

Übungen zu Informatik B

Sommersemester 2004

Blatt 8

Aufgabe 1 (5 Punkte)

Was versteht man in Java unter einem *Applet*?

Worin unterscheidet sich ein *Applet* von einer normalen Java Anwendung? Wie wird ein Applet erzeugt, und welche Methoden besitzen im Lebenszyklus eines *Applets* eine besondere Bedeutung?

Welche Rolle spielt in diesem Zusammenhang die Klasse *AppletContext*?

Aufgabe 2 (6 Punkte)

Erstellen Sie eine Version Ihres Spiels „Vier Gewinnt“ aus Aufgabe 2 von Blatt 6, die als *Applet* in einem Browser ausgeführt werden kann.

Die Größe des Spielfelds soll dabei über *Parameter-Elemente* auf der HTML-Seite gesetzt werden können (ist also nicht mehr fest). Wenn keine *Parameter-Elemente* hierfür angegeben sind, wird eine voreingestellte Größe verwendet.

Ersetzen Sie die Statusanzeige in Ihrem Programm durch eine Anzeige in der Statuszeile des Browser-Fensters, in dem das Applet angezeigt wird. Erweitern Sie Ihr Programm auch um eine Schaltfläche, mit der das Spiel neu gestartet werden kann.

Aufgabe 3 (9 Punkte)

Lesen Sie das Manual zu dem Unix-Kommando *du*, und implementieren Sie dann in Java eine vereinfachte Version dieses Kommandos. Die Aufruf-Syntax des Programms soll dabei folgendermaßen aussehen:

```
java Du [-a|-s] [-b|-h|-k|-m] [-c] [-X datei] [datei ...]
```

Ihr Programm bekommt auf der Kommandozeile eine Liste von Datei- oder Katalognamen übergeben und gibt die Größe dieser Dateien bzw. Kataloge in Kilobyte (1024 Byte) aus. Wenn kein Dateiarargument angegeben ist, wird der aktuelle Katalog verwendet („.“).

Bei einem Katalog bedeutet dabei die „Größe“ die Summe der Größen aller darin enthaltenen Dateien und Kataloge, es muß also dazu *rekursiv* die Größe aller Unterkataloge ermittelt werden. In diesem Fall wird auch für jeden Unterkatalog die hierfür jeweils ermittelte Größe mit ausgegeben.

Hinweis: In Java liefert die Klasse `java.io.File` Informationen über Dateien und Kataloge.

— bitte wenden —

Um die Verarbeitung der Schalter auf der Kommandozeile zu erleichtern, enthält das Paket `cmdline` aus dem Katalog `uebung/Blatt8` eine kleine Hilfsklasse `Getopt`, die das Erkennen der Schalter (und evtl. dazugehöriger Werte) übernimmt.

Benutzen Sie diese Klasse, um folgenden Schalter zu implementieren:

- `-a` Größenangaben einzeln für alle Dateien
- `-b` Größenangaben in Byte ausgeben
- `-c` Gesamtsumme über alle Argumente am Ende anzeigen
- `-h` Größenangaben in „lesbarer“ Form (z.B. 3K oder 2M)
- `-k` Größenangaben in Kilobyte (1024 Byte) ausgeben (Voreinstellung)
- `-m` Größenangaben in Megabyte (1048576 Byte) ausgeben
- `-s` Größenangaben nur für die Argumente anzeigen
- `-X datei`
Dateien bzw. Kataloge überspringen, deren Name in der angegebenen Datei aufgelistet ist (ein Datei- oder Katalogname pro Zeile, siehe unten)

Die Schalter `-a` und `-s` schließen sich dabei gegenseitig aus. Außerdem darf nur maximal einer der Schalter `-b`, `-h`, `-k`, und `-m` zur Festlegung des Ausgabeformats angegeben werden. Bruchteile werden dabei nicht mit ausgegeben, sondern immer aufgerundet.

Gelegentlich ist es sinnvoll, bestimmte Dateien oder Kataloge von der Größenberechnung auszunehmen. Das Kommando bietet daher die Möglichkeit, beim Ermitteln der Einträge in einem Katalog (während einer rekursiven Größenberechnung) alle Dateien und Kataloge auszulassen, deren Namen auf einer vorgegebenen Liste auftauchen. Diese Namensliste wird dem Kommando in Form einer Textdatei übergeben, die diese Namen enthält (ein Name pro Zeile), und deren Name als Wert für den Schalter `-X` angegeben wird. Explizit auf der Kommandozeile angegebene Dateinamen werden aber in jedem Fall verarbeitet, auch wenn sie auf dieser Liste auftauchen.

Verwenden Sie zur Realisierung dieser Funktionalität eine eigene Klasse, die das Interface `java.io.FileFilter` implementiert und die dann beim rekursiven Durchlaufen der Kataloge verwendet wird.

Gesamtpunkte: 20