JAVA - EINFÜHRUNG

Musterlösungen

Hubert Partl

Inhaltsverzeichnis

Java Einführung

von Hubert Partl, ZID BOKU Wien

Musterlösungen zu den Übungsaufgaben

Wenn Sie Java erfolgreich lernen wollen, empfehle ich Ihnen dringend, die Musterlösungen erst dann auszudrucken und anzusehen, wenn Sie bereits *alle* Übungsbeispiele selbständig fertig programmiert haben, also erst am Ende des Kurses.

Es gibt immer mehrere Möglichkeiten, eine Aufgabe zu programmieren. Die hier gezeigten Musterlösungen sind also nicht notwendiger Weise "besser" als das, was Sie selbst geschrieben haben. Außerdem hat sich gezeigt, dass manche von diesen Beispielen in bestimmten Java-Implementierungen *nicht* fehlerfrei funktionieren.

1 HelloWorld-Applikation

1.1 HelloWorld.java

```
public class HelloWorld {
  public static void main (String[] args) {

    System.out.println("Hello World!");
  }
}
```

1.2 HelloText.java

```
public class HelloText {

  public String messageText = "Hello World!";
  // or: private ...

  public void printText() {
    System.out.println (messageText);
  }

  public static void main (String[] args) {
    HelloText h = new HelloText();
    h.printText();
  }
}
```

1.3 MultiText.java

```
public class MultiText {
  public static void main (String[] args) {

    HelloText engl = new HelloText();
    HelloText germ = new HelloText();
    germ.messageText = "Hallo, liebe Leute!";
    HelloText cat = new HelloText();
    cat.messageText = "Miau!";

    engl.printText();
    germ.printText();
    engl.printText();
    cat.printText();
    engl.printText();
  }
}
```

2 Programmdokumentation (javadoc)

2.1 HelloDoc.java

```
/**
* ein einfaches Hello-World-Programm.
* <p>
* Im Gegensatz zum kurzen, rein statischen "HelloWorld"
* ist dieses Programm ein Musterbeispiel
* für eine <b>objekt-orientierte</b> Java-Applikation.
*
* @author Hubert Partl
* @version 99.9
* @since JDK 1.0
* @see HelloWorld
*/
public class HelloDoc {

  /** der Text, der gedruckt werden soll.
  */
  public String messageText = "Hello World!";

  /** druckt den Text messageText auf System.out aus.
  *   @see #messageText
  */
  public void printText() {
    System.out.println (messageText);
  }

  /** Test des HelloDoc-Objekts.
  */
  public static void main (String[] args) {
    HelloDoc h = new HelloDoc();
    h.printText();
  }
}
```

3 Quadratzahlen

3.1 Quadrat.java

```
public class Quadrat {
  public static void main (String[] args) {
    int zahl, quad;
    System.out.println(" Quadratzahlen:");
    for ( zahl=1; zahl<=20; zahl++ ) {
      quad = zahl * zahl;
      System.out.println (zahl + " * " + zahl +
        " = " + quad );
    }
  }
}
```

4 Steuer

4.1 Steuer.java

```
public class Steuer {
  public static void main (String[] args) {
    double brutto = 100; /* inkl. USt. */
    double prozentsatz = 20;
    System.out.println (brutto + " brutto");
    double steuer = brutto * prozentsatz / (100.0 + prozentsatz);
    System.out.println (steuer + " USt.");
    double netto = brutto - steuer;
    System.out.println (netto + " netto");
  }
}
```

5 Sparbuch

5.1 Bank.java

```
public class Bank{
  public static void main (String[] args) {

    double invest = 10000.0;
    double rate = 4.0;
    double factor = (100.0 + rate) / 100.0;
    double amount;
    int numYears = 10;

    System.out.println("Investment =      " + invest);
    System.out.println("Interest Rate =   " + rate);

    amount=invest;
    for (int year=1; year<=numYears; year++) {
      amount = amount * factor;
      System.out.println ("Year " + year +
        "   Amount = " + amount );
    }
  }
}
```

6 erweitertes Sparbuch

6.1 BankEx.java

```
import java.text.*;

public class BankEx {
  public static void main (String[] args) {
    double invest, rate, factor, amount;
    int numYears;

    DecimalFormat df = new DecimalFormat("#,###,##0.00");
    DecimalFormat intf = new DecimalFormat("00");

    if (args.length != 2) {
      System.out.println ("Usage: java BankEx amount rate");
      System.exit(1);
    }

    try {
      invest = Double.valueOf(args[0]).doubleValue();
      rate = Double.valueOf(args[1]).doubleValue();
      factor = (100.0 + rate) / 100.0;
      numYears = 10;

      System.out.println("Investment =      " +
        df.format(invest) );
      System.out.println("Interest Rate =   " +
        df.format(rate) );

      amount=invest;
      for (int year=1; year<=numYears; year++) {
        amount = amount * factor;
      System.out.println ("Year " + intf.format(year) +
          "   Amount = " + df.format(amount) );
      }
    } catch (Exception e) {
        System.out.println("*** error: " + e);
    }
  }
}
```

6.2 BankEx2.java

```
import java.text.*;

public class BankEx2 {
  public static void main (String[] args) {
    double invest, rate, factor, amount;
    int numYears;
    DecimalFormat df = new DecimalFormat("#,###,##0.00");
    DecimalFormat intf = new DecimalFormat("00");

    try {
      invest = Double.valueOf(args[0]).doubleValue();
    } catch (Exception e) {
      // ArrayIndexOutOfBoundsException or NumberFormatException
      System.out.println("Investment not specified * " + e);
      invest = 1000;
    }
```

```
    try {
      rate = Double.valueOf(args[1]).doubleValue();
    } catch (Exception e) {
      // ArrayIndexOutOfBoundsException or NumberFormatException
      System.out.println("Rate not specified * " + e);
      rate = 3.5;
    }

    factor = (100.0 + rate) / 100.0;
    numYears = 10;
    System.out.println("Investment =      " +
      df.format(invest) );
    System.out.println("Interest Rate =   " +
      df.format(rate) );

    amount=invest;
    for (int year=1; year<=numYears; year++) {
      amount = amount * factor;
      System.out.println ("Year " + intf.format(year) +
        "   Amount = " + df.format(amount) );
    }
  }
}
```

7 Kurs

7.1 Kurs.java

```
public class Kurs {
  private String kursTitel = null;
  private boolean kostenlos = false;
  private int anzahl = 0;
  private String[] teilnehmer;

  public Kurs (String kursTitel, int maxAnzahl) {
    setKursTitel(kursTitel);
    teilnehmer = new String [maxAnzahl];
  }

  private void setKursTitel (String kursTitel) {
    this.kursTitel = kursTitel;
  }
  public String getKursTitel() {
    return kursTitel;
  }
  public void setKostenlos (boolean kostenlos) {
    this.kostenlos = kostenlos;
  }
  public boolean isKostenlos() {
    return kostenlos;
  }
  // no setTeilnehmer(String[])
  public String[] getTeilnehmer() {
    return teilnehmer;
  }
  private void setTeilnehmer (int i, String name)
      throws ArrayIndexOutOfBoundsException {
    this.teilnehmer[i] = name;
  }
  public String getTeilnehmer (int i)
      throws ArrayIndexOutOfBoundsException {
    return teilnehmer[i];
  }
  public void addTeilnehmer (String name)
      throws ArrayIndexOutOfBoundsException {
    if (anzahl >= teilnehmer.length )
      throw new ArrayIndexOutOfBoundsException();
    else {
      anzahl++;
      setTeilnehmer (anzahl-1, name);
    }
  }
  // no setAnzahl(int)
  public int getAnzahl() {
    return anzahl;
  }
  public int getMaxAnzahl() {
    return teilnehmer.length;
  }
  public boolean equals (Object other) {
   return ( other instanceof Kurs &&
     ((Kurs)other).getKursTitel().equals (this.kursTitel) );
  }
  public String toString() {
```

```
    String s = getKursTitel();
    if ( isKostenlos() )
      s = s + " (kostenlos)";
    s = s + ": " + getAnzahl() + " Teilnehmer, "
      + (getMaxAnzahl()-getAnzahl()) + " Plätze frei";
    return s;
  }

  public static void main (String[] args) {
    Kurs java1 = new Kurs ("Java Einfuehrung", 15);
    java1.setKostenlos(true);
    Kurs java2 = new Kurs ("Java fuer Fortgeschrittene", 8);
    java1.addTeilnehmer("Hubert");
    java2.addTeilnehmer("Hubert");
    java1.addTeilnehmer("Clemens");
    java1.addTeilnehmer("Ernst");
    java2.addTeilnehmer("Markus");
    System.out.println (java1);
    for (int i=0; i < java1.getAnzahl(); i++)
      System.out.println ("  " + java1.getTeilnehmer(i));
    System.out.println (java2);
    for (int i=0; i < java2.getAnzahl(); i++)
      System.out.println ("  " + java2.getTeilnehmer(i));
  }
}
```

8 Person und Student

8.1 Person.java

```
public class Person {

   private String vorname;
   private String zuname;

   public Person (String zuname, String vorname) {
      this.setZuname(zuname);
      this.setVorname(vorname);
   }

   protected void setZuname (String s) {
      this.zuname = s;
   }
   public String getZuname() {
      return this.zuname;
   }

   protected void setVorname (String s) {
      this.vorname = s;
   }
   public String getVorname() {
      return this.vorname;
   }

   public String toString() {
      return this.vorname + " " + this.zuname;
   }

}
```

8.2 Student.java

```
public class Student extends Person {

   private String uni;

   public Student (String zuname, String vorname,
         String uni) {
      super (zuname, vorname);
      this.uni = uni;
    }

   protected void setUni (String uni) {
      this.uni = uni;
   }
   public String getUni() {
      return this.uni;
   }

  public String toString() {
      String s;
      s = super.toString() + ", studiert an " + uni;
      return s;
   }
}
```

8.3 PersStud.java

```
public class PersStud {
  public static void main (String[] args) {
    Person georg, willi, anna, barbara;

    anna = new Person ("Schmidt", "Anna");
    barbara = new Student ("Paulus", "Barbara", "TU Wien");
    georg = new Student ("Fischer", "Georg", "BOKU");
    willi = new Person ("Schlosser", "Wilhelm");

    System.out.println();
    System.out.println("Vornamen:");
    System.out.println( anna.getVorname() );
    System.out.println( barbara.getVorname() );
    System.out.println( georg.getVorname() );
    System.out.println( willi.getVorname() );
    System.out.println();
    System.out.println("Komplette Informationen:");
    System.out.println( anna );
    System.out.println( barbara );
    System.out.println( georg );
    System.out.println( willi );
    System.out.println();
    }
}
```

9 Konto

9.1 KontoNichtGedecktException.java

```
public class KontoNichtGedecktException
   extends IllegalArgumentException {}
```

9.2 Konto.java

```
public class Konto {

   private Person inhaber;
   private double guthaben = 0.0;

   public Konto (Person inhaber) {
      this.setInhaber(inhaber);
   }

   private void setInhaber (Person inhaber) {
      this.inhaber = inhaber;
   }
   public Person getInhaber() {
      return this.inhaber;
   }

   private void setGuthaben (double neuerBetrag) {
        this.guthaben = neuerBetrag;
   }
   public double getGuthaben() {
      return this.guthaben;
   }

   public void einzahlen (double betrag) {
      this.setGuthaben( this.guthaben + betrag );
   }

   public void abheben (double betrag) {
      double neuerBetrag = this.guthaben - betrag;
      if (neuerBetrag >= 0.0) {
        this.setGuthaben( neuerBetrag );
      }
   }

   public String toString() {
      return this.inhaber + ": " + this.guthaben + " Euro";
   }

   public static void main (String[] args) {

     Person hubert = new Person ("Partl", "Hubert");
     Konto kontoH = new Konto (hubert);
     System.out.println(kontoH);
     kontoH.einzahlen(1000.0);
     System.out.println(kontoH);
     kontoH.abheben(500.0);
     System.out.println(kontoH);
     kontoH.abheben(9999.0);
     System.out.println(kontoH);

     Student anna = new Student ("Fleißig", "Anna", "Uni Wien");
```

```
      Konto kontoA = new Konto (anna);
      System.out.println(kontoA);
      kontoA.einzahlen(1000.0);
      System.out.println(kontoA);
      kontoA.abheben(500.0);
      System.out.println(kontoA);

   }
}
```

10 Katzenmusik

10.1 Katzenmusik.java

```
public class Katzenmusik {
  public static void main (String[] args) {

    LautesTier tier1 = new Katze();
    LautesTier tier2 = new Hund();
    for (int i=1; i<=10; i++) {
      tier1.gibtLaut();
      tier2.gibtLaut();
    }
  }
}
```

10.2 LautesTier.java

```
public interface LautesTier {
  public void gibtLaut() ;
}
```

10.3 Katze.java

```
public class Katze implements LautesTier {
  public void gibtLaut() {
    System.out.println("Miau!");
  }
}
```

10.4 Hund.java

```
public class Hund implements LautesTier {
  public void gibtLaut() {
    System.out.println("Wau wau!");
  }
}
```

11 Layout-Manager

11.1 LayoutTests.java

```
import java.awt.* ;

public class LayoutTests extends Frame {

  private Button b1, b2;
  private Label lab;
  private TextField tf;

  // activate (un-comment) one of the following lines:
  private FlowLayout myLayout = new FlowLayout();
  // private FlowLayout myLayout = new FlowLayout( FlowLayout.LEFT );
  // private GridLayout myLayout = new GridLayout(2,2);
  // private GridLayout myLayout = new GridLayout(4,1);
  // private BorderLayout myLayout = new BorderLayout();

  public void init() {
    setLayout ( myLayout );
    b1 = new Button("ok");
    add(b1,"West");
    b2 = new Button("cancel");
    add(b2,"South");
    lab = new Label("Hallo!");
    add(lab,"Center");
    tf = new TextField("",20);
    add(tf,"East");
    setSize(300,200); // or: pack();
    setVisible(true);
  }

  public static void main (String[] args) {
    LayoutTests f = new LayoutTests();
    f.init();
  }

}
```

12 Canvas Verkehrsampel

12.1 TrafficCanvas.java

```
import java.awt.*;

public class TrafficCanvas extends Canvas {

  public Dimension getMinimumSize() {
    return new Dimension(100,260);
  }
  public Dimension getPreferredSize() {
    return getMinimumSize();
  }

  public void paint (Graphics g) {
    g.setColor (Color.black);
    g.fillRect (10, 10, 80, 240);
    g.setColor (Color.red);
    g.fillOval (20, 20, 60, 60);
    g.setColor (Color.yellow);
    g.fillOval (20, 100, 60, 60);
    g.setColor (Color.green);
    g.fillOval (20, 180, 60, 60);
  }

  public static void main (String[] args) {
    Frame f = new Frame("Traffic Light Test");
    TrafficCanvas theLight = new TrafficCanvas();
    f.setLayout ( new FlowLayout() );
    f.add(theLight);
    f.setSize(200,300); // or: pack();
    f.setVisible(true);
    // no event handling, use Ctrl-C to stop this program.
  }
}
```

13 einfache GUI-Applikation

13.1 Gui.java

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

public class Gui extends Frame
  implements ActionListener, WindowListener {

  Label titleLabel;
  Button openButton, closeButton;
  Panel northPanel, southPanel;

  public Gui (String s) {
    super(s);
  }

  public void init () {

    setLayout (new BorderLayout() );

    // north panel = title label

    northPanel = new Panel();
    northPanel.setLayout (new FlowLayout() );
    titleLabel = new Label ("GUI Exercise ");
    northPanel.add(titleLabel);
    add (northPanel, "North");

    // south panel = two buttons

    openButton = new Button(" Open ");

    closeButton = new Button(" Close ");
    closeButton.addActionListener (this);
    closeButton.setActionCommand ("close");

    southPanel = new Panel();
    southPanel.setLayout (new FlowLayout() );
    southPanel.add(openButton);
    southPanel.add(closeButton);
    add (southPanel, "South");

    addWindowListener(this);
    pack(); // setSize(500,500);
    setVisible(true);
  }

  public static void main (String[] args) {
    Gui f = new Gui("Test");
    f.init();
  }

  public void actionPerformed (ActionEvent e) {
    String which = e.getActionCommand();
    if ( which.equals("close") ) {
      dispose();
      System.exit(0);
    }
```

```
  }
  public void windowClosing (WindowEvent e) {
    dispose();
    System.exit(0);
  }
  public void windowClosed (WindowEvent e) { }
  public void windowOpened (WindowEvent e) { }
  public void windowIconified (WindowEvent e) { }
  public void windowDeiconified (WindowEvent e) { }
  public void windowActivated (WindowEvent e) { }
  public void windowDeactivated (WindowEvent e) { }
}
```

14 HelloWorld Applet

14.1 HelloApp.java

```
import java.awt.*;
import java.applet.*;

public class HelloApp extends Applet {

  private String s = "Hello World!";

  public void paint (Graphics g) {
    g.drawString (s, 25, 25);
  }
}
```

14.2 hello.html

```
<html>
<head>
<title>Example</title>
</head>
<body>

<h1>Example</h1>

<p align=center>
<applet code="HelloApp.class" width=200 height=100
        alt="Hello World!">
  Hello World!
</applet>
</p>

</body>
</html>
```

15 Applet Thermostat (in einer Klasse)

15.1 Thermo.java

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

public class Thermo extends Applet
  implements ActionListener {

  private int temp = 22;
  private Label tempDisplay;
  private Button plusButton, minusButton;
  private Panel centerPanel, southPanel;

  public void init () {

    setLayout (new BorderLayout() );

    // center panel = temperature display

    tempDisplay = new Label("00", Label.CENTER);
    Font bigFont = new Font ("Helvetica", Font.BOLD, 20);
    tempDisplay.setFont(bigFont);
    showTemp();

    centerPanel = new Panel();
    centerPanel.setLayout (new FlowLayout() );
    centerPanel.add(tempDisplay);
    add (centerPanel, "Center");

     // south panel = two buttons

    minusButton = new Button(" -1");
    minusButton.addActionListener (this);
    minusButton.setActionCommand ("minus");

    plusButton = new Button("+1");
    plusButton.addActionListener (this);
    plusButton.setActionCommand ("plus");

    southPanel = new Panel();
    southPanel.setLayout (new FlowLayout() );
    southPanel.add(minusButton);
    southPanel.add(plusButton);
    add (southPanel, "South");

    setVisible(true);
  }

  private void showTemp() {
    tempDisplay.setBackground(Color.white);
    if (temp<20) {
      tempDisplay.setForeground(Color.blue);
    }
    else if (temp>25) {
      tempDisplay.setForeground(Color.red);
    }
    else {
```

```
      tempDisplay.setForeground(Color.magenta);
    }
    tempDisplay.setText("" + temp);
  }

  public void actionPerformed (ActionEvent e) {
    String which = e.getActionCommand();
    if ( which.equals("minus") ) {
      temp--;
    }
    else if ( which.equals("plus") ) {
      temp++;
    }
    showTemp();
  }

}
```

15.2 Thermo.html

```
<html>
<applet code="Thermo.class" width=200 height=100>
  <p>Without Java, you cannot play with the temperature, sorry.
</applet>
</html>
```

16 Applet Thermostat (in drei Klassen)

16.1 ThermoModel.java

```
public class ThermoModel {

  private int temp = 22;

  public void setTemp (int temp) {
    this.temp = temp;
  }

  public void addTemp (int diff) {
    temp = temp + diff;
  }

  public int getTemp() {
    return temp;
  }
}
```

16.2 ThermoView.java

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

public class ThermoView extends Applet {

  private ThermoModel model;
  private ThermoController controller;
  private Label tempDisplay;
  private Button plusButton, minusButton;
  private Panel centerPanel, southPanel;

  public void init () {

    model = new ThermoModel();
    model.setTemp(22);
    controller = new ThermoController (this, model);

    setLayout (new BorderLayout() );

    // center panel = temperature display

    tempDisplay = new Label("00", Label.CENTER);
    Font bigFont = new Font ("Helvetica", Font.BOLD, 20);
    tempDisplay.setFont(bigFont);
    showTemp();

    centerPanel = new Panel();
    centerPanel.setLayout (new FlowLayout() );
    centerPanel.add(tempDisplay);
    add (centerPanel, "Center");

     // south panel = two buttons

    minusButton = new Button(" -1");
    minusButton.addActionListener ( controller );
    minusButton.setActionCommand ("minus");
```

```
    plusButton = new Button("+1");
    plusButton.addActionListener ( controller );
    plusButton.setActionCommand ("plus");

    southPanel = new Panel();
    southPanel.setLayout (new FlowLayout() );
    southPanel.add(minusButton);
    southPanel.add(plusButton);
    add (southPanel, "South");

    setVisible(true);
  }

  protected void showTemp() {
    int temp = model.getTemp();
    tempDisplay.setBackground(Color.white);
    if (temp<20) {
      tempDisplay.setForeground(Color.blue);
    }
    else if (temp>25) {
      tempDisplay.setForeground(Color.red);
    }
    else {
      tempDisplay.setForeground(Color.magenta);
    }
    tempDisplay.setText("" + temp);
  }
}
```

16.3 ThermoController.java

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

public class ThermoController
  implements ActionListener {

  private ThermoView view;
  private ThermoModel model;

  public ThermoController (ThermoView v, ThermoModel m) {
    this.view = v;
    this.model = m;
  }

  public void actionPerformed (ActionEvent e) {
    String which = e.getActionCommand();
    if ( which.equals("minus") ) {
      model.addTemp(-1);
    }
    else if ( which.equals("plus") ) {
      model.addTemp(+1);
    }
    view.showTemp();
  }
}
```

16.4 ThermoView.html

```
<html>
<applet code="ThermoView.class" width=200 height=100>
  <p>Without Java, you cannot play with the temperature, sorry.
</applet>
</html>
```

17 Applet Verkehrsampel

17.1 TrafficLight.java

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

public class TrafficLight extends Applet
  implements ActionListener {

  private Label titleLabel;
  private TheLight theLight; // defined below
  private Button nextButton, stopButton;
  private Panel northPanel, centerPanel, southPanel;

  public void init () {

    theLight = new TheLight();

    setLayout (new BorderLayout() );

    // north panel = title label

    northPanel = new Panel();
    northPanel.setLayout (new FlowLayout() );
    titleLabel = new Label ("Traffic Light Exercise ");
    northPanel.add(titleLabel);
    add (northPanel, "North");

    // center panel = traffic light drawing canvas

    centerPanel = new Panel();
    centerPanel.setLayout (new FlowLayout() );
    centerPanel.add(theLight);
    add (centerPanel, "Center");

    // south panel = two buttons

    nextButton = new Button("Next");
    nextButton.addActionListener (this);
    nextButton.setActionCommand ("next");

    stopButton = new Button("Stop");
    stopButton.addActionListener (this);
    stopButton.setActionCommand ("stop");

    southPanel = new Panel();
    southPanel.setLayout (new FlowLayout() );
    southPanel.add(nextButton);
    southPanel.add(stopButton);
    add (southPanel, "South");

    setVisible(true);
  }

  public void actionPerformed (ActionEvent e) {
    String which = (String) e.getActionCommand();
    if ( which.equals("next") ) {
      theLight.nextState();
```

```
      theLight.repaint();
    }
    if ( which.equals("stop") ) {
      theLight.setState(TheLight.RED);
      theLight.repaint();
    }
  }
}
```

17.2 TheLight.java

```
import java.awt.*;

public class TheLight extends Canvas {

  public static final int RED = 1;
  public static final int REDYELLOW = 2;
  public static final int GREEN = 3;
  public static final int YELLOW = 4;
  private static final int MAXSTATE = 4;

  private int lightState = RED;

  public int getState() {
     return lightState;
  }
  public void setState (int state) {
    lightState=state;
  }
  public void nextState() {
    lightState++;
    if (lightState>MAXSTATE) {
      lightState=1;
    }
  }

  public Dimension getMinimumSize() {
    return new Dimension(100,260);
  }
  public Dimension getPreferredSize() {
    return getMinimumSize();
  }

  public void paint (Graphics g) {
    switch (lightState) {
      case RED:
        paintLights (g, true, false, false);
        break;
      case REDYELLOW:
        paintLights (g, true, true, false);
        break;
      case GREEN:
        paintLights (g, false, false, true);
        break;
      case YELLOW:
        paintLights (g, false, true, false);
        break;
      }
    }

  private void paintLights (Graphics g,
```

```
      boolean red, boolean yellow, boolean green) {
    g.setColor (Color.black);
    g.fillRect (10, 10, 80, 240);
    if (red) {
      g.setColor (Color.red);
      g.fillOval (20, 20, 60, 60);
    }
    if (yellow) {
      g.setColor (Color.yellow);
      g.fillOval (20, 100, 60, 60);
    }
    if (green) {
      g.setColor (Color.green);
      g.fillOval (20, 180, 60, 60);
    }
  }
}
```

17.3 TrafficLight.html

```
<html>
<applet code="TrafficLight.class" width=300 height=400>
  <p>Without Java, you cannot play with the traffic light, sorry.
</applet>
</html>
```

18 Blinklicht

18.1 BlinkLight.java

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

public class BlinkLight extends Applet
  implements Runnable {

  private boolean runFlag;
  private Image img;
  private Graphics g;
  private Thread t = null;
  private int width, height, diameter, x1, y1;

  public void init() {

    Dimension d = getSize();
    img = createImage(d.width, d.height);
    g=img.getGraphics();
    width = d.width;
    height = d.height;
    if (width < height) {
      diameter = 2 * width / 3;
    }
    else {
      diameter = 2 * height / 3;
    }
    x1 = (width - diameter) / 2;
    y1 = (height - diameter) / 2;
  }

  public void start() {
    if (t == null) {
      t = new Thread (this);
      t.start();
    }
  }
  public void stop() {
    if (t != null) {
      runFlag = false; // t.stop();
      t=null;
    }
  }

  public void run () {
    boolean onOff = false;
    runFlag = true;
    // costant background of image
    g.setColor (Color.black);
    g.fillRect (0, 0, width, height);

    while (runFlag) {

      // prepare image
      if (onOff) {
        g.setColor (Color.yellow);
      }
```

```
      else { // also needed to bypass an error of some 1.1.x compiler optimizations
        g.setColor (Color.black);
      }
      g.fillOval (x1, y1, diameter, diameter);
      // paint the new image (without flicker)
      repaint();

      onOff = ! onOff;
      // wait 1 second
      try { Thread.sleep(1000); }
      catch (InterruptedException e) {}
    }
  }

  public void update (Graphics g) {
    paint(g); // no clear necessary
  }

  public void paint (Graphics g) {
    if (img != null)
      g.drawImage(img,0,0,null);
  }

}
```

18.2 BlinkLight.html

```
<html>
<applet code="BlinkLight.class" width=100 height=300>
  <p>Without Java, you cannot play with the blinking light, sorry.
</applet>
</html>
```

19 zeilenweises Ausdrucken eines Files

19.1 PrintFile.java

```
import java.io.*;

public class PrintFile {
  public static void main (String[] args) {
    String thisLine;
    try {
      BufferedReader in = new BufferedReader (
        new FileReader ("PrintFile.java") );
      while( (thisLine = in.readLine()) != null ) {
        System.out.println(thisLine);
      }
      in.close();
    } catch (Exception e) {
        System.out.println("error " + e);
    }
  }
}
```

20 zeichenweises Kopieren eines Files

20.1 CopyFile.java

```
import java.io.*;

public class CopyFile {
  public static void main (String[] args) {
    int ch;
    try {
      BufferedInputStream in = new BufferedInputStream (
        new FileInputStream ("CopyFile.java") );
      BufferedOutputStream out = new BufferedOutputStream (
        new FileOutputStream ("test.out") );
      while( (ch = in.read()) != -1 ) {
        out.write(ch);
      }
      in.close();
      out.flush();
      out.close();
    } catch (Exception e) {
      System.out.println("error " + e);
    }
  }
}
```

20.2 CopyFast.java

```
import java.io.*;

public class CopyFast {
  public static void main (String[] args) {
    byte buff[] = null;
    int length = 0;
    try {
      File f = new File("CopyFast.java");
      length = (int) f.length();
      buff = new byte[length];
      FileInputStream in = new FileInputStream (f);
      FileOutputStream out = new FileOutputStream ("test.out");
      int copied = 0;
      while (copied < length) {
        copied = copied +
         in.read ( buff, copied, length-copied );
      }
      out.write (buff, 0, length);
      in.close();
      out.flush();
      out.close();
    } catch (Exception e) {
      System.out.println("error " + e);
    }
  }
}
```

20.3 CopyText.java

```
import java.io.*;

public class CopyText {
  public static void main (String[] args) {
    int ch;
    try {
      BufferedReader in = new BufferedReader (
        new FileReader ("CopyText.java") );
      BufferedWriter out = new BufferedWriter (
        new FileWriter ("test.out") );
      while( (ch = in.read()) != -1 ) {
        out.write(ch);
      }
      in.close();
      out.flush();
      out.close();
    } catch (Exception e) {
        System.out.println("error " + e);
    }
  }
}
```

21 Lesen eines Files über das Internet

21.1 PrintURL.java

```
import java.io.*;
import java.net.*;

public class PrintURL {
  public static void main (String[] args) {
    String thisLine;
    String thisURL="http://java.sun.com/products/jdk/1.1/README";

    System.out.println("*** Connecting to " + thisURL);
    System.out.println();
    try {
      BufferedReader in = new BufferedReader (
        new InputStreamReader
          ( (new URL(thisURL)).openStream(), "8859_1" ) );
      while( (thisLine = in.readLine()) != null ) {
        System.out.println(thisLine);
      }
      in.close();
      System.out.println();
      System.out.println("*** Connection finished.");
    } catch (Exception e) {
        System.out.println("error " + e);
    }
  }
}
```

22 Datenbank-Abfragen

22.1 GebJahr.java

```
import java.sql.*;

public class GebJahr {
 public static void main (String[] args) {
  try {
    System.out.println("* Treiber laden");
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    System.out.println("* Datenbank-Verbindung beginnen");
    Connection con = DriverManager.getConnection
        ("jdbc:odbc:TestDB1", "", "");
    con.setReadOnly(true);
    System.out.println("* Statement beginnen");
    Statement stmt = con.createStatement();

    System.out.println("* Abfrage beginnen");
    ResultSet rs = stmt.executeQuery
      ("SELECT Vorname, Geburtsjahr FROM Mitarbeiter ORDER BY Vorname");
    System.out.println("* Ergebnisse anzeigen");
    while (rs.next()) {
      System.out.println( rs.getString(1) + " " + rs.getInt(2) );
    }

    System.out.println("* Statement beenden");
    stmt.close();
    System.out.println("* Datenbank-Verbindung beenden");
    con.close();
  } catch (Exception e) {
      System.out.println("*** Fehler: " + e);
  }
 }
}
```

22.2 Gehalt.java

```
import java.sql.*;

public class Gehalt {
 public static void main (String[] args) {
  double gehalt, summe, durchschnitt;
  int anzahl;
  try {
    System.out.println("* Treiber laden");
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    System.out.println("* Datenbank-Verbindung beginnen");
    Connection con = DriverManager.getConnection
        ("jdbc:odbc:TestDB1", "", "");
    con.setReadOnly(true);
    System.out.println("* Statement beginnen");
    Statement stmt = con.createStatement();

    System.out.println("* Abfrage beginnen");
    ResultSet rs = stmt.executeQuery
      ("SELECT Gehalt FROM Mitarbeiter");
    System.out.println("* Ergebnis berechnen");
    summe = 0.0;
    anzahl = 0;
```

```
    while (rs.next()) {
      gehalt=rs.getDouble(1);
      summe = summe + gehalt;
      anzahl = anzahl + 1;
    }
    durchschnitt = summe / (double) anzahl ;
    System.out.println("Durchscnittsgehalt = " + durchschnitt);

    System.out.println("* Statement beenden");
    stmt.close();
    System.out.println("* Datenbank-Verbindung beenden");
    con.close();
  } catch (Exception e) {
      System.out.println("*** Fehler: " + e);
  }
 }
}
```