SAARLAND UNIVERSITY
COMPUTER SCIENCE

Arbeitsgruppe Programmierung
Jun.-Prof. Dr. Sebastian Hack
Dipl.-Inform. Christoph Mallon

Programmierung 2 — Übungsblatt 2

Abgabe: **Dienstag, 12. Mai 2009, 11.45 Uhr, Geb. E1 3, Briefkasten im EG**
Lösung mit **Name**, **Matrikelnummer** und **Name des Tutors** beschriften!

Aufgabe 2.1: Klassen, Konstruktoren, Methoden, Kapselung

Erstellen Sie eine Klasse `Vector3D` die einen dreidimensionalen Vektor repräsentiert.

- Für die Typen der Felder verwenden Sie `double`, dessen Werte Gleitkommazahlen sind.
- Erstellen Sie zwei Konstruktoren:
  1. `public Vector3D()` setzt alle Komponenten auf `0.0`.
  2. `public Vector3D(double x, double y, double z)` kopiert die Werte der formalen Parameter in die entsprechenden Komponenten
- Kapseln Sie die Felder und implementieren Sie Getter für jede Komponente.
- Implementieren Sie eine Methode

  ```
  public void scale(double factor) { ... }
  ```

  die den Vektor skaliert, d.h. alle Komponenten mit `factor` multipliziert.
- Implementieren Sie eine Methode

  ```
  public double dot(Vector v) { ... }
  ```

  die das Skalarprodukt

  $$x \times x_v + y \times y_v + z \times z_v$$

  des Vektors mit dem Vektor v berechnet und zurückgibt.
- Implementieren Sie eine Methode

  ```
  public String toString() { ... }
  ```

  die aus dem Vektor eine Zeichenkette erzeugt. Diese soll den Vektor wie folgt darstellen:

  ```
  [ x-Koordinate, y-Koordinate, z-Koordinate ]
  ```

  Hinweis: `toString()` gibt nur einen `String` zurück, gibt aber selbst keinen Text aus.
- Erstellen Sie eine Klasse VectorTest mit der Methode

  ```
  public static void main(String[] args) { ... }
  ```

  Implementieren Sie den Rumpf der Methode wie folgt:
  - Legen Sie zwei Objekte der Klasse Vektor an.
  - Geben Sie die beiden Vektoren auf dem Bildschirm aus.
  - Berechnen Sie das Skalarprodukt der beiden Vektoren und speichern Sie es in einer lokalen Variable.
  - Geben Sie diese auf dem Bildschirm aus.

Aufgabe 2.2: Objekte und Referenzen

Was gibt folgendes Programm aus? Zeichnen Sie den Objektgraphen.

```
class A {
  public String a;
  public B b;
  public static String c;
}
class B {
  public A a;
  public String b;
}
public class Main {
  public static void main(String[] args) {
    A a = new A();
    System.out.println(a.a);
    a.a = "Hallo";
    A.c = a.a;
    System.out.println(a.b);
    B b = new B();
    b.a = a;
    System.out.println(b.a.b);
    b.a.b = b;
    System.out.println(b.a.a);
    b.b = a.a;
    System.out.println(b.b);
    a.a = "Tschuess";
    System.out.println(b.b);
    System.out.println(a.c);
    b.a = new A();
    A.c = b.a.a;
    System.out.println(a.b.a.b);
    System.out.println(a.c);
    b.b = a.a;
    System.out.println(b.b);
  }
}
```

Hinweis: Nur übersetzen und ausführen ist möglich, aber nicht sinnvoll.

Aufgabe 2.3: Ausdrücke und Typanpassung

1. `++x;` erhöht den Wert einer Variablen um eins. Der Versuch dies ausführlicher mit `x = x + 1;` für eine Variable des Typs `short` zu umschreiben wird vom Java-Übersetzer mit der Fehlermeldung „possible loss of precision“ quittiert. Warum? Wie kann dies behoben werden?
2. Gegeben sind drei Variablen `short a, b, c;`. Der Übersetzer meldet bei dem Ausdruck `a = b | c;` ebenfalls einen „possible loss of precision“. Kann dies je auftreten? Begründen Sie Ihre Antwort kurz.

Aufgabe 2.4: Anweisungen

1. In der Vorlesung wurde `while (b) S` mithilfe von `if` beschrieben. Setzen Sie nun umgekehrt `if (b) S1 else S2` mittels `while` um.

   Hinweis: Die Bedingung `b` darf nicht mehrfach ausgewertet werden, denn sie könnte Seiteneffekte enthalten.
2. Die Anweisung `break;` beendet umgehend die Ausführung der sie umgebenden Schleife und setzt die Ausführung nach der Schleife fort. Vereinfachen Sie damit die Lösung der vorigen Teilaufgabe.