



Einführung in die Informatik

für Hörer alle Fakultäten

(WS-03/04)

Übungsblatt 8

Ausgabe: Do. 2003-12-11

Abgabe: Do. 2003-12-18

Besprechung: Di. 2003-12-23 / Do. 2004-01-08

Beachten Sie auch für die Aufgaben 8.1 und 8.2 dieses Aufgabenblatts die Hinweise auf Übungsblatt 4 zur Abgabe der Lösungen mit Praktomat unter:

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

Aufgabe 8.1: Kollekte (2 Punkte)

Erstellen Sie eine öffentliche Klasse `Kollekte` die eine Liste von `Number` (`java.lang.Number`) verwaltet. Sie können und sollen dazu intern eine Liste aus dem Java-Standard-Bibliothek verwenden, z.B. `java.util.LinkedList` oder `java.util.ArrayList`. Implementieren Sie (mindestens) folgenden Methoden:

- `public Kollekte()`
Konstruktor zur Erzeugung einer leeren Kollekte.
- `public void add(double d)`
Fügt der Kollekte einen neuen Wert hinzu. Dieser muss in ein `Double` gekapselt werden.
- `public void add(long l)`
Fügt der Kollekte einen neuen Wert hinzu. Dieser muss in ein `Long` gekapselt werden.
- `public double sum()`
Berechnet die Summe aller Elemente der Liste. (Hinweis: Verwenden Sie die Oberklasse `Number` von `Double` und `Long`.)
- `public void reverse()`
Kehrt die Reihenfolge der Elemente in der List um.
- `public List getList()`
Liefert die intern zur Implementierung verwendete Liste.
- `public static void main(String[] args)`
Eine Main-Methode, die die Funktionalität der Klasse `Kollekte` 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.

Aufgabe 8.2: Zeichnen (3 Punkte)

Gegeben ist das Interface PaintableFigure:

```
import java.awt.Graphics;
/**
 * Das Interface <code>PaintableFigure</code> definiert die gemeinsame
 * Schnittstelle für verschiedene geometrische Figuren, die sich zeichnen
 * lassen.
 * @author fischer
 * @version 1.0
 */
public interface PaintableFigure {
    /**
     * Diese Methode Zeichnet die Figur.
     * Zeichenbare Figuren überschreiben diese Methode mit dem Code,
     * der die Figur zeichnet.
     * @param g Graphic Kontext, auf dem die Figur gezeichnet werden soll.
     */
    public void paint(Graphics g);
}
```

Erstellen Sie dazu folgende öffentliche Klassen mit den angegebenen Methoden und Konstruktoren:

PaintableCircle (implementiert PaintableFigure):

- `public PaintableCircle()`
Konstruktor zur Erzeugung eines zeichenbaren Kreises.
- `public Color getColor()`
Liefert die aktuelle Farbe.
- `public void setColor(Color c)`
Setzt die aktuelle Farbe auf c.
- `public Color getBorderColor()`
Liefert die aktuelle Rahmenfarbe.
- `public void setBorderColor(Color c)`
Setzt die aktuelle Rahmenfarbe auf c.
- `public int getX()`
Liefert die Mittelpunktskoordinate X.
- `public void setX(int x)`
Setzt die Mittelpunktskoordinate X auf x.
- `public int getY()`
Liefert die Mittelpunktskoordinate Y.
- `public void setY(int y)`
Setzt die Mittelpunktskoordinate Y auf y.
- `public int getRadius()`
Liefert den Radius.
- `public void setRadius(int r)`
Setzt den Radius r.

PaintableRectangle (implementiert PaintableFigure):

- `public PaintableRectangle()`
Konstruktor zur Erzeugung eines zeichenbaren Rechtecks.
- `public Color getColor()`
Liefert die aktuelle Farbe.
- `public void setColor(Color c)`
Setzt die aktuelle Farbe auf c.
- `public Color getBorderColor()`
Liefert die aktuelle Rahmenfarbe.
- `public void setBorderColor(Color c)`
Setzt die aktuelle Rahmenfarbe auf c.
- `public int getX()`
Liefert die X-Koordinate der linke oberen Ecke.
- `public void setX(int x)`
Setzt die X-Koordinate der linke oberen Ecke auf x.

- `public int getY()`
Liefert die Y-Koordinate der linke oberen Ecke.
- `public void setY(int y)`
Setzt die Y-Koordinate der linke oberen Ecke auf `y`.
- `public int getWidth()`
Liefert die Breite.
- `public void setWidth(int width)`
Setzt die Breite auf `width`.
- `public int getHeight()`
Liefert die Höhe.
- `public void setHeight(int height)`
Setzt die Höhe auf `height`.

`PaintableFigureContainer` (erweitert `java.awt.Component`):

- `public PaintableFigureContainer()`
Konstruktor zur Erzeugung eines leeren Containers für `PaintableFigure`'s. Die `PaintableFigure`-Objekte sollen in einer Liste abgespeichert werden. Sie können (sollen) dazu Klassen aus der Java-Standard-Bibliothek `java.util` verwenden.
- `public void paint(Graphics g)`
Überschreiben Sie die Methode `paint()` so, dass sie alle im Container enthaltenen `PaintableFigure`'s zeichnet.
- `public void addPaintableFigure(PaintableFigure figure)`
Fügt dem Container eine neue `PaintableFigure` hinzu.
- `public List getPaintableFigureList()`
Liefert eine Liste der im Container enthaltenen `PaintableFigure`'s.
- `public Iterator iterator()`
Liefert einen Iterator für die im Container enthaltenen `PaintableFigure`'s. (Hinweis: Verwenden Sie den Iterator der Liste die Sie verwalten!)

`FigureFrame` (erweitert `java.awt.Frame`):

- `public FigureFrame()`
Konstruktor, der ein `java.awt.Frame` initialisiert, so dass das Fenster die Größe 640x480, den Titel "FigureFrame" und die Möglichkeit es zu schließen hat. Weiterhin soll dem Fenster ein `PaintableFigureContainer` als zu zeichnende Komponente hinzugefügt werden und das Fenster dargestellt werden.
- `public void addPaintableFigure(PaintableFigure figure)`
Fügt dem im Frame enthaltenen `PaintableFigureContainer` eine weitere `PaintableFigure` hinzu.
- `public PaintableFigureContainer getContainer()`
Liefert den im Frame enthaltenen `PaintableFigureContainer`.
- `public static void main(String[] args)`
Die Main-Methode soll die Funktionalität Ihrer Klassen testen und demonstrieren. Dazu wird ein `FigureFrame` erzeugt und mindestens drei `PaintableFigures` eingefügt.

Es ist die Hauptklasse `FigureFrame`, sowie die Klassen `PaintableFigureContainer`, `PaintableCircle` und `PaintableRectangle` abzugeben. Das Interface `PaintableFigure` kann, muss aber nicht mit abgegeben werden.

Abgabe der Aufgaben 8.3 und 8.4:

Bitte geben Sie die Aufgaben 8.3 und 8.4 bis zum

18.12.2003, 12.00 Uhr

schriftlich im mit dem Vorlesungstitel bezeichneten Briefkasten des Lehrstuhls Informatik V vor Raum AH001 (Informatikgebäude) ab.

Bitte geben Sie auf den Lösungen Ihren Namen und Matrikelnummer an.
Es sind maximal zwei Personen pro Lösungsblatt erlaubt.

Informationen zu Vorlesung und Übung, sowie Script- und Übungsblattdownload unter:

<http://www5.informatik.uni-wuerzburg.de/eihaf>

Fragen zu den Aufgaben 8.3 und 8.4 an:

Dipl. Inform. Holger Englert <englert@informatik.uni-wuerzburg.de>

Zimmer BH016, Informatikgebäude

Aufgabe 8.3: Zeichen und Worte (3 Punkte)

Gegeben seien die beiden folgenden Worte über dem Alphabet $A = \{ „ „, a, b, c, \dots, z \}$:

$u =$ „informationsrepraesentationsverarbeitung“

$v =$ „arbeit“

- Stellen Sie das Wort v durch eine geeignete Konkatenation von Zeichen des Wortes u dar.
- Bestimmen Sie:
 $\text{sstr}(u, 3, 5)$, $\text{sstr}(u, 5, 3)$, $|\text{sstr}(u, 5, 3)|$, $|\text{sstr}(u, 3, 5)|$ und $\$|\text{conc}(\text{sstr}(u, 31, 36), \emptyset)$
- Ist v ein Infix von u ?
- Ist $w = \text{„info“}$ ein Infix von u ?
- Zeigen Sie, dass die folgende Aussage für beliebige Worte x und y **falsch** ist:

$$x \text{ Präfix, Infix und Suffix von } y \Rightarrow x = y$$

Warum ist die Umkehrung dieser Folgerung jedoch richtig?

Aufgabe 8.4: Kodierungen (2 Punkte)

Sei $A = \{a_0, \dots, a_{n-1}\}$ ein beliebiges Alphabet. Bei welchen der folgenden Zuordnungen handelt es sich um Kodierungen? Geben Sie jeweils entweder einen Beweis oder ein Gegenbeispiel an!

a) $c(a_i) = u^{-1}(i)$ ($c(a_0) = 0$, $c(a_1) = 1$, $c(a_2) = 10$, $c(a_3) = 11$, $c(a_4) = 100$, ...)

(Binärkodierung ohne führende Nullen)

b) $c(a_i) = 1^{i+1}$ ($c(a_0) = 1$, $c(a_1) = 11$, $c(a_2) = 111$, ... , $c(a_{n-1}) = \underbrace{1 \dots 1}_{n\text{-mal}}$)

c) $c(a_i) = 1^{i+1}0$ ($c(a_0) = 10$, $c(a_1) = 110$, $c(a_2) = 1110$, ... , $c(a_{n-1}) = \underbrace{1 \dots 1}_{n\text{-mal}}0$)

d) $c(a_i) = 0^i 10^{n-i-1}$ ($c(a_0) = 10 \dots 0$, $c(a_1) = 010 \dots 0$, $c(a_2) = 0010 \dots 0$, ... , $c(a_{n-1}) = 0 \dots 01$)