SAARLAND UNIVERSITY
COMPUTER SCIENCE

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

Programmierung 2 — Übungsblatt 4

Abgabe: **Dienstag, 26. 5 2009, 11.45 Uhr, Geb. E1 3, Briefkasten im EG**
Lösung mit **Name**, **Matrikelnummer** und **Name des Tutors** beschriften!

Aufgabe 4.1: Anweisungen, Schleifen: Approximation der Quadratwurzel

Mit folgendem iterativen Verfahren kann die Quadratwurzel $\sqrt{S}$ einer Zahl S angenähert werden:

$$\begin{aligned} x_0 &= \text{beliebiger Startwert} \\ x_{n+1} &= \frac{1}{2}\left(x_n + \frac{S}{x_n}\right) \end{aligned}$$

Implementieren Sie eine Methode `double sqrt(double s)`, welche die Quadratwurzel einer Zahl näherungsweise berechnet. Iterieren Sie solange, bis sich zwei aufeinander folgende Reihenglieder nicht um mehr als `EPS` unterscheiden.

```
class SquareRoot {
  public static final double EPS = 1e-10;

  public static double sqrt(double s) {
    // Your implementation goes here
  }

  public static void main(String[] args) {
    // Test your function here
  }
}
```

Bemerkung: Je näher x_0 an $\sqrt{S}$, desto schneller konvergiert das Verfahren. Wählen Sie den Startwert also in Abhängigkeit von S.

Aufgabe 4.2: Aufgabe 4.2: Anweisungen, Schleifen: Sieb des Eratosthenes

Das Sieb des Eratosthenes berechnet alle Primzahlen von 2 bis zu einer gegebenen Zahl S: Alle Zahlen von 2 bis S werden auf ein Papier geschrieben. Iterativ werden nun alle Zahlen, die keine Primzahlen sind, durchgestrichen. Dies funktioniert so: Nimm die kleinste, noch nicht gestrichene Zahl. (In der ersten Iteration ist das die 2.) Die Zahl selbst ist eine Primzahl. Streiche alle echten Vielfachen (alle $k \times$ Zahl für $k > 1$) dieser Zahl. Gehe weiter zur nächsten, nicht gestrichenen Zahl. Diese ist wiederum eine Primzahl, streiche alle ihre echten Vielfachen, usw. Schreiben Sie eine Methode `int[] primes(int s)`, die eine Reihung zurück liefert, die alle Primzahlen von 2 bis einschließlich `s` enthält.
Bemerkung: Betrachten wir eine beliebige Iteration des Siebs des Eratosthenes. Sei z die Zahl dieser Iteration. Mit dem Streichen der Vielfachen von z kann bei z^2 begonnen werden, denn alle Vielfachen von z in $[z+1, ..., z^2[$ wurden durch vorherige Iterationen schon gestrichen.

Aufgabe 4.3: Verallgemeinerung und Vererbung

Betrachten Sie nochmals die Klasse `Vector3D` vom zweiten Übungsblatt. Erstellen Sie eine neue Klasse `Vector`, die Vektoren beliebiger Dimension implementiert. Dementsprechend hat die Klasse nur noch einen Konstruktor `Vector(int n)`, der die gewünschte Dimension entgegennimmt und den Vektor als Nullvektor initialisiert. Implementieren Sie analog die Methoden `dot()` und `scale()`. Fügen Sie Getter/Setter hinzu, die jeweils das i-te Element des Vektors liefern/setzen. Wie in der Informatik üblich, soll das erste Element mit dem Index 0 angesprochen werden. Fügen Sie des Weiteren eine Methode `getDim()` hinzu, die die Dimension des Vektors liefert. Die Methode `double dot(Vector v)` soll zusätzlich noch folgende Bedingung erfüllen: Falls die Dimension von `v` von der von `this` abweicht, sollen die nicht vorhandenen Elemente wie `0.0` behandelt werden.

Reimplementieren Sie nun `Vector3D` als Unterklasse von `Vector`. Verwenden Sie, soweit möglich, die Implementierung der Oberklasse wieder. Alle bisheringen Methoden von `Vector3D` sollen auch weiterhin zur Verfügung stehen.

Aufgabe 4.4: Statischer und dynamischer Typ, polymorpher Methodenaufruf

Was gibt folgendes Programm aus? Können Sie eine der `bar()`-Methoden allgemeingültig vereinfachen?

```
class A {
      void foo() { System.out.println("A"); }
      void bar() { System.out.println(this instanceof B); }
}

class B extends A {
      void foo() { System.out.println("B"); }
}

class C extends A {
      void foo() { System.out.println("C"); }
      void bar() { System.out.println(this instanceof A); }
}

class Main {
  public static void main(String[] args) {
    B b = new B();
    C c = new C();
    A a = new A();
    a.foo();
    a.bar();
    b.foo();
    b.bar();
    c.foo();
    c.bar();
    a = b;
    a.foo();
    a.bar();
    a = c;
    a.foo();
    a.bar();
  }
}
```