

Course 2DJava: 2D-Computer Graphics with Java

Chapter C3: The XML Project

Copyright © by V. Miszalok, last update: 26-05-2008

- ↓ [Ein leeres Fenster](#)
- ↓ [Malprogramm mit dynamischem Array](#)
- ↓ [File-Menu](#)
- ↓ [Text-File schreiben](#)
- ↓ [XAML-File schreiben](#)
- ↓ [SVG-File schreiben](#)
- ↓ [Text-, XAML- und SVG-Files lesen](#)
- ↓ [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: XML1 → Finish

File → New → Class → Name: XML1 → Finish

Ersetzen Sie den Code von XML1.java durch:

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

class XML1 extends JFrame {public static void main(String args[]) { new XML1();}
    Point p0, p1;
    Vector polygon = new Vector();
    StringBuffer polygon_string = new StringBuffer();
    Stroke lineStroke = new BasicStroke(5);
    Color lineColor = Color.red;
    public XML1() {
        super("XML1: Read and Write Polygons");
        setSize(800, 600);
        show();
    }
}
```

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

Malprogramm mit dynamischem Array

Beenden Sie Ihr Programm XML1.

Erweitern Sie den Konstruktor `public XML1()` und schreiben Sie zusätzlich eine `processWindowEvent`-Methode und eine `DrawArea`-Klasse, welche die Maus-Ereignisse behandelt:

```
public XML1() {
    super("XML1: Read and Write Polygons");
    setSize(800, 600);
    Container p = getContentPane();
    p.setLayout(new BorderLayout());
    p.add("Center", new DrawArea());
    show();
}

protected void processWindowEvent(WindowEvent e) {
    super.processWindowEvent(e);
    if (e.getID() == WindowEvent.WINDOW_CLOSING)
        System.exit(0);
}

class DrawArea extends JComponent implements MouseListener, MouseMotionListener {
    public DrawArea() {
        addMouseListener(this);
        addMouseMotionListener(this);
    }
}
```

```

public void paint(Graphics g) {
    Graphics2D g2 = (Graphics2D) g;
    Dimension size = getSize();
    int fontHeight = g.getFont().getSize();
    g.clearRect(0, 0, size.width, 2*fontHeight);
    g.setColor(Color.red);
    g.drawString("Press the left mouse button and move!", size.width/2-50, fontHeight);
    g.setColor(Color.black);
    g.drawString(polygon_string.toString(), 0, 2*fontHeight);
    g2.setStroke(lineStroke);
    g.setColor(lineColor);
    for (int i=0; i < polygon.size()-1; i++) {
        Point p0 = (Point) polygon.get(i);
        Point p1 = (Point) polygon.get(i+1);
        g.drawLine(p0.x, p0.y, p1.x, p1.y);
    }
}

public void mouseClicked(MouseEvent e) {}
public void mouseEntered(MouseEvent e) {}
public void mouseExited (MouseEvent e) {}
public void mouseMoved (MouseEvent e) {}

public void mousePressed(MouseEvent e) {
    polygon.clear();
    repaint();
    polygon.add(p0 = new Point(e.getX(), e.getY()));
}

public void mouseReleased(MouseEvent e) {
    if (polygon.size() < 2) return;
    polygon_string.setLength(0);
    polygon_string.append("\n");
    for (int i=0; i < polygon.size(); i++) {
        Point p = (Point) polygon.get(i);
        if (i > 0) polygon_string.append(", ");
        polygon_string.append((int)p.x);
        polygon_string.append(",");
        polygon_string.append((int)p.y);
    }
    polygon_string.append("\n");
    p0 = null;
    repaint();
}

public void mouseDragged(MouseEvent e) {
    p1 = new Point(e.getX(), e.getY());
    if (p1.distance(p0) < 10) return;
    Graphics g = getGraphics();
    Graphics2D g2 = (Graphics2D)g;
    g2.setStroke(lineStroke);
    g.setColor(lineColor);
    g.drawLine(p0.x, p0.y, p1.x, p1.y);
    polygon.add(p1);
    p0 = p1;
}
} //end of class DrawArea
} //end of public class XML1

```

Erproben Sie die Persistenz der Zeichnung nach verkleinern.

File-Menu

Beenden Sie Ihr Programm XML1.

Fügen Sie der Klassendeklaration von class XML1 extends JFrame die Erweiterung implements ActionListener hinzu:

```
public class XML1 extends JFrame implements ActionListener {
    public static void main(String args[]) { new XML1(); }
```

Schreiben Sie die folgende Variablendeklarationen in den Kopf der Klasse XML1() noch vor die Zeile Point p0, p1;

```
JMenuItem miSaveTXT = new JMenuItem("SaveAsTXT");
JMenuItem miSaveXAML = new JMenuItem("SaveAsXAML");
JMenuItem miSaveSVG = new JMenuItem("SaveAsSVG");
JMenuItem miRead = new JMenuItem("Read", KeyEvent.VK_R);
JMenuItem miExit = new JMenuItem("Exit", KeyEvent.VK_E);
JMenu menuFile = new JMenu("File");
```

Schreiben Sie folgende weiteren Befehle zu Beginn des Konstruktors von public XML1() unter den Befehl super("XML1: Read and Write Polygons");:

```
JMenuBar menuBar = new JMenuBar();
setJMenuBar(menuBar);
menuFile.setMnemonic(KeyEvent.VK_F);
menuBar.add(menuFile);
JMenuItem[] mi = {miSaveTXT, miSaveXAML, miSaveSVG, miRead, miExit};
for (int i=0; i < mi.length; i++) {
    mi[i].addActionListener(this);
    menuFile.add(mi[i]);
}
```

Schreiben Sie folgende neuen Methoden in die Klasse XML1() hinter der letzten Klammer des Konstruktors vor die Methode protected void processWindowEvent(WindowEvent e).

```
public void actionPerformed(ActionEvent e) {
    Object source = e.getSource();
    if (source == miSaveTXT) menuFileSaveAsTXT();
    else if (source == miSaveXAML) menuFileSaveAsXAML();
    else if (source == miSaveSVG) menuFileSaveAsSVG();
    else if (source == miRead) menuFileRead();
    else if (source == miExit) menuFileExit();
}
private void menuFileSaveAsTXT() {}
private void menuFileSaveAsXAML() {}
private void menuFileSaveAsSVG() {}
private void menuFileRead() {}
private void menuFileExit() { System.exit(0); }
```

Erproben Sie das Menü mit den noch funktionslosen Menüpunkten.

Text-File schreiben

Beenden Sie Ihr Programm XML1.

Schreiben Sie folgenden Code in die leere Funktion `void menuFileSaveAsTXT()` und rücken Sie die `{}`-Klammern der Funktion richtig ein:

```

ByteArrayOutputStream outs = new ByteArrayOutputStream();
PrintWriter out = new PrintWriter(outs);
out.println("This file comes from Prof. Miszalok's XML1");
out.println("polyline points ");
out.println(polygon_string);
out.flush();
saveDialogWrite("txt files", ".txt", outs.toByteArray());

```

Schreiben Sie eine neue Methode `saveDialogWrite(..)` und eine neue Klasse `SimpleFileFilter{..}` in die Klasse XML1() am besten unterhalb der Zeile: `private void menuFileExit() { System.exit(0); }`

```

private void saveDialogWrite(String defaultDescription,
                             String defaultExt, byte[] data){
    JFileChooser chooser = new JFileChooser(".\\");
    chooser.setFileFilter(new SimpleFileFilter(defaultDescription +
        " (*" + defaultExt + ")", defaultExt));
    if (chooser.showSaveDialog(this) == JFileChooser.APPROVE_OPTION)
        if (!chooser.getSelectedFile().isDirectory()) {
            String fn = chooser.getSelectedFile().getAbsolutePath();
            if (!fn.toLowerCase().endsWith(defaultExt.toLowerCase())) fn += defaultExt;
            try{
                FileOutputStream stream = new FileOutputStream(fn);
                stream.write(data); stream.close();
            }
            catch (Exception e) { e.printStackTrace(); }
        }
}

class SimpleFileFilter extends javax.swing.filechooser.FileFilter {
    String description;
    String ext;
    public SimpleFileFilter(String description, String ext) {
        this.description = description;
        this.ext = ext.toLowerCase();
    }
    public boolean accept(File f) {
        if ((f.isDirectory()) ||
            (f.getAbsolutePath().toLowerCase().endsWith(ext))) return true;
        return false;
    }
    public String getDescription() { return description; }
}

```

Malen Sie eine 1 und speichern Sie diese mit `SaveAsTXT` unter `"one.txt"`, malen Sie eine 2 und speichern Sie diese unter `"two.txt"` usw. (Die Extension `.txt` ist nicht obligatorisch, wenn sie fehlt, hängt XML1 diese automatisch an Ihren Filenamen an.)

Kontrollieren Sie, ob Sie sich beim Speichern in der Directory `c:\temp\XML1` befinden. (Ansonsten müssen Sie `one.txt` später mühsam suchen.)

Starten Sie Notepad oder Editor oder Textpad. Öffnen Sie das File `c:\temp\XML1\one.txt`. Vergleichen Sie die Zeilen des Files mit dem Code von `void menuFileSaveAsTXT()`.

Ändern Sie in der Funktion `void menuFileSaveAsTXT()` den String `"This file comes from Prof. Miszaloks XML1"` übersetzen Sie neu und versuchen Sie alles noch einmal.

Gehen Sie in die Directory `c:\temp\XML1`. Doppelklicken Sie dort auf `one.txt`. Ihr Standardeditor (meistens Notepad) wird den Inhalt anzeigen.

XAML-File schreiben

Beenden Sie Ihr Programm XML1.

Schreiben Sie folgenden Code in die leere Funktion `void menuFileSaveAsXAML()` und rücken Sie die `{}`-Klammern der Funktion richtig ein:

```

ByteArrayOutputStream outs = new ByteArrayOutputStream();
PrintWriter out = new PrintWriter(outs);
out.println("<Canvas xmlns =\"http://schemas.microsoft.com/winfx/2006/xaml/presentation\"");
out.println("        xmlns:x=\"http://schemas.microsoft.com/winfx/2006/xaml\">");
out.println("<!--This file comes from Prof. Miszalok's XML1<-->");
out.println("<Polyline Points = ");
out.println(polygon_string + " Stroke=\"Red\" StrokeThickness=\"4\" />");
out.println("</Canvas>");
out.flush();

```

```
saveDialogWrite("xaml files", ".xaml", outs.toByteArray());
```

Malen Sie eine 1 und speichern Sie diese mit `SaveAsXAML` unter `"one.xaml"`, malen Sie eine 2 und speichern Sie diese unter `"two.xaml"` usw. (Die Extension `.xaml` ist nicht obligatorisch, wenn sie fehlt, hängt XML1 diese automatisch an Ihren Filenamen an.)

Kontrollieren Sie, ob Sie sich beim Speichern in der Directory `c:\temp\XML1` befinden.

(Ansonsten müssen Sie `one.html` später mühsam suchen.)

Starten Sie Notepad oder Editor oder Textpad. Öffnen Sie das File `c:\temp\XML1\one.xaml`.

Vergleichen Sie die Zeilen des Files mit dem Code von `void menuFileSaveAsXAML()`.

Ändern Sie in der Funktion `void menuFileSaveAsXAML()`

den String `"This file comes from Prof. Miszaloks XML1"` übersetzen Sie neu und versuchen Sie alles noch einmal.

Gehen Sie in die Directory `c:\temp\XML1`. Doppelklicken Sie dort auf `one.xaml`.

Falls der Internet Explorer ab Version 7 Ihr Standardbrowser ist, wird dieser Ihr File interpretieren und darstellen. Frühere Versionen und Mozilla Firefox tun das nicht freiwillig.

Man muss in diesem Fall ein Plug-In installieren: <http://www.microsoft.com/silverlight/install.aspx>.

SVG-File schreiben

Beenden Sie Ihr Programm XML1.

Schreiben Sie folgenden Code in die leere Funktion `void menuFileSaveAsSVG()` und rücken Sie die `{}`-Klammern der Funktion richtig ein:

```

ByteArrayOutputStream outs = new ByteArrayOutputStream();
PrintWriter out = new PrintWriter(outs);
out.println("<?xml version=\"1.0\" encoding=\"UTF-8\" standalone=\"no\"?>");
out.println("<svg xmlns=\"http://www.w3.org/2000/svg\" width=\"100%\" height=\"100%\">");
out.println("<title>This file comes from Prof. Miszalok's XML1</title>");
out.println("<polyline fill = \"none\" stroke = \"red\" stroke-width = \"4\" points = ");
out.println(polygon_string);
out.println(" />");
out.println("</svg>");
out.flush();
saveDialogWrite("svg files", ".svg", outs.toByteArray());

```

Malen Sie eine 1 und speichern Sie diese mit `SaveAsSVG` unter `"one.svg"`, malen Sie eine 2 und speichern Sie diese unter `"two.svg"` usw. (Die Extension `.svg` ist nicht obligatorisch, wenn sie fehlt, hängt XML1 diese automatisch an Ihren Filenamen an.)

Kontrollieren Sie, ob Sie sich beim Speichern in der Directory `c:\temp\XML1` befinden.

(Ansonsten müssen Sie `one.svg` später mühsam suchen.)

Starten Sie Notepad oder Editor oder Textpad. Öffnen Sie das File `c:\temp\XML1\one.svg`.

Vergleichen Sie die Zeilen des Files mit dem Code von `void menuFileSaveAsSVG()`.

Ändern Sie in der Funktion `void menuFileSaveAsSVG()`

den String `"This file comes from Prof. Miszaloks XML1"` übersetzen Sie neu und versuchen Sie alles noch einmal.

Downloaden Sie den Adobe SVG Viewer (ASV) von www.adobe.com/svg/viewer/install (2.3 MByte) und installieren Sie diesen.

Der ASV ist als Plugin für Microsoft Internet Explorer ab 4.0, für Firefox und für Opera verfügbar.

Gehen Sie in die Directory `c:\temp\XML1`. Doppelklicken Sie dort auf `one.svg`.

Falls das ASV-Plugin richtig installiert ist, wird Ihr Browser das File interpretieren und darstellen.

Text-, XAML- und SVG-Files lesen

Beenden Sie Ihr Programm XML1.

Schreiben Sie folgenden Code in die leere Funktion `void menuFileRead()` und rücken Sie die `{}`-Klammern der Funktion richtig ein:

```
JFileChooser chooser = new JFileChooser(".\\");
if (chooser.showOpenDialog(this) != JFileChooser.APPROVE_OPTION) return;
if (chooser.getSelectedFile().isDirectory()) return;
String fn = chooser.getSelectedFile().getAbsolutePath();
try {
    StringBuffer sb = new StringBuffer();
    BufferedReader br = new BufferedReader(new FileReader(fn));
    while (true) {
        String s = br.readLine();
        if (s == null) break;
        sb.append(s); sb.append('\n');
    }
    br.close();
    String file_string = sb.toString();
    int n0, n1, n2, n3;
        n0 = file_string.indexOf("polyline"    );
    if (n0 < 0) n0 = file_string.indexOf("Polyline"    );
    if (n0 < 0) return;
        n1 = file_string.indexOf("points", n0+8);
    if (n1 < 0) n1 = file_string.indexOf("Points", n0+8);
    if (n1 < 0) return;
    if ((n2 = file_string.indexOf("\""      , n1+6)) < 0) return;
    if ((n3 = file_string.indexOf("\""      , n2+1)) < 0) return;
    StringTokenizer st = new StringTokenizer(file_string.substring(n2+1, n3), ",");
    polygon.clear();
    while (st.hasMoreTokens())
        polygon.add(new Point(Integer.parseInt(st.nextToken().trim()),
            Integer.parseInt(st.nextToken().trim())));
    repaint();
} // end try
catch (Exception e) { e.printStackTrace(); }
```

Sie können jetzt alle Ihre gespeicherten Files `one.*` und `two.*` mit XML1 lesen und anzeigen.

Weitere Aufgaben

Öffnen Sie ein Browser-Fenster mit einer Suchmaschine Ihrer Wahl.

Geben Sie in das Such-Feld `Java` und einen oder mehrere (durch Leerzeichen getrennte) der folgenden Suchbegriffe ein: `StringBuffer`, `Vector`, `ArrayList`, `JMenu`, `JFileChooser`, `FileFilter`, `FileOutputStream`, `BufferedReader`, `StringTokenizer`, `indexOf` ein.

Mit diesem Wissen wandeln Sie den Code ab und erzeugen Varianten.

Erfinden und erproben Sie neue Varianten des Programms (in Form von neuen Projekten `XML2`, `XML3` usw. nach obigem Muster).

Einfach: Ändern Sie Farbe und Pinseldicke im Kopf von `class XML1` und in `void menuFileSaveAsVML()`, so dass beide zu einander passen.

Nicht einfach: Versuchen Sie eine Version zu schreiben mit einem übergeordneten `Vector`, die Unter-`Vektoren` `polygon[i]` enthält, um mehrere Polygone gleichzeitig zu verwalten, zu schreiben und zu lesen.

Lesen Sie: <http://de.wikipedia.org/wiki/XAML>.

Lesen Sie: www.adobe.com/svg/overview/svg.htm.