapselung der Objekte, da private Fields durch public Methoden geteilt werden.

3.

static Variables:

Diese müssen nicht synchronisiert werden, da der Wert, der in ihnen gespeichert ist konstant bleibt und nicht verändert wird.

Fields/ Attribute in Objekten:

Fields und Attribute von Objekten sollten synchronisiert werden, um das Lost Update Problem zu vermeiden.

Stackvariablen:

Jeder Thread hat seinen eigenen Stack. Das heißt, dass jeder Thread seine eigene lokalen Variablen besitzt und diese nicht mit anderen Threads geteilt werden.

-> Threadsafe

Heapvariablen:

Mit new() allokierte Heapvariablen sind ebenfalls Thread Safe. Obwohl die memory auf dem Heap ist, ist der Pointer zur Position im Stack gespeichert. Das heißt, dass nur der Thread den Pointer zum Memory auf dem Heap hat.

5.

Thread Killing ist inhärent unsicher, da es alle Monitors öffnet, die der Thread verschlossen hat. Ein Monitor ist ein Mechanismus, der kontrolliert, wer Zugriff auf ein Objekt hat. Dadurch kann es passieren, dass andere Threads auf inkonsistente Objekte zugreifen.

6.

Wenn die Applikation parallel läuft ist sie auf Multi-Core Maschine schneller. Wenn sie Concurrent läuft ist sie auf der Maschine schneller, die eine höhere Single-Core Performance besitzt.

7.

Nicht deterministische Ergebnisse können durch falsches Überschreiben von Werten verursacht werden. Durch verschiedene Laufzeiten gleicher Methoden, kann es zu unterschiedlichen Ergebnissen kommen, obwohl der gleiche Code ausgeführt wurde.

8.