TECHNISCHE UNIVERSITÄT MÜNCHEN
FAKULTÄT FÜR INFORMATIK

Lehrstuhl für Sprachen und Beschreibungsstrukturen
Praktikum Grundlagen der Programmierung
F. Forster, T. Gawlitza, A. Flexeder

WS 2007/2008
Lösungsvorschläge zu Blatt 14
1. Februar 2008

Übungen zu Praktikum Grundlagen der Programmierung

Aufgabe 60 Animation (Lösungsvorschlag)

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

public class Kanonenspiel extends Applet implements ActionListener{

    Image kanone,ziel,explosion, leinwand;
    Graphics buffer;
    double x,y, dy,dx,xmax, ymax;
    AnimationThread t;

    public void schuss(){
    x +=dx;
    y -=dy;
    dy -= 0.06;

    buffer.setColor(Color.white);
    buffer.fillRect(0,0,(int)xmax,(int)ymax);
    buffer.drawImage(kanone, 20, 300-kanone.getHeight(this), this);

    //Beende Thread und Applet, falls Boden/Rand des Applets erreicht ist
    if(x<0 || x>xmax || y>ymax){
        t.fertig();
        buffer.drawImage(explosion,(int)x-explosion.getWidth(this)/2,
                 (int)y-explosion.getHeight(this)/2,this);
        repaint();
        return;
    }
    buffer.setColor(Color.black);
    buffer.fillOval((int)x,(int)y,8,8);

    repaint();
    }

    public void init(){
    MediaTracker m = new MediaTracker(this);
```

```
//laden der bilder
ziel = getImage(getCodeBase(), "ziel1.jpeg");
kanone = getImage(getCodeBase(), "kanone1.jpeg");
explosion = getImage(getCodeBase(), "explosion.jpg");

m.addImage(explosion,1);
m.addImage(kanone,2);

xmax = 600;
ymax = 300;

//warten bis alle Bilder geladen sind, bevor auf Leinwand zeichnet
try{
    m.waitForAll();
}catch(Exception e){}

leinwand = createImage((int)xmax,(int)ymax);
buffer = leinwand.getGraphics();
buffer.setColor(Color.white);
buffer.fillRect(0,0,(int)xmax,(int)ymax);
buffer.drawImage(kanone, 20, 300-kanone.getHeight(this), this);
setBackground(Color.white);

//Knopf zum Starten der Animation
Button button = new Button("Feuer!!");
add(button);
button.setBackground(Color.red);
button.addActionListener(this);

repaint();
}


public void feuer(){
//reset-Funktionalitaet
xmax = 600;
ymax = 300;
dx = 4* Math.cos(1.05);
dy = 4* Math.cos(1.05);
x=100;
y=ymax-40;

t = new AnimationThread(this);
t.start();
}

public void actionPerformed(ActionEvent ae){
feuer();
}

public void paint(Graphics g){
g.drawImage(leinwand,0,0,this);
}
public void update(Graphics g){
    paint(g);
}
```

```
}

class AnimationThread extends Thread{
    private Kanonenspiel app;
    boolean running;

    public AnimationThread(Kanonenspiel app){
    this.app = app;
    running=true;
    }
    public void fertig(){
    running = false;
    }

    public void run(){
    while(running){
        try { sleep(40); }catch (InterruptedException ie){ break; }
        app.schuss();
    }
    }
}
```

Aufgabe 61 Schiebe-Puzzle (Lösungsvorschlag)

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

public class Puzzle2 extends Frame {
    class Teil extends Button {
    private int xPos, yPos;

    public Teil(int n, int xp, int yp) {
        super("" + n);
        setSize(40,40);
        setPos(xp, yp);
        addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e) {
            bewegeZuFreierPosition();
            }
        });
    }

    public void bewegeZuFreierPosition() {
        if  ( ( yFreieStelle == yPos // gleiche Zeile
            && Math.abs(xFreieStelle - xPos) == 1 // Nachbarzelle
            ) || (
              xFreieStelle == xPos // gleiche Spalte
              && Math.abs(yFreieStelle - yPos ) == 1 // Nachbarzelle
              )
          ) {
        int xTmp = xFreieStelle;
        int yTmp = yFreieStelle;
        xFreieStelle = xPos;
        yFreieStelle = yPos;
        setPos(xTmp, yTmp);
        }
```

```
    }

    public void setPos(int xp, int yp) {
        xPos = xp;
        yPos = yp;
        setLocation(40 + 40 * xPos, 40 + 40 * yPos);
    }
    }

    static private Random random = new Random(System.currentTimeMillis());
    private Teil[] teile;
    private int breite, xFreieStelle, yFreieStelle;

    public Puzzle2(int breite) {
    super("Puzzle");
    this.breite = breite;
    xFreieStelle = yFreieStelle = breite - 1;
    setLayout(null);
    setSize(80 + 40 * breite, 80 + 40 * breite);
    teile = new Teil[breite * breite - 1];
    for (int x=0; x<breite;x++)
        for(int y=0; y < ((x==breite-1)?(breite-1):breite); y++) {
        Teil teil = teile[breite*x+y] = new Teil(breite*x+y+1,x,y);
        add(teil);
        }

    addWindowListener(new WindowAdapter () {
        public void windowClosing(WindowEvent e) {
            System.exit(0);
        }
        });
    pack();
    }

    public void shuffle() {
    for (int i = 0; i < 500 * breite; i++) {
        int pos = random.nextInt(breite*breite-1);
        teile[pos].bewegeZuFreierPosition();
    }
    }

    public static void main(String[] argv){
    int breite = 4;
    try { breite = Integer.parseInt(argv[0]); } catch (Exception e) { }
    Puzzle p = new Puzzle(breite);
    p.shuffle();
    p.setVisible(true);
    }
}
```