



Technische Universität Darmstadt
 Fachbereich Informatik
 Prof. Dr. Johannes Fürnkranz

Allgemeine Informatik 1 im WS 2006/07

Übungsblatt 13

Bearbeitungszeit: 05.02. bis 11.02.2007

Aufgabe 1: Scope von Variablen

Betrachten Sie das folgende Programm genau. Nehmen sie an, dass mit `Tester test = new Tester (42, 23);` eine Instanz dieser Klasse angelegt wird und anschließend mit `test.m();` aufgerufen wird. Welchen Wert haben die einzelnen Attribute von `test` nach dem Ende des Aufrufs?

Achtung: Diese Aufgabe mit Hilfe von KarelJ zu lösen ist nicht gestattet! In Ihrem eigenen Interesse - halten Sie sich bitte daran. In der Klausur hilft Ihnen auch kein PC!
 Sie können allerdings Ihre auf dem Papier entstandene Lösung mit KarelJ verifizieren.

```
class Tester {

    int x, y, z = 0;

    public Tester(int a, int z) {
        x = a;
        y = z;
        z = x + y;
    }

    public void m() {
        int v, w, x = 19;
        v = f(3) * z;
        w = y / x - f(f(z + x));
        x = x + f(y);
        y = x - v;
        z = f(v) - f(w) + f(z);
    }

    public int f(int x) {
        return z / y + x;
    }
}
```

Aufgabe 2: Restklassenverfahren

Wie Sie wissen, gibt es nicht nur das Dezimalsystem, sondern auch andere Zahlensysteme. So ist die Zahl 255_{10} im Dezimalsystem äquivalent zu 11111111_2 im Binär-, 100110_3 im Ternär-, 377_8 im Oktal- und FF_{16} im Hexadezimalsystem.

Ihre Aufgabe ist es nun, gemäß folgenden Vorgaben ein Programm zu schreiben, das Zahlen aus dem Dezimalsystem in ein beliebiges System zur Basis 2 bis 36 umwandeln und auf dem Bildschirm ausgeben kann. Lesen Sie sich die komplette Aufgabe durch, bevor Sie anfangen!

- a) Erstellen Sie eine Roboterklasse **Converter** mit zwei Attributen **int base** und **String digits**.
- b) Schreiben Sie einen Konstruktor, der nur einen Parameter **int base** erwartet. Erzeugen Sie zuerst mittels **super** einen beliebigen Roboter erzeugt werden und füllen Sie **digits** mit „0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ“. Ihr Programm soll dann überprüfen, ob der Parameter im erlaubten Bereich [2, 36] liegt. **Falls nein**, soll auf dem Bildschirm eine entsprechende Fehlermeldung ausgegeben werden und das Klassenattribut **base** mit 10 initialisiert werden. **Falls ja**, soll das Attribut mit dem übergebenen Parameter initialisiert werden.
*Hinweis: auf das Klassenattribut müssen Sie mit **this.base** zugreifen, sonst greifen Sie auf den Parameter zu!*
- c) Schreiben Sie eine Methode **convert**, die einen Parameter **int number** erwartet und einen **String** zurückgibt. Erzeugen Sie zuerst einen leeren String. Mit dem in der Vorlesung kennen gelerntem (und im Programmierprojekt verwendeten) Verfahren berechnen Sie Ziffer um Ziffer die Zahlendarstellung zur Basis **base**. Zu jeder neu gewonnenen Ziffer brauchen Sie die korrekte Darstellung im neuen Zahlensystem, also das entsprechende Zeichen in **digits**. Auf das Zeichen Nummer **i** in diesem String können Sie mit **digits.charAt(i)** zugreifen. Diesen Ausdruck können Sie in einer Variable vom Typ **char** zwischenspeichern oder direkt weiterverwenden. Jede neu gewonnene Ziffer hängen Sie nun **vorne** an den String an. Um Strings oder andere Variablen an Strings zu hängen, verbinden Sie die Variablen in der richtigen Reihenfolge mit dem Operator **+**. Zuletzt muss der String, in dem nun (hoffentlich) das richtige Ergebnis steht, mittels **return** zurückgegeben werden.
- d) Schreiben Sie eine Methode **void out**, die einen Parameter **int number** erwartet. Diese soll einen Text im folgenden Format auf dem Bildschirm ausgeben:
38 dezimal = **123** zur Basis **5**
Die fett gedruckten Zahlen müssen Sie natürlich durch Variablen bzw. einen Methodenaufruf ersetzen. Vergessen Sie die entsprechenden Leerzeichen nicht!
- e) Testen Sie Ihre Methode, z.B. mit den in dieser Aufgabe angegebenen Zahlen.

Aufgabe 3: Array umkehren

Fügen Sie der Klasse **Ascending** aus Aufgabe 4 in Übung 12 (Sie müssen dazu die Aufgabe nicht gelöst haben, die Datei **uebung12-4.task** von der Webseite reicht aus) eine Methode **void reverse()** hinzu.

Diese soll das Array umkehren, also den ersten Wert mit dem letzten vertauschen, den zweiten mit dem vorletzten etc.

Testen Sie die Methode, indem Sie dem **task** einen entsprechenden Funktionsaufruf hinzufügen.

Aufgabe 4: Rekursion

Platzieren Sie in einem **task** eine senkrechte Wand, die über die erste Straße geht, bei einer beliebigen Avenue > 1 . Schreiben Sie nun eine Roboterklasse, die eine Methode **int distanceToWall()** enthält. Diese Methode soll rekursiv(!!!) herausfinden und zurückgeben, wie viele Schritte es bis zur Wand sind. Der Roboter soll nach dem Aufruf der Methode wieder an der Ausgangsposition stehen! Sie können davon ausgehen, dass er westlich der Wand steht und nach Osten schaut.

Zum Test erzeugen Sie im **task** dann einen Roboter dieser Klasse auf der ersten Straße in einer beliebigen Avenue westlich der Wand mit 9999 Bepern und Blickrichtung Osten. Legen Sie mit Hilfe der Methode an der Ausgangsposition des Roboters so viele Beeper, wie es Schritte bis zur Wand sind. Sorgen Sie dann dafür, dass der Roboter vom Beeperstapel verschwindet (**turnLeft(); move();**).

Klausur

Die Klausur Allgemeine Informatik 1 im Wintersemester 2006/2007 findet am Donnerstag, den 8. März 2007 vormittags statt. Der Ort und die genaue Zeit stehen noch nicht fest.

In den nächsten Tagen wird es eine (verpflichtende!) Online-Anmeldung geben. Das sowie die genaue Zeit der Klausur wird dann auf der Webseite und im Forum angekündigt.