

Übungsblatt 10: Objektorientierte Programmierung

Ausgabe: 20.12.2013

Abgabe: 10.01.2014

Im Folgenden ist eine Klasse mit dem Namen `PSpicture` gegeben, welche Methoden zum Erstellen eines PostScript-Bildes enthält. Einen kurzen Überblick über die Methoden erhalten Sie hier:

`void drawLine(int,int,int,int)` zeichnet eine Linie von einem Startpunkt zu einem Endpunkt

`void continueLine(int,int)` zeichnet eine Linie vom letzten verwendeten Punkt zu einem Endpunkt

`void drawCircle(Circle)` zeichnet einen Kreis

`String toString()` gibt die PostScript-Kommandos des Bildes aus

Wenn die PostScript-Kommandos in eine Datei (möglichst mit der Endung `.ps`) gespeichert werden, kann das Bild mit einem PostScript-Viewer (z. B. Ghostview) betrachtet werden.

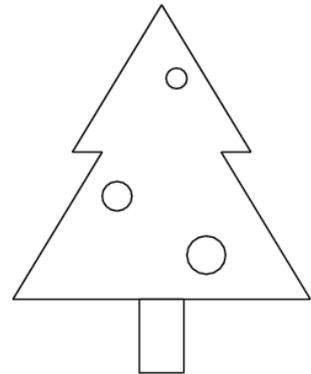
```
1 public class PSpicture
2 {   private String picture;
3     private int lastX, lastY;
4
5     public PSpicture()
6     {   picture = "";
7         lastX = -1; lastY = -1;
8     }
9     public void drawLine(int x1, int y1, int x2, int y2)
10    {   picture += x1 + " " + y1 + " moveto" + (x2-x1) + " " + (y2-y1) +
11        " rlineto\nstroke\n";
12        lastX = x2; lastY = y2;
13    }
14    public void continueLine(int x2, int y2)
15    {   if(lastX >=0 && lastY >= 0)
16        drawLine(lastX, lastY, x2, y2);
17        else
18            System.out.println("FEHLER: Kann Linie nicht fortsetzen, " +
19                                "da letzter Punkt unbekannt ist!");
20    }
21    public void drawCircle(Circle circ)
22    {   picture += circ.getX() + " " + circ.getY() + " " +
23        circ.getRadius() + " 0 360 arc\nstroke\n";
24    }
25    public String toString()
26    { return (picture + "showpage\n"); }
27    // weitere Methoden ...
28 }
```

Damit wir einen kleinen Weihnachtsbaum zeichnen können, fehlen uns noch einige Klassen und Methoden. Unser Ziel ist mit den links dargestellten Anweisungen das rechts angegeben Bild zu zeichnen:

```

1 PSpicture picture = new PSpicture(600,400);
2 picture.drawLine(200,300,400,300);
3 picture.continueLine(340,400);
4 picture.continueLine(360,400);
5 picture.continueLine(300,500);
6 picture.continueLine(240,400);
7 picture.continueLine(260,400);
8 picture.continueLine(200,300);
9 Circle circ = new Circle(330, 330, 13);
10 picture.drawCircle(circ);
11 circ.move(-60, 40);
12 circ.scale(0.8);
13 picture.drawCircle(circ);
14 circ.move(40, 80);
15 circ.scale(0.7);
16 picture.drawCircle(circ);
17 Rectangle rect = new Rectangle(285, 250, 30, 50);
18 picture.drawRectangle(rect);

```



Aufgabe: PSpicture ((2 + 5) + (5 + 2 + 2, Bonuspunkte))

1. Teil: Die Klasse PSpicture besitzt leider noch keine Kommentare. Erstellen Sie mindestens für jede Methode sinnvolle und aussagekräftige Kommentare!

2. Teil: Erstellen Sie eine Klasse Circle, welche als private Attribute die x- und y-Koordinate sowie einen Radius (jeweils vom Typ int) besitzt. Es gibt genau einen Konstruktor, welcher alle benötigten Daten (in der angegebenen Reihenfolge) als Parameter erhält. Schreiben Sie des Weiteren folgende Methoden:

- getX, getY und getRadius, welche jeweils die x-, y-Koordinate bzw. den Radius zurückgeben
- move welche als Parameter eine Verschiebung (Differenz zwischen neuer und aktueller Koordinate) in x-Richtung und in y-Richtung erhält und entsprechend den Kreis verschiebt
- scale eine Methode welche einen Faktor (Typ double) erhält und den Kreis entsprechend skaliert (es darf dabei abgerundet werden!)

3. Teil: Erstellen Sie eine Klasse Rectangle, welche als private Attribute die x- und y-Koordinate sowie Breite und Höhe (jeweils vom Typ int) besitzt. Es gibt genau einen Konstruktor, welcher alle benötigten Daten (in der angegebenen Reihenfolge) als Parameter erhält. Schreiben Sie des Weiteren folgende Methoden:

- getX, getY, getWidth und getHeight welche jeweils die x-, y-Koordinate, die Breite bzw. die Höhe zurückgeben
- move welche als Parameter eine Verschiebung in x-Richtung und in y-Richtung erhält und entsprechend das Rechteck verschiebt
- scale welche einen Faktor (Typ double) erhält und das Rechteck entsprechend skaliert (es darf dabei abgerundet werden!)

4. Teil: Fügen Sie nun der Klasse `PSpicture` eine Methode `drawRectangle` hinzu, welche als Parameter ein Objekt der Klasse `Rectangle` erhält und das Rechteck zeichnet. Zeichnen bedeutet in diesem Fall, dass die entsprechenden PostScript-Kommandos erzeugt und dem Bild hinzugefügt werden. Dazu dürfen Sie bereits vorhandene Methoden der Klasse `PSpicture` verwenden und bei Bedarf weitere Methoden hinzufügen.

5. Teil: Erstellen Sie nun eine Klasse `PSpictureTest` welche als Konsolenparameter einen Dateinamen erhält, und in dieser Datei mittels der `PSpicture`-Klasse die PostScript-Kommandos aus unserem Beispiel schreibt. Wenn Sie die erzeugte Datei mit einem PostScript-Viewer öffnen, sollten Sie den Weihnachtsbaum aus unserem Beispiel sehen.

Senden Sie die Java-Dateien zu den Klassen `PSpicture`, `PSpictureTest`, `Circle` und `Rectangle` in einer ZIP-Datei ein!

Viel Erfolg und frohe Weihnacht!