

Prüfung
Einführung in das Programmieren (Java)
188.062
Do, 14. Jänner 1999

Die Lösungen sind fett gedruckt.

Insgesamt 100 Punkte

1. Kontrollstrukturen (30 Punkte)

1a) (10 Punkte) Welches Ergebnis x liefert das folgende Programmstück, wenn a vorher die angegebenen Werte hat? (a ist vom Typ int)

<code>int x = 10;</code>	a	x
<code>switch ((a%10)) {</code>	23	8
<code> case 3:</code>	92	2
<code> case 4: x += 3;</code>	45	5
<code> case 5: x -= 5; break;</code>	104	8
<code> case 7: x += 7; break;</code>	888	10
<code> case 8:</code>	9	10
<code> case 9: break;</code>	200	2
<code> case 1:</code>	31	2
<code> default: x /= 5;</code>	16	2
<code>}</code>	37	17

1b) (10 Punkte) Welchen Wert haben x und y nach der Ausführung des folgenden Programmstückes? Das Array a wird mit den Ziffern Ihrer Matrikelnummer initialisiert.

```
int a[] = { , , , , , , };
int x = 0;
int y = 0;

for (int i=1; i < a.length; i++) {
  if (a[i] > a[x]) x = i;
  if (a[i] < a[y]) y = i;
}
```

Wert von x: **Der Index der größten Zahl im Array, also bei allen angetretenen Kollegen 0, weil die Matrikelnummer mit 9 beginnt.**

Wert von y: **Der Index der kleinsten Zahl um Array. Falls sie mehrfach vorkommt: Das erste Vorkommnis.**

1c) (10 Punkte) Schreiben sie eine Methode, die als Argument ein int-Array erhält und als Returnvalue die Summe der Einträge des Arrays, deren Index gerade ist, minus der Summe der Einträge, deren Index ungerade ist, liefert. D.h. das 0., 2., etc. Element werden addiert und davon die Summe des 1., 3., etc. Elements subtrahiert. Die Methode muß für beliebig lange Arrays funktionieren.

Es gibt 2 gleichwertige Lösungen:

```
int summendifferenz(int array[]) {
    int summe = 0;
    for (int i = 0; i < array.length; i++)
        if ((i % 2) == 0)
            summe += array[i];
        else
            summe -= array[i];
    return summe;
}
```

```
int summendifferenz(int array[]) {
    int gerade = 0,
        ungerade = 0;
    for (int i = 0; i < array.length; i += 2)
        gerade += array[i];
    for (int i = 1; i < array.length; i += 2)
        ungerade += array[i];
    return gerade - ungerade;
}
```

2. Klassen und Objekte - BeBe-Sprache (30 Punkte)

Kinder lernen gerne und auch relativ schnell die BeBe-Sprache, sodaß die Erwachsenen sie nicht so einfach verstehen können. Die Wörter der BeBe-Sprache entstehen aus den normalen deutschen Wörtern dadurch, daß an jeden Vokal (Selbstlaut a, e, i, o, u, ä, ö, ü) ein b angehängt wird und danach der Vokal nochmals gesprochen wird.

Beispiele:

Deutsch	BeBe-Sprache
Kinder	Kibindeber
Vokal	Vobokabal
neue	Nebeubuebe
Wörter	Wöbörteber

Anmerkung: Hier wird eine vereinfachte Form der BeBe-Sprache verwendet, bei der Zwielaute als 2 getrennte Vokale betrachtet werden. Kinder sprechen natürlich Zwielaute (ei, eu, äü, ie) wie einen einzigen Vokal aus.

Die folgende Klasse BeBe definiert Objekte der BeBe-Sprache mit ihren wichtigsten Methoden zum Übersetzen von und ins Deutsche.

```

class BeBe {
    private String s; // interne Darstellung als String in
                      // BeBe-Sprache

    public BeBe (String deutsch){ // Konstruktor. Argument
                                   // ist Wort in deutscher Sprache
        ...
    }

    public String toString() {
                                   // für Ausgabe eines BeBe-Textes
        return s;
    }

    public static boolean isBeBe(String x) {
        // Klassenmethode. Stellt fest, ob x ein String
        // der BeBe-Sprache ist
        ...
    }
    ...
}

```

Aufgaben:

2a) Schreiben Sie den Konstruktor mit dem Kopf wie oben angegeben. (10 Punkte)

```

private final static String vokale = "aeiouäöüAEIOUÄÖÜ";
public BeBe(String deutsch) {
    s = "";
    for (int i=0; i<deutsch.length(); i++) {
        char c = deutsch.charAt(i);
        if (-1 < vokale.indexOf(c))
            s = s + c + 'b';
        s = s + c;
    }
}

```

2b) Schreiben Sie die Methode isBeBe deren Kopf oben angegeben ist. (10 Punkte)

```

public static boolean isBeBe(String x) {
    for (int i=0; i<x.length(); i++) {
        char c = x.charAt(i);
        if (-1 < vokale.indexOf(c))
            if ('b' == x.charAt(i+1)
                && c == x.charAt(i-2))
                i += 2;
            else
                return false;
        }
    return true;
}

```

2c) Geben Sie für die noch fehlende Methode zur Umwandlung eines bestehenden BeBe-Objektes ins Deutsche den Methodenkopf samt Kommentar an. (5 Punkte)

```
public String deutsch(){ // Gibt den Inhalt eines
    ... } // BeBe-Objektes auf deutsch zurück
```

2d) Schreiben Sie eine Klassenmethode, die auf der Konsole ein als Argument übergebenes deutsches Wort sowohl in deutsch als auch in BeBe-Sprache ausgibt. (5 Punkte)

```
public static void printboth (String deutsch) {
    System.out.println(deutsch + " " +
        new BeBe(deutsch).toString());
}
```

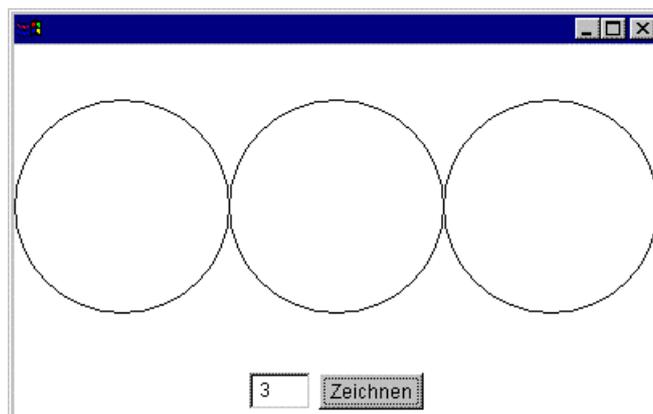
3. AWT (40 Punkte)

Schreiben Sie ein Programm, das in einem Window im oberen Teil eine Anzahl von gleich großen Kreisen und darunter nebeneinander ein Textfield, in dem man die Anzahl der Kreise eingeben kann und einen Button, mit dem man das Neuzeichnen der entsprechenden Anzahl von Kreisen veranlassen kann, darstellt.

Der Canvas, in auf dem die Kreise gezeichnet werden, soll 200 Pixel hoch und 400 Pixel breit sein. Das Textfield und der Button sollen darunter nebeneinander gemeinsam zentriert angezeigt werden.

Der Default für die Anzahl der Kreise ist 3.

Das Programm wird durch Anklicken des Kreuzes in der rechten oberen Ecke des Fensterrahmens beendet.



```

import java.awt.*;
import java.awt.event.*;

public class Kreise extends Frame implements ActionListener {
    MyCanvas canvas;
    TextField textfield;

    public Kreise () {
        setLayout(new BorderLayout());
        canvas = new MyCanvas(3);
        textfield = new TextField(2);
        Button button = new Button("Zeichnen");
        Panel panel = new Panel();
        panel.setLayout(new FlowLayout(FlowLayout.CENTER));
        panel.add(textfield);
        panel.add(button);
        add("North", canvas);
        add("South", panel);
        button.addActionListener(this);
        addWindowListener(new MyWindowListener());
        pack();
        show();
    }

    public void actionPerformed(ActionEvent e) {
        try {
            canvas.setNumber(Integer.parseInt(textfield.getText()));
        }
        catch (Exception ee) {
            System.err.println(ee);
        }
    }

    public static void main(String args[]) {
        new Pruefung0199();
    }
}

class MyCanvas extends Canvas {
    int anzahl;
    MyCanvas (int a) {
        anzahl = a;
        setSize(400, 200);
    }

    public void paint(Graphics g) {
        int d = Math.min(200, 400/anzahl);
        for (int i=0; i< anzahl; i++)
            g.drawOval(i*d, 100-d/2, d, d);
    }

    void setNumber(int a) {
        anzahl = a;
        repaint();
    }
}

class MyWindowListener extends WindowAdapter {
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
}

```

Das Setzen des Layouts ist in diesem Fall nicht notwendig, weil sowohl zentriertes FlowLayout der Default für Panels ist, als auch BorderLayout der Default für Frames.