Einführung in das Programmieren

Prüfungsvorbereitung Theorie

Andreas Krall

Prüfungsablauf

Prüfungsdauer:	90 Minuten
Prüfungsart:	schriftlich
Unterlagen:	erlaubt (Buch, Folien, Beispiele)
Computer, Handy:	verboten
Identifikation:	amtlicher Lichtbildausweis (Studentenausweis)
Bewertung:	Multiplikation mit Praxisteil
Tips:	zu Beginn Angaben aller Beispiele lesen
Beispielarten:	Konstruktion und Ablauf

Design einer Klasse Box

Die Parameter einer Schachtel sind Länge, Breite und Höhe.

Damit lassen sich drei Merkmale folgendermaßen berechnen:

Grundfläche: Länge * Breite

Volumen: Länge*Breite*Höhe

Oberfläche: 2*(Länge*Breite+Länge*Höhe+Breite*Höhe)

Schreiben Sie eine Klasse Box, deren Instanzvariablen die oben beschriebenen Parameter abbilden (double), einen geeigneten Konstruktor, sowie die Methoden zur Berechnung der weiteren Merkmale.

```
public class Box
{
   private double länge;
   private double breite;
   private double höhe;

   public Box(double l, double b, double h) {
      länge = l; breite = b; höhe = h;
   }
   public double grundFläche() {
      return länge * breite;
   }
   public double volumen() {
      return grundFläche() * höhe;
   }
   public double oberFläche() {
      return 2 * (grundFläche() + länge * höhe +
                                  breite * höhe);
   }
}
```

Programm `BoxTester`

Um das Design zu Überprüfen, schreiben Sie ein Programm `BoxTester` welches eine Schachtel mit den Werten 2.5, 5.0 und 6.0 erzeugt und anschliessend die Grundfläche, Volumen und die Oberfläche ausgibt.

```
import eprog.*;

public class BoxTester extends EprogIO
{
   public static void main (String[] args) {
      Box b = new Box (2.5, 5.0, 6.0);

      println ("Grundfläche = " + b.grundFläche());
      println ("Volumen =     " + b.volumen());
      println ("Oberfläche =  " + b.oberFläche());
   }
}
```

```
public class SimpleMatrix {
   public SimpleMatrix (int rows, int columns){…}
   public int getRows () {…}
   public int getColumns () {…}
   public void setElement (int row, int column, int
      val) throws OutOfMatrixDimensionException {…}
   public int getElement (int row, int column)
      throws OutOfMatrixDimensionException {…}
   public void add (SimpleMatrix matrix)
      throws MismatchMatrixDimensionsException {…}
   public void print () {…}
}
```

```
public class Matrix extends SimpleMatrix {

   public void subtract (Matrix matrix) throws
      MismatchMatrixDimensionsException {

      if (getRows() != matrix.getRows() ||
          getColumns() != matrix.getColumns())
        throw MismatchMatrixDimensionsException;

      for (int r = 0; r < getRows(); r++)
        for (int c = 0; c < getColumns(); c++)
          setElement(r, c, getElement(r, c) -
                    matrix.getElement(r, c));
   }
```

```
   public Matrix multiply (Matrix matrix) throws
      MismatchMatrixDimensionsException {
      Matrix result =
         new Matrix (getRows(), matrix.getColumns());

      if (getColumns() != matrix.getRows())
         throw MisMatchMatrixDimensionsException;

      for (int r=0; r < getRows(); r++)
         for (int c=0; c< matrix.getColumns(); c++) {
            int sum = 0;
            for (int i = 0; i < getColumns(), i++)
               sum += getElement(r, i) *
                     matrix.getElement(i, c)
            result.setElement(r, c, sum);
         }
      return result;
   }
}
```

```
public class A extends EprogIO {
   public a(int i)    {println("ia von A");}
   public a(float f)  {println("fa von A");}
}
public class B extends A {
   public a(int a)    {println("ia von B");}
   public a(String a) {println("sa von B");}
}
public class Test extends EprogIO {
   public static void main (String[] args) {
      A a1 = new A();
      B b1 = new B();
      A a2 = b1;
      b1.a(3); b1.a("3");
      ((A)b1).a(3);b1.a(3.0);
      a1.a(3); a1.a(3.0);
      a2.a(3); a2.a(3.0);
   }
}
```

```
public class A {
   public int i;
}
public class B extends A {
   public int i;
   public void set (int i) {this.i = i;}
}
public class Test extends EprogIO {
   public static void main (String[] args) {
      A a1 = new A();
      B b1 = new B();
      A a2 = b1;
      a1.i = 1; println(a1.i+","+a2.i+","b1.i);
      a2.i = 2; println(a1.i+","+a2.i+","b1.i);
      b1.set(3); println(a1.i+","+a2.i+","b1.i);
      ((A)b1).i = 4; println(a1.i+","+a2.i+","b1.i);
      ((B)a2).i = 5; println(a1.i+","+a2.i+","b1.i);
   }
}
```

```
public class Test extends EprogIO {
   public static void main (String[] args) {
      println(3 + 4 + " = 34");
      println(3 + (4 + " = 34"));
      println(34 + " = 34");
      println("34 = " + 3 + 4);
      println("" + 3 + 4 + " = 34");
      println("34" + (3 + 4));
   }
}
```

```
public class A {
   int a;
   protected int b;
   private int c;
// a, b, c sichtbar
}
public class B extends A {
   public int d;
// a, b, d sichtbar
}
public class Test extends EprogIO {
   public static void main (String[] args) {
      A a1 = new A();
      B b1 = new B();
// a1.a, a1.b sichtbar
// b1.a, b1.b, b1.d sichtbar
   }
}
```

```
public class Test extends EprogIO {
   // Setzen Sie Ihre eigene Matrikelnummer ein
   static final String mat = "1234567";
   public static void main (String[] args) {
      char cmat [][] = new char [2][];
      cmat[0] = mat.toCharArray();
      cmat[1] = new char [7];
      for (int i = 0; i < 7; i++)
         cmat[1][i] = cmat[0][i] < cmat[0][6-i]
            ? cmat[0][i] : cmat[0][6-i];
      for (int i = 0; i < 7; i++)
         print(cmat[1][i]);
   }
}
```