

Course 2DJava: 2D-Computer Graphics with Java

Chapter C1: The Intro Project

Copyright © by V. Miszalok, last update: 25-05-2008

- ↓ [Ein leeres Fenster](#)
- ↓ [DrawString: Hallo Welt](#)
- ↓ [Fenstergröße anzeigen](#)
- ↓ [links, rechts, oben, unten](#)
- ↓ [Line, Rectangle, Ellipse](#)
- ↓ [Zeichenstifte mit zufälligen Farben zeichnen einen Stern mit zufälligen Strichlängen](#)
- ↓ [Polygon zeichnen](#)
- ↓ [Weitere Aufgaben](#)

Ein leeres Fenster

Beschreibung für: Java Version 6, Update 6 und die Eclipse Platform Version 3.3.2.

Starten Sie Eclipse und wählen im Menu:

File → New → Java Project → Project name: intro1 → Finish

File → New → Class → Name: intro1 → Finish

Ersetzen Sie den Code von intro1.java durch:

```
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;

class intro1 extends JFrame {
    public static void main(String args[]) { new intro1(); }
    public intro1() {
        super("intro1: Introduction into Drawing Lines and Shapes");
        setSize(800,600);
        getContentPane().setBackground(Color.white);
        setVisible(true);
    }
}
```

Run → Run. Das Ergebnis ist ein leeres weißes Fenster mit Überschrift. Schließen Sie dieses Fenster.

drawString: Hallo Welt

Schreiben Sie hinter den Konstruktor `public intro1()`, aber noch vor die letzte geschweifte Klammer:

```
public void paint(Graphics g) {
    Graphics2D g2 = (Graphics2D)g;
    super.paint(g2);
    String s0 = "Hello world, here is Intro1 !";
    g2.drawString(s0, 10, 40);
}
```

Fenstergröße anzeigen

Beenden Sie das Programm.

Erweitern Sie die Methode `public void paint(Graphics g)` bis sie so aussieht:

```
public void paint(Graphics g) {
    super.paint(g);
    Graphics2D g2 = (Graphics2D)g;
    int CPwidth  = getContentPane().getWidth();
    int CPheight = getContentPane().getHeight();
    int CPx      = ( getWidth() - CPwidth ) / 2;
    int CPy      = getHeight() - CPheight - CPx;
    g2.translate( CPx, CPy );
    String s0 = "Hello world, here is Intro1 !";
    String s1 = "Change the size of your window by dragging a corner !";
    String s2w = "Form : Width = " + getWidth();
    String s3w = "Client: Width = " + CPwidth;
    String s2h = "  Height= "      + getHeight();
    String s3h = "  Height= "      + CPheight;
    String s4b = "Border Size   = " + CPx;
    String s4t = "  Height of the Title Line =" + (CPy - CPx);
    g2.drawString(s0      , 10, 20);
    g2.drawString(s1      , 10, 40);
    g2.drawString(s2w + s2h, 10, 60);
    g2.drawString(s3w + s3h, 10, 80);
    g2.drawString(s4b + s4t, 10, 100);
}
```

Ziehen Sie am Fensterrand oder an den Fensterecken und beobachten Sie den Fensterinhalt.

links, rechts, oben, unten

Beenden Sie das Programm.

Schreiben Sie sechs weitere Zeilen in die Methode `public void paint(Graphics g)`:

```
// Version 3
Point mid = new Point(CPwidth/2, CPheight/2);
g2.drawString("left"   ,10      , mid.y);
g2.drawString("right"  ,CPwidth-30, mid.y);
g2.drawString("top"    ,mid.x    , 20);
g2.drawString("bottom" ,mid.x    , CPheight-10);
```

Line, Rectangle, Ellipse

Beenden Sie das Programm.

Schreiben Sie folgende weitere Zeilen in die Methode `public void paint(Graphics g)`:

```
// Version 4
int w5 = CPwidth  / 5;
int h5 = CPheight / 5;
g2.drawLine(0      , 0      , w5      , h5      );
g2.drawLine(4*w5  , 4*h5    , CPwidth  , CPheight);
g2.drawLine(CPwidth,0      , 4*w5    , h5      );
g2.drawLine(0      , CPheight, w5      , 4*h5    );
g2.drawRect(w5     , h5     , 3*w5    , 3*h5    );
g2.drawOval(w5     , h5     , 3*w5    , 3*h5    );
```

Zeichenstifte mit zufälligen Farben zeichnen einen Stern mit zufälligen Strichlängen

Beenden Sie das Programm.

Schreiben Sie folgende weitere Zeilen in die Methode `public void paint(Graphics g)`:

```
// Version 5
Polygon splash = new Polygon();
Random r = new Random();
g2.setStroke(new BasicStroke(15, BasicStroke.CAP_ROUND, BasicStroke.JOIN_ROUND));
int nn = 120;
double arcus = 2. * Math.PI / nn; // angular increment
double radius_x = 1.35 * w5; // horizontal elliptical radius
double radius_y = 1.35 * h5; // vertical elliptical radius
for (int i=0; i < nn; i++) {
    Color multicolor = new Color( r.nextInt(255), r.nextInt(255), r.nextInt(255) );
    double factor = Math.max( 0.25, r.nextDouble() );
    double cosinus = radius_x * factor * Math.cos( i * arcus );
    double sinus = radius_y * factor * Math.sin( i * arcus );
    Point end = new Point(mid.x + (int)cosinus, mid.y + (int)sinus );
    g2.setColor(multicolor);
    g2.drawLine(mid.x, mid.y, end.x, end.y);
    splash.addPoint(end.x, end.y);
}
```

Polygon zeichnen

Beenden Sie das Programm.

Schreiben Sie folgende weitere Zeilen in die Methode `public void paint(Graphics g)`:

```
//Version 6
g2.setColor(Color.red);
g2.fillPolygon(splash);
g2.setColor(Color.blue);
g2.drawString("Splash !", mid.x-20, mid.y);
```

Weitere Aufgaben

Wandeln Sie den Code ab und erzeugen Varianten.

Löschen Sie die gesamte Directory `intro1` in Ihrem Eclipse-Workspace-Verzeichnis

Starten Sie Eclipse wieder und erzeugen dasselbe Programm so oft, bis Sie `intro1` ohne Hilfsmittel programmieren können.

Benutzen Sie Drag&Drop beim Schreiben von Code = verschieben und kopieren mit Maus mit/ohne Strg-Taste.

Erfinden und erproben Sie neue Varianten des Programms (in Form von neuen Projekten `intro2`, `intro3` usw. nach obigem Muster), z.B. bunte Striche waagrecht parallel, senkrecht parallel, bunte Rechtecke und Ellipsen an zufälligen Stellen etc.