

Übungen zu Informatik B

Sommersemester 2004

Blatt 9

Dies ist das vorletzte Aufgabenblatt.

Aufgabe 1 (6 Punkte)

Wie werden *Threads* in Java realisiert, und was hat das mit dem Interface `Runnable` zu tun? Erläutern Sie die Thread-Operationen `interrupt()` und `join()`.

Was versteht man in Java unter den Begriffen *Synchronisation* und *Monitor*? Wann ist *Synchronisation* erforderlich, und wie wird ein *Monitor* dazu verwendet?

Erklären Sie anhand des Beispiels `code/20/Listing2213.java` aus der Vorlesung den Einsatz von `wait()`, `notify()`. Welches Objekt wird hier als *Monitor* verwendet und warum?

Aufgabe 2 (7 Punkte)

Implementieren Sie ein Java-Programm, das nach einer Datei mit einem vorgegebenen Namen in einer Liste von Verzeichnissen suchen kann. Die Aufruf-Syntax des Programms soll dabei folgendermaßen aussehen:

```
java FindFile datei verzeichnis ...
```

Ihr Programm bekommt auf der Kommandozeile einen Dateinamen sowie eine Liste von Verzeichnisnamen übergeben und sucht *rekursiv* in den angegebenen Verzeichnissen (einschließlich Unterverzeichnissen) nach einer Datei mit dem angegebenen Namen.

Unterverzeichnisse sollen dabei *parallel* durchsucht werden, indem für jedes Verzeichnis ein eigener *Thread* gestartet wird. Die Liste der im Laufe der Suche gefundenen Pfade soll während des Programmablaufs in einer Collection vom Typ *ArrayList* gesammelt und am Programmende ausgegeben werden.

Aufgabe 3 (7 Punkte)

Erweitern Sie Ihr „Vier Gewinnt“ Java Applet aus Aufgabe 2 von Blatt 8 (oder die entsprechende Musterlösung) um eine Begrenzung für die Zeit, die ein Spieler für seinen Zug hat (z.B. 30 Sekunden). Verwenden Sie dazu einen eigenen (nebenläufigen) *Thread*, in dem ein Zähler sekundenweise heruntergezählt wird.

Dazu sollte es in der Oberfläche ein Anzeigeelement (`JLabel`) geben, in dem fortlaufend die dem aktuellen Spieler noch verbleibende Zeit in Sekunden angezeigt (d.h. ebenfalls heruntergezählt) wird. Ist die Zeit ohne eine Aktion des Benutzers verstrichen, wählt das Programm automatisch eine zufällige Spalte für den Spielstein, und wirft ihn dort hinein. Anschließend ist der andere Spieler an der Reihe.

— bitte wenden —

Wenn der Spieler innerhalb der 30 Sekunden eine Spalte ausgewählt hat, wird der Zähler über einen Aufruf von `interrupt()` angehalten, und dann für den nächsten Spieler neu gestartet (sofern das Spiel noch nicht beendet ist).

Hinweis: Da die Swing-Grafikkomponenten *nicht threadsafe* sind (d.h. sie dürfen nur aus einem Thread zur Zeit angesprochen werden), muß die Aktualisierung der Darstellung hier vom sogenannten *Event-Thread* ausgeführt werden. Dazu gibt es die vordefinierte Methode *invokeAndWait()* in der Klasse `javax.swing.SwingUtilities`, die einen Methodenaufruf an den Event-Thread zur Ausführung übergeben kann. Ein Beispiel dazu befindet sich in der Java-Dokumentation zu dieser Methode.

Gesamtpunkte: 20