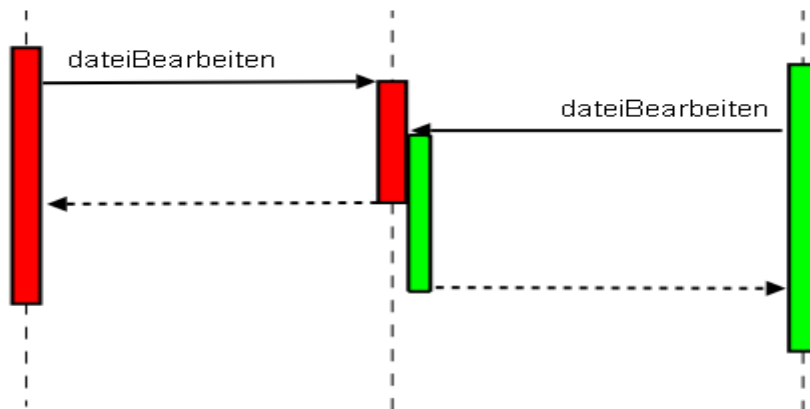


Kritischer Abschnitt

Monitorkonzept zur Lösung des Synchronisationsproblems

Synchronisationsproblem: Ein Synchronisationsproblem kann entstehen, wenn der Prozessor, während ein Prozess oder Thread läuft, die Zeit wegnimmt und einem neuen oder anderen Prozess/Thread startet, der auf den selben Hintergrund zugreift und dadurch kann es zu Problemen



kommen.

An dieser Grafik kann man sehr gut sehen, dass dem roten Prozess die Arbeitszeit vom Prozessor weggenommen wird und der grüne auf den selben Prozesshintergrund zugreift.

Bsp.: Ein Ehepaar greift auf das gleiche Konto zu, das nicht in den negativen Bereich fallen darf. Dabei wird ein Programm geschrieben das der Grafik oben entspricht. Im Programm wird zuerst geprüft, ob man noch das gewünschte Geld vom Konto abheben kann. Wenn das der Fall ist wird der Betrag vom Konto abgehoben. Es ist aber möglich, dass der Prozess nach der Prüfung angehalten wird und der andere Prozess gestartet wird. Dann hebt der 2. Prozess Geld ab und der 1. kann ebenfalls nochmal abheben, weil die Prüfung nicht mehr gemacht wird. Dann passiert es, dass das Konto in den negativen Bereich fallen kann, das ist das Synchronisationsproblem.

Lösung des Problems: Die Lösung des Problems ist, dass man die Prozesse synchronisiert, so dass der Prozess nicht abgebrochen wird, auch wenn die Arbeitszeit durch den Prozessor eigentlich schon beendet wurde. Dieses Art der Lösung nennt man Monitorkonzept.

Bsp.:

```
class Something {
    private SomeType sharedData;

    synchronized public void fct1 (...) {
        ...}
    synchronized public void fct2 (...) {
        ... }}
```