

Grundlagen der Programmierung 2 SS 2005 - Aufgabenblatt 6

Ausgabe: 20.05.05

Aufgabe 20 (Nebenläufigkeit)

Betrachten Sie das Programm `Increaser.java`. Dieses versucht auf parallele Weise eine Summe zu errechnen. Compilieren Sie das Programm und führen Sie es mehrfach aus. Welchen Effekt können Sie beobachten und welches Ergebnis hätten Sie erwartet? Wie könnte man das Programm verbessern?

Aufgabe 21 (Thread-Sichere Datenstrukturen)

a)

Die Datenstruktur `Stack` soll mit Hilfe von linearen Listen implementiert werden. `Stack`-Objekte dieser Implementierung sollen von mehreren Threads benutzt werden können. Geben Sie an, welche der Methoden `push`, `pop`, `top` und `empty` als `synchronized` charakterisiert werden müssen und begründen Sie Ihre Entscheidungen.

b)

Implementieren Sie die Klasse `Stack` aus (a). Benutzen Sie eine Hilfsklasse für die Verkettung. Der `Stack` soll Elemente beliebiger Klassen aufnehmen können.

c)

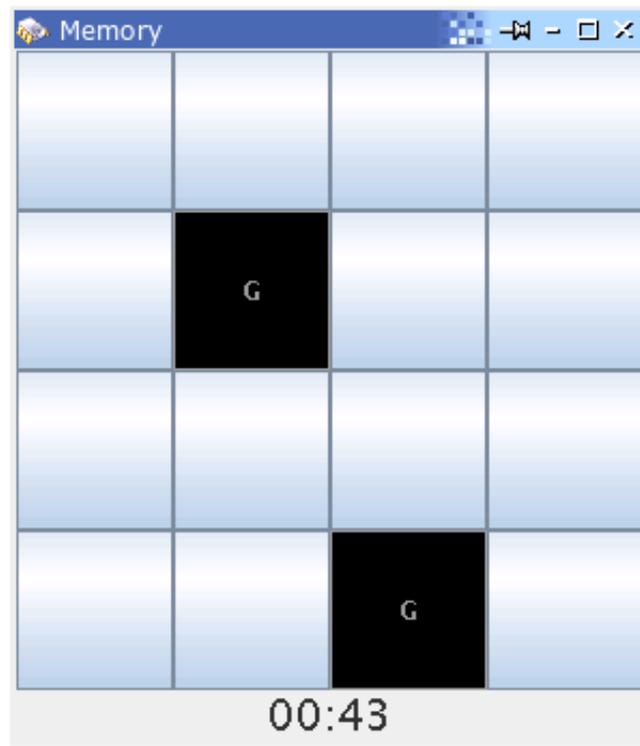
Implementieren Sie eine Prozessklasse `StackUser` (als Unterklasse von `Thread`), die einen `Stack` wie folgt verwendet: der Prozess legt in einer ersten Phase zunächst einige `String`-Objekte auf den `Stack` und in einer zweiten Phase werden die `String`-Objekte wieder entfernt. Fügen Sie kurze Pausen (mit der Methode `Thread.sleep`) und Bildschirmausgaben ein, damit man das Verhalten des Prozesses beobachten kann.

d)

Ergänzen Sie ein Hauptprogramm, das zwei `StackUser`-Prozesse erzeugt und diese auf einem gemeinsamen `Stack` parallel arbeiten lässt.

Aufgabe 22 (Thread zur Zeitmessung)

Erweitern Sie das Memory-Spiel `Memory.java` aus Aufgabenblatt 4 so, dass zusätzlich eine Uhr eingeblendet wird (siehe Screenshot), welche die bisherige Spieldauer in Minuten und Sekunden anzeigt. Hat der Spieler alle Karten aufgedeckt, so wird die Zeitmessung angehalten und eine entsprechende Meldung ausgegeben (z.B. über die Klasse `JOptionPane`).



Hinweis: Die Klassenmethode `System.currentTimeMillis()` liefert die Zeit (in Millisekunden), die seit dem Einschalten des Rechners vergangen ist. Damit kann eine präzise Zeitmessung realisiert werden. Das Ergebnis der Methode ist vom Typ `long`!

Aufgabe 23 (Threads zum Ausführen neben der Oberfläche)

In der Datei `Prime.java` finden Sie ein Programm, welches bestimmt, ob eine Zahl eine Primzahl ist. Jedoch wird während der Berechnung die grafische Oberfläche des Programmes blockiert. Wenn Sie sehr grosse Zahlen eingeben, z.B. eine Eins gefolgt von 15 oder 16 Nullen, ist der Effekt am blinkenden Textcursor deutlich zu sehen. Erweitern Sie das Programm derart, dass der aufwändige Teil der Berechnung in einem Thread parallel zur grafischen Oberfläche abläuft.