

Prüfung
Einführung in das Programmieren (Java)
188.062
Di, 8. Juni 1999

Nachname, Vorname:

Matrikelnummer:

Kennnummern:

Insgesamt gibt es 100 Punkte.

1. Datenstrukturen (30 Punkte)

Gegeben sind die folgenden beiden Klassendefinitionen.

```
class Ziffer {
    int wert;
    Ziffer next;

    Ziffer(int w, Ziffer n) {
        wert = w;
        next = n;
    }
}

class Beispiell {
    static Ziffer build (int m[]) {
        Ziffer z = null;
        for (int i=m.length-1; i>=0; i--)
            z = new Ziffer(m[i], z);
        return z;
    }

    static void fragel (Ziffer z) {
        if (z != null) {
            fragel(z.next);
            System.out.print(z.wert);
        }
    }

    static int frage2 (Ziffer z) {
        if (z.next == null)
            return z.wert;
        else {
            int i = frage2(z.next);
            if (i < z.wert)
                return i;
            else
                return z.wert;
        }
    }
}
```

```

static int frage3(int a[]) {
    int x = a[0];
    int i = a.length;
    while (0 < i--)
        if (a[i] % 4 == 0)
            x = a[i];
    return x;
}

```

Das Array matrikelnummer wird mit den Ziffern Ihrer Matrikelnummer initialisiert. Welche Ausgabe liefern die folgenden Zeilen:

```

int matrikelnummer[] = { , , , , , , };
Ziffer z = Beispiell.build(matrikelnummer);
Beispiell.fragel(z); System.out.println();

```

(10 Punkte)

```

System.out.println(Beispiell.frage2(z));

```

(10 Punkte)

```

System.out.println(Beispiell.frage3(matrikelnummer));

```

(10 Punkte)

2. Strings (20 Punkte)

Schreiben Sie eine Klassenmethode, die einen String als Argument erhält und als Return Value einen String liefert, der dem Argument entspricht, nur daß

| | | |
|--------------------|-------|----------------------------|
| \t | durch | TAB |
| \n | durch | NEWLINE |
| \\ | durch | \ |
| \{beliebiger Text} | durch | COMMANDO "beliebiger Text" |

ersetzt wurde. Backslashes (\), die von anderen Zeichen gefolgt werden, werden als normale Zeichen behandelt, d.h. unverändert ausgegeben. Also zum Beispiel

Argument:

```
Eins\tzwei\ndrei\\vier\ ein\halb\{fünf}sechs
```

Return Value:

```
EinsTABzweiNEWLINEdrei\vier\ ein\halbCOMMANDO "fünf"sechs
```

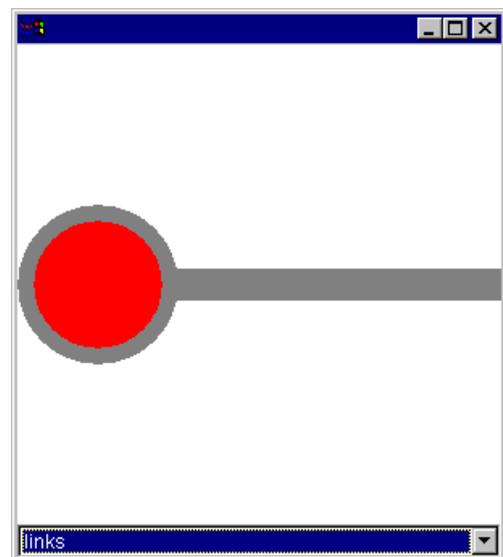
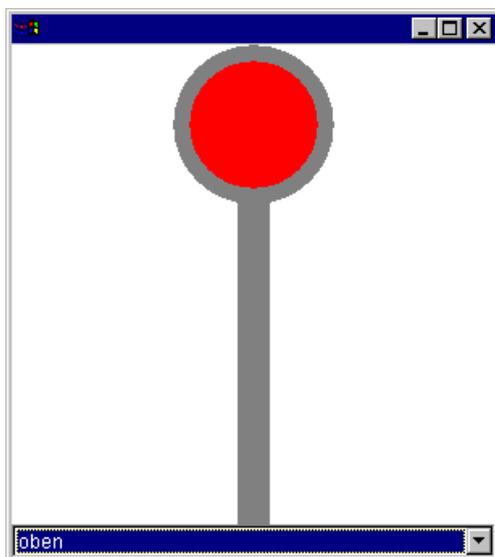
3. AWT (50 Punkte)

Schreiben sie eine Java-Application, die in einem Fenster eine Signalkelle darstellt, und darunter eine Choice-Box, in der die Position der Kelle geändert werden kann.

Die Werte in der Choice-Box sind: oben, unten, links, rechts.

Die Kelle wird in einem Quadrat von 300 mal 300 Pixel gezeichnet. Die Breite des grauen Stiels ist 20, der Außendurchmesser des grauen Randes der roten Scheibe beträgt 100, der Durchmesser der roten Scheibe selbst 80. Die Länge der Kelle ist 300.

Das Programm wird durch Anklicken des Schließenbuttons im rechten oberen Ecke des Windows beendet.



Lösung

1. Datenstrukturen

Die Methode `build` liest das Array mit der Matrikelnummer von hinten und fügt die einzelnen Ziffern in dieser Reihenfolge an den Anfang der Liste ein. Daher stehen die Ziffern in der Liste wieder in der selben Reihenfolge wie im Array.

`frage1` gibt zuerst den Rest der Liste durch einen rekursiven Aufruf aus und dann das aktuelle Element. Die Liste wird als von hinten nach vorne ausgegeben, d.h. die Ziffern der Matrikelnummer verkehrt herum ausgegeben.

`frage2` gibt das Minimum der übergebenen Liste als Returnvalue zurück. Auch hier wird wieder zuerst der Rest der Liste bearbeitet und dann das aktuelle Element mit dem Minimum der restlichen Liste verglichen.

`frage3` geht das übergebene Array von hinten nach vorne durch und kopiert jede durch 4 teilbare Zahl in `x`. Der Returnvalue ist daher diejenige durch 4 teilbare Ziffer, die den kleinsten Index hat, d.h. am weitesten links steht. Sollte keine Ziffer durch 4 teilbar sein, wird das die erste Ziffer (mit Index 0) zurückgegeben.

2. Strings

```
static String replace (String in) {
    int i;
    String out = "";
    for (i=0; i<in.length() - 1; i++) {
        if (in.charAt(i) == '\\') {
            char c = in.charAt(++i);
            if (c == 't')
                out += "TAB";
            else if (c == 'n')
                out += "NEWLINE";
            else if (c == '\\')
                out += "\\";
            else if (c == '{') {
                int e = in.indexOf('}', i);
                if (e == -1)
                    out += '{';           (siehe Anmerkung unten)
                else {
                    out += "COMMANDO \"" + in.substring(i+1, e) + "\"";
                    i = e;
                }
            }
            else
                out += c;
        }
    }
    if (i < in.length())
        out += in.charAt(i);
    return out;
}
```

Anmerkung: Es ist nicht angegeben, was in dem Fall passieren soll, in dem der öffnenden Klammer keine schließende folgt. In dieser Lösung wird einfach die öffnende Klammer ausgegeben und keine Fehlermeldung ausgegeben. Die Ausgabe einer Fehlermeldung wäre auch möglich, war aber nicht verlangt.

3. AWT

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

public class Pruefung0699 extends Frame implements ItemListener {
    Choice choice;
    KelleCanvas canvas;

    public Pruefung0699 () {
        choice = new Choice();
        for (int i=0; i < KelleCanvas.richtungsbezeichnung.length; i++)
            choice.add(KelleCanvas.richtungsbezeichnung[i]);
        choice.addItemListener(this);
        canvas = new KelleCanvas();
        addWindowListener(new Closer());
        add("North", canvas);
        add("South", choice);
        pack();
        setVisible(true);
    }

    public void itemStateChanged(ItemEvent e) {
        canvas.setRichtung(choice.getSelectedIndex());
    }

    public static void main(String args[]) {
        new Pruefung0699();
    }

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

class KelleCanvas extends Canvas {
    static String richtungsbezeichnung[] =
        {"oben", "unten", "rechts", "links"};
    private static final int mitteX[] = {150, 150, 250, 50};
    private static final int mitteY[] = {50, 250, 150, 150};
    private static final int anfangX[] = {150, 150, 0, 50};
    private static final int anfangY[] = {50, 0, 150, 150};

    private int richtung = 0;
    KelleCanvas() {
        setSize (300, 300);
    }

    void setRichtung(int r) {
        richtung = r;
        repaint();
    }

    public void paint (Graphics g) {
        g.setColor(Color.gray);
        if (richtung < 2)
            g.fillRect(anfangX[richtung] - 10, anfangY[richtung], 20, 290);
        else
            g.fillRect(anfangX[richtung], anfangY[richtung] - 10, 290, 20);
        g.fillOval(mitteX[richtung] - 50, mitteY[richtung] - 50, 100, 100);
        g.setColor(Color.red);
        g.fillOval(mitteX[richtung] - 40, mitteY[richtung] - 40, 80, 80);
    }
}
```