



Einführung in die Informatik

für Hörer alle Fakultäten

(WS-03/04)

Übungsblatt 7

Ausgabe: Do. 2003-12-04

Abgabe: Do. 2003-12-11

Besprechung: Di./Do. 2003-12-16/18

Beachten Sie auch für dieses Aufgabenblatt die Hinweise auf Übungsblatt 4 zur Abgabe der Lösungen mit Praktomat unter:

<https://jop.informatik.uni-wuerzburg.de/infohaf/praktomat.cgi>

Aufgabe 7.1: Vererbte Figuren (6 Punkte)

Gegeben ist das Interface Figure71:

```
/**
 * Das Interface Figure definiert die gemeinsame
 * Schnittstelle für verschiedene geometrische Figuren.
 * @author fischer
 * @version 1.0
 */
public interface Figure71 {
    /**
     * Berechnet die Fläche der geometrischen Figur.
     * @return Die Fläche der Figur.
     */
    public double area();

    /**
     * Berechnet den Umfang der Figur, d.h. die Länge der "Außenkanten".
     * @return Der Umfang der Figur.
     */
    public double circumference();

    /**
     * Berechnet, ob die Figur den Punkt mit den angegebenen Koordinaten
     * überdeckt.
     * @param x Die x-Koordinate des zu überprüfenden Punktes.
     * @param y Die y-Koordinate des zu überprüfenden Punktes.
     * @return true wenn die Figur den Punkt überdeckt,
     *         false sonst.
     */
    public boolean overlaps(double x, double y);
}
```

```

/**
 * Liefert einen, die Figur beschreibenden String, mit dem Namen der Figur
 * und den wichtigsten Attributen.
 * @return Eine textuelle Beschreibung der Figur.
 */
public String toString();

/**
 * Setzt die Farbe der Figur mit den angegebenen RGB Werten.
 * @param red Rot-Wert der Farbe (Wertebereich: 0 bis 255)
 * @param green Grün-Wert der Farbe (Wertebereich: 0 bis 255)
 * @param blue Blau-Wert der Farbe (Wertebereich: 0 bis 255)
 */
public void setColor(int red, int green, int blue);

/**
 * Liefert den Rot-Wert der Farbe der Figur
 * @return der Rot-Wert der Farbe (0-255)
 */
public int getColorRed();

/**
 * Liefert den Grün-Wert der Farbe der Figur
 * @return der Grün-Wert der Farbe (0-255)
 */
public int getColorGreen();

/**
 * Liefert den Blau-Wert der Farbe der Figur
 * @return der Blau-Wert der Farbe (0-255)
 */
public int getColorBlue();
}

```

Erstellen Sie eine öffentliche abstrakte Klasse `Figure71Impl`, die das Interface `Figure71` implementiert. Dabei sollen nur die Methoden `getColorRed()`, `getColorGreen()`, `getColorBlue()` und `setColor(int red, int green, int blue)` implementiert werden, die restlichen Methoden aus `Figure71` bleiben abstrakt.

Erstellen Sie weiterhin die Klassen `Circle71` und `Rectangle71` die von der Klasse `Figure71Impl` abgeleitet sind und alle noch benötigten (also bei `Figure71Impl` abstrakten) Methoden implementieren. Darüberhinaus soll die Klasse `Circle71` so geändert werden, dass ein Kreis auch eine "Randfarbe" (`borderColor`) hat, dessen Farbe sich aus der Farbe des Kreises ableitet. Und zwar soll es die Farbe sein, die sich ergibt, wenn man alle Farbwerte der normalen Farbe halbiert (und abrundet).

Zum Auslesen bietet die Klasse `Circle71` dann zusätzlich folgende Methoden an:

- `public int getBorderColorRed()`
liefert den Rot-Wert der RandFarbe.
- `public int getBorderColorGreen()`
liefert den Grün-Wert der RandFarbe.
- `public int getBorderColorBlue()`
liefert den Blau-Wert der RandFarbe.

Weiterhin erstellen Sie noch eine Klasse `Figure71Component`, die auch von `Figure71Impl` abgeleitet ist, die eine Zusammensetzung von Figuren repräsentiert. Dazu verwaltet die Klasse ein Array von `Figure71`'s. Die Methoden `circumference()` und `area()` sollen so implementiert werden, dass sie die Summe über alle enthaltenen Figuren bilden. Mögliche Überschneidungen der Figuren bleiben dabei unberücksichtigt. Die Methode `overlaps(double x, double y)` soll `true` liefern, wenn (mindestens) eine der Figuren diesen Punkt überdeckt, `false` nur dann, wenn keine der Figuren das tut.

Zur Verwaltung der Figuren in `Figure71Component` stellen Sie außerdem noch folgende Methoden zur Verfügung:

- `public Figure71Component()`
Erzeugt eine `Figure71Component` mit Platz für 10 `Figure71`-Objekte.
- `public Figure71Component(int maxSize)`
Erzeugt eine `Figure71Component` mit Platz für `maxSize`-viele `Figure71`-Objekte.
- `public void addFigure(Figure71 f)`
Fügt eine neue `Figure71` hinzu, sofern das Array nicht schon voll ist.
- `public Figure71 getFigure(int pos)`
Liefert die Figur an der Position `pos`.

Abschließend erstellen Sie noch eine Klasse `Figure71Test`, mit einer `main`-Methode, die die Funktionalität der anderen Klassen demonstriert bzw. testet. Diese ist auch als Hauptklasse abzugeben.

Aufgabe 7.2: Birthday (4 Punkte)

Erstellen Sie eine öffentliche Klasse `BirthDay` die einen Geburtstag repräsentiert. Verwenden Sie zur Implementierung die Klassen `java.util.Calendar`, `java.util.GregorianCalendar` und `java.util.Date` (das wird ihre Arbeit erheblich erleichtern!) Implementieren Sie (mindestens) folgende Methoden:

- `public BirthDay(Calendar cal)`
Konstruiert einen neuen Geburtstag, für den in `cal` gegebenen Tag.
- `public BirthDay(int year, int month, int day)`
Konstruiert einen neuen Geburtstag, für den angegebenen Tag. Dabei ist das Jahr (`year`) ungekürzt anzugeben, also z.B. 2003 für das aktuelle, der Monat (`month`) im Bereich 1-12, und der Tag (`day`) je nach Monat von 1-28, 1-29, 1-30 oder 1-31.
- `public long getDaysSinceBirthDay()`
Berechnet die Anzahl der Tage seit dem Geburtstag bis heute.
- `public Calendar getNextBirthDay()`
Liefert den nächsten Geburtstag (ab heute).
- `public long getDaysTillNextBirthDay()`
Berechnet die Anzahl der Tage bis zum nächsten Geburtstag.
- `public Calendar getNextRoundDay()`
Berechnet den nächsten "runden" Tag, d.h. den nächsten Tag, an dem die Anzahl der Tage seit der Geburt ein Vielfaches von 1000 ist.
- `public Calendar getBirthday()`
Liefert ein `Calendar`-Objekt für den Geburtstag zurück.
- `public static void main(String[] args)`
Eine `Main`-Methode, die die Funktionalität der Klasse `BirthDay` testet und demonstriert. Dies wird nicht automatisch geprüft, Sie sollten jedoch trotzdem eine sinnvolle Demonstration implementieren, da das bei der manuellen Abschlusskorrektur berücksichtigt wird. Darüber hinaus dient es für Sie selbst zur Überprüfung Ihrer Klassen und Methoden.