Übungsblatt #08

Abgabetermin: Donnerstag, 02.07.2009

Zur Motivation...

Dieses Übungsblatt stellen wir schwerpunktmäßig mal unter das Thema „Logik", d.h. hier konkret: unter den Umgang mit „Bool'sche Variablen".

Bislang führten sie ein eher nebensächliches Schatten-Dasein in unseren Programmen, aber diese Woche lernen wir kennen, wie man sie auch zum Finden von möglichen Lösungen bei Berücksichtigung mehrerer Bedingungen einsetzen kann.

Für die Bearbeitung dieser Aufgabe solltet Ihr ggf. nochmals das Skript zur Aussagenlogik des vergangenen Semesters zu Rate ziehen (Teil 6), da einiges, was damals formal dort beschrieben wurde, nun konkret zur Anwendung gebracht werden soll.

Aufgabe 1 soll nicht am Rechner sondern „mit Papier und Bleistift" gelöst werden. Und auch bei Aufgabe 2 solltet Ihr erst ohne Rechner die nötigen Vorbereitungen (Umformulierung der Aussagen) treffen.

Aufgabe 1: Bedingungen und Schleifen... (10 Punkte)

klausurziemlichverdächtig

Bekanntlich führen ja viele Wege nach Rom, und auch in der Programmierung (ob in Java oder in anderen Sprachen) stehen mitunter verschiedene Möglichkeiten zur Verfügung, das Gewünschte zu erreichen. Teilweise ist es eine Frage des persönlichen Geschmacks, für was man sich entscheidet, teilweise ist es eine Frage des guten Programmierstils. Zwei Beispiele dafür sehen wir uns in dieser Aufgabe einmal an.

```
// a bekommt eine Zahl zugewiesen
int a = holeZahlIrgendwoHer();
if (a > 5) {
   return true;
} else {
   return false;
}
```

a) Bool'sche Ausdrücke (2 Punkte)

Gegegeben sei der links unten stehende Ausschnitt aus einem Java-Programm. Schaut Euch die IF-Bedingung an und prüft, ob sie sich auch wesentlich kürzer ausdrücken lässt (wenn ja: wie?)!

b) WHILE- vs. FOR-Schleife, die erste (3 Punkte)

In unseren Beispielsprogrammen verwenden wir mal die FOR- und mal die WHILE-Schleife. Anstelle einer WHILE- kann ebenso gut auch die FOR-Schleife verwendet werden (oder umgekehrt). Diese Teilaufgabe zeigt einen solchen Fall. Nehmt daher die nachfolgende WHILE-Schleife und schreibt sie in eine FOR-Schleife um.

```
int i=0;
while (i < 100) {
   doSomething();
   i++;
}
```


Ein von der Informatik gebeutelter WiWi-Student im 2. Semester geht zum Psychiater, weil er sich vom vielen abstrakten, logischen Denken beim Programmieren irgendwie krank fühlt. Die Antwort des Psychiaters fällt aber leider nicht ganz so aus, wie sich unser Student das wünscht, denn die Diagnose des Psychiaters lautet wie folgt:

- „Lieber Mann, Sie sind hysterisch oder größenwahnsinnig oder beides!
- Wenn Sie größenwahnsinnig sind, jedoch keinen Minderwertigkeitskomplex haben, dann haben Sie Zwangsvorstellungen.
- Mit Sicherheit haben Sie nicht sowohl Zwangsvorstellungen als auch Depressionen!
- Sollten Sie einen Minderwertigkeitskomplex haben, dann haben Sie keine Depressionen!
- Niemand kann zugleich Depressionen haben und hysterisch sein, das sollten Sie wissen!
- Falls Sie Zwangsvorstellungen haben, sind Sie zwar nicht größenwahnsinnig, wohl aber haben Sie dann einen Minderwertigkeitskomplex.
- Falls Sie aber keine Zwangsvorstellungen haben, dann haben Sie Depressionen."

c) WHILE- vs. FOR-Schleife, die zweite (3 Punkte)

Überlegt Euch eine WHILE-Schleife, die sich (*mit den bislang erworbenen Kenntnissen*) Eurer Meinung nach nicht durch eine FOR-Schleife ersetzen lässt und erklärt Eurem Tutor, warum das (*vermeintlich*) nicht möglich ist!

d) DO-WHILE-Schleifen (2 Punkte)

In den Übungen wurden Euch als dritter Schleifentyp die DO-WHILE-Schleifen vorgestellt. Arbeitet nochmals den Unterschied zwischen ihnen und den WHILE-DO-Schleifen heraus und erklärt ihn Eurem Tutor!

Aufgabe 2: Macht Java krank? (15 Punkte)

In der Vorlesung habt Ihr etwas ausführlicher über den Datentyp `boolean` gesprochen. Bislang kannten wir ihn nur, um ihn z.B. in WHILE-Schleifen oder der IF-Bedingung zu verwenden.

Andererseits habt Ihr im ersten Semester aber auch die Wahrheitstafel und Umformungsregeln kennengelernt (Skript-Teil 6: „Aussagenlogik", Folie 12) und wisst daher, wie man mit logischen Ausdrücken umgeht.

In dieser Aufgabe kombiniert Ihr nun beides!

Schaut Euch den Kasten oben auf dieser Seite an. Schreibt ein Java-Programm, dass durch Anwendung der im Kasten genannten Aussagen eine korrekte Diagnose stellt, in dem es ausgibt, unter welcher Kombination der 5 möglichen unser armer Student leidet. (Es ist aufgrund der gegebenen Aussagen nur eine Diagnose - d.h. gültige Kombination der Krankheiten - möglich!)

Geht dabei wie folgt vor:

- Formuliert mit Hilfe der Wahrheitstafeln des ersten Semesters die gegebenen Aussagen so um, dass sich jede Aussage als Bool'scher Ausdruck schreiben lässt (Tip: „*Kontraposition*")!
- Legt für jede der möglichen Krankheiten eine bool'sche Variable an (Motto: entweder er hat die Krankheit oder er hat sie nicht)!
- Schafft mit Hilfe „geeignet verschachtelter Schleifen" die Möglichkeit, *alle* Kombinationen von Krankheiten genau einmal zu erreichen!
- Prüft für jede dieser Kombinationen jeweils nach, ob jede der im Kasten oben genannten Bedingungen dabei erfüllt ist. Sobald (mindestens) eine *nicht* erfüllt ist, kann die aktuelle Kombination keine gültige Diagnose mehr sein!
- Gebt - wenn alle Bedingungen erfüllt sind - die aktuelle Diagnose aus, in dem Ihr die vorliegenden Krankheiten auflistet!

Gute Besserung!