

Putting the 'Free' into JFreeChart

25 February 2006

Dave Gilbert
JFreeChart Project Leader

Overview

- The Java Trap;
- JFreeChart;
 - Java2D (Graphics2D);
- The Free Stack:
 - Cairo (CairoGraphics2D);
 - GNU Classpath;
 - samples;
- A Real World Application:
 - StatCVS;

I. The Java™ Trap

- “Free But Shackled – The Java Trap” by Richard Stallman:
 - <http://www.gnu.org/philosophy/java-trap.html>
 - 12 April 2004;
- Executive Summary: Free software that depends on a non-free platform is not 100% free;
 - JFreeChart falls into this category;
 - so does a vast collection of free and open source software;
 - a motivation for GNU Classpath;

II. JFreeChart

- a free library for creating charts:
 - <http://www.jfree.org/jfreechart/>
 - JFreeChart 1.0.0 released in December 2005;
 - can be used for server-side (servlet/JSP) or client-side development (Swing-based applications or applets);
 - requires Java 1.3.1 or later;
 - relies (heavily) on the Java2D API;
 - GNU Lesser General Public Licence (LGPL);

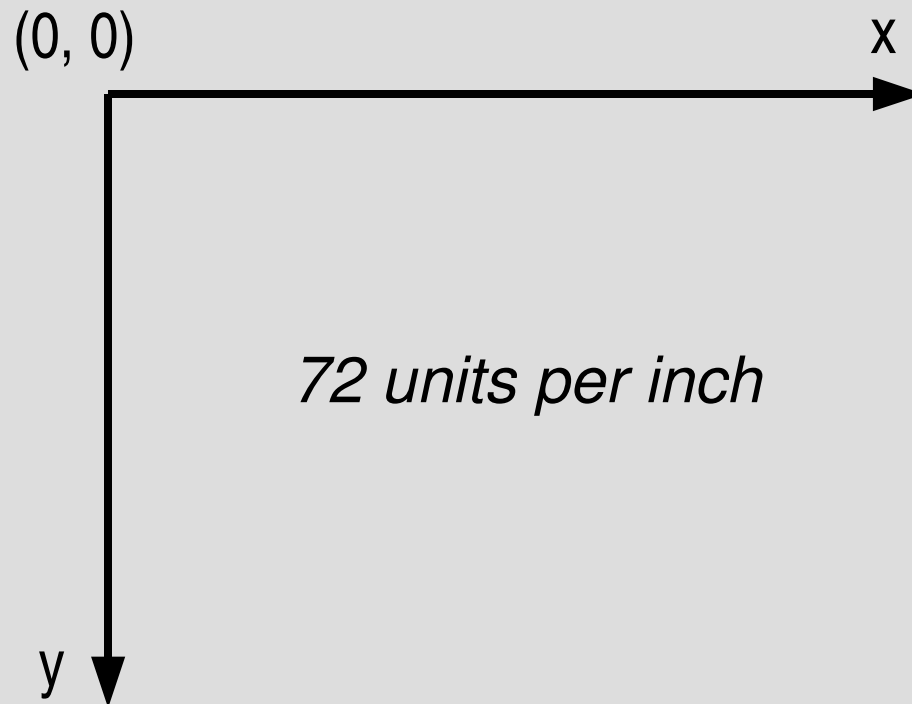
JFreeChart (2)

- project started December 1999;
- some statistics:
 - 32 packages;
 - 462 classes/interfaces;
 - 90+ contributors;
 - an estimated 40,000 developers use JFreeChart;
- funding:
 - JFreeChart Developer Guide;
 - consulting;

Java2D

- Java's device independent vector graphics API:
 - introduced in JDK 1.2;
 - centers around abstract class `Graphics2D`;
 - small API, big implementation;
- core library provides implementations for output to:
 - screen: `javax.swing.JComponent`;
 - printers: `java.awt.print.PrinterJob`;
 - images: `java.awt.BufferedImage`;

Java2D - User Space

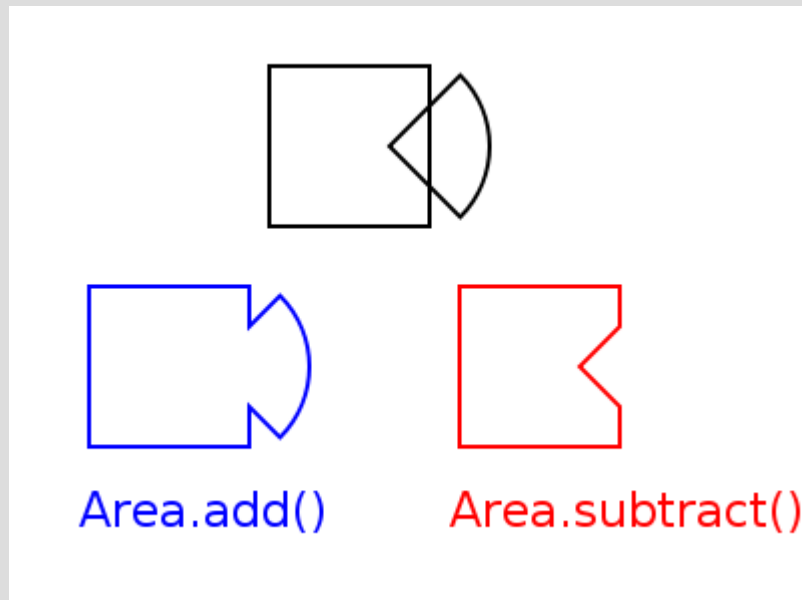


Java2D - Geometry

- Shapes via `java.awt.Shape` interface:
 - points: `java.awt.geom.Point2D`;
 - lines: `java.awt.geom.Line2D`;
 - rectangles: `java.awt.geom.Rectangle2D`;
 - polygons: `java.awt.Polygon`;
 - arcs: `java.awt.geom.Arc2D`;
 - curves:
 - `java.awt.geom.QuadCurve2D/CubicCurve2D`;
 - paths: `java.awt.geom.GeneralPath`;

Java2D – Constructive Area Geometry

- `java.awt.geom.Area` class;
 - allows addition and subtraction of arbitrary shapes;
 - for example, a rectangle and an arc:



Graphics2D

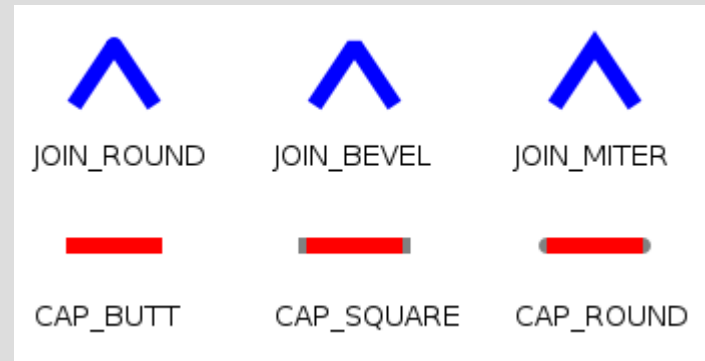
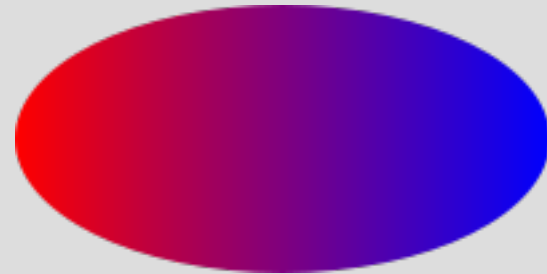
- set current color:
 - `public void setPaint(Paint);`
- fill shapes
 - `public void fill(Shape);`
- set pen style:
 - `public void setStroke(Stroke);`
- draw shapes:
 - `public void draw(Shape);`
- draw text:
 - `public void setFont(Font);`
 - `public void drawString(String, float, float);`

Java2D - Transformations

- **java.awt.AffineTransform:**
 - `public Shape createTransformedShape(Shape pSrc);`
- **translation:**
 - `public static AffineTransform getTranslateInstance(double tx, double ty)`
- **rotation:**
 - `public static AffineTransform getRotateInstance(double theta, double x, double y)`
- **scale:**
 - `public static AffineTransform getScaleInstance(double sx, double sy)`
- **shear:**
 - `public static AffineTransform getShearInstance(double shx, double shy)`

Java2D - Rendering

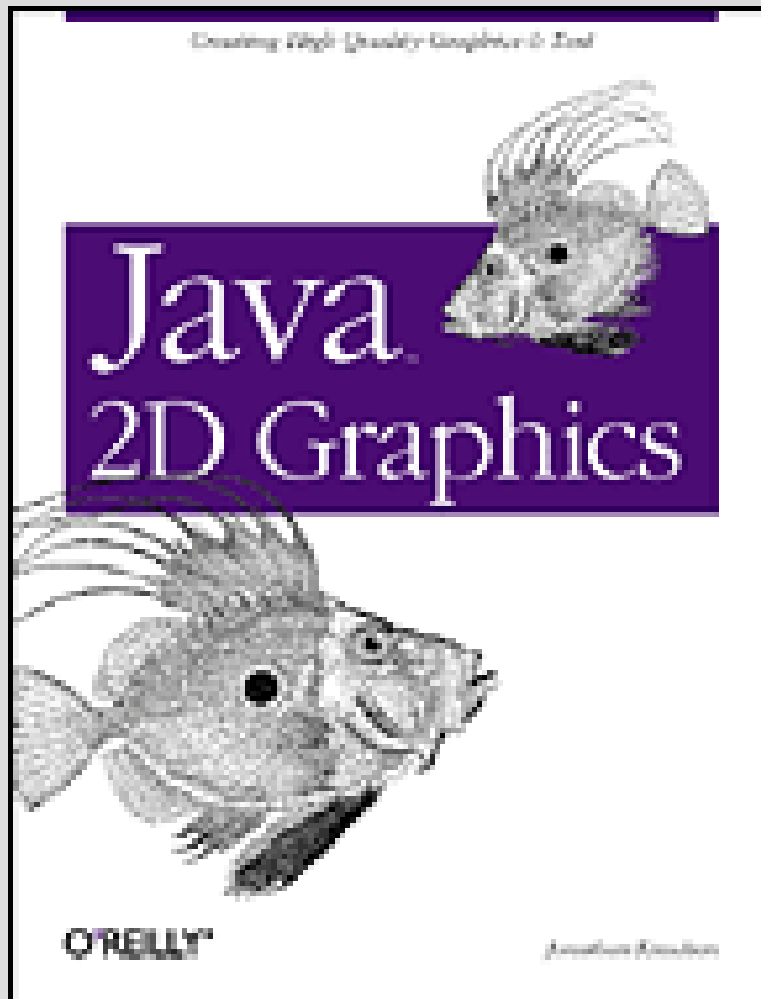
- colors:
 - `java.awt.Paint`;
 - RGB;
 - gradients;
 - alpha-transparency;
- pen style:
 - `java.awt.Stroke`;
 - line width control;
 - line decorations;
 - dashed lines;



Java2D – Rendering (2)

- text:
 - font control;
 - Unicode support;
 - text outlines as shapes;
 - all transformations possible;
- images:
 - `java.awt.BufferedImage`
- hints:
 - speed vs quality, for example antialiasing;
 - `java.awt.RenderingHints`

Java2D Book



- Recommended:
 - Jonathan Knudsen
 - published by O'Reilly
 - First Edition, May 1999
- <http://www.oreilly.com/catalog/java2d/>

Graphics2D - Implementations

- external implementations available for:
 - Adobe Portable Document Format (PDF):
 - iText – MPL;
 - <http://www.lowagie.com/iText/>
 - Scalable Vector Graphics (SVG):
 - Batik - Apache license;
 - <http://xmlgraphics.apache.org/batik/>
 - Encapsulated Postscript (EPS):
 - jibble.org – GPL;
 - <http://www.jibble.org/epsgraphics/>

Java2D in GNU Classpath?

- still a lot of work to be done;
- an initial implementation, based on Cairo, not yet enabled by default;

III. The Free Stack

- What is it and what can it do?
 - Cairo;
 - cairo-java;
 - CairoGraphics2D;
 - GNU Classpath;
- a demo application;
- a range of examples;

Cairo

- a vector graphics library;
 - written in C;
 - bindings for many programming languages, including Java;
 - <http://www.cairographics.org>
 - Redhat;
- widely used:
 - GTK+;
 - Eclipse on Linux (backend for SWT graphics);
 - Mono;
 - GNU Classpath (work in progress);

cairo-java

- Java bindings from the Java-Gnome project
 - <http://java-gnome.sourceforge.net/>

CairoGraphics2D

- need a bridge between:
 - JFreeChart 1.0.0;
 - Cairo;
- wrote a `CairoGraphics2D` class:
 - incomplete but functional;
 - maps `Graphics2D` API calls (by JFreeChart) to Cairo methods (via `cairo-java`);
- helper class `CairoImage` to:
 - provide `Graphics2D` instance;
 - write to PNG file

Sample Program

```
public static void main(String[] args) {
    CategoryDataset dataset = createDataset();
    JFreeChart chart = createChart(dataset);
    CairoImage image = new CairoImage(600, 400);
    Graphics2D g2 = image.createGraphics2D();
    chart.draw(g2,
        new Rectangle2D.Double(0.0, 0.0, 600.0, 400.0));
    image.saveToPNG(new File("LineChartTest.png"));
}
```

Sample Program (2)

```
private static CategoryDataset createDataset() {
    DefaultCategoryDataset dataset = new
        DefaultCategoryDataset();
    dataset.addValue(212, "Classes", "JDK 1.0");
    dataset.addValue(504, "Classes", "JDK 1.1");
    dataset.addValue(1520, "Classes", "SDK 1.2");
    dataset.addValue(1842, "Classes", "SDK 1.3");
    dataset.addValue(2991, "Classes", "SDK 1.4");
    return dataset;
}
```

Sample Program (3)

```
private static JFreeChart createChart(CategoryDataset dataset) {  
  
    // create the chart...  
    JFreeChart chart = ChartFactory.createLineChart(  
        "Java Standard Class Library",    // chart title  
        "Release",                        // domain axis label  
        "Class Count",                   // range axis label  
        dataset,                          // data  
        PlotOrientation.VERTICAL,        // orientation  
        false,                            // include legend  
        false,                            // tooltips  
        false,                            // urls  
    );  
}
```

Sample Program (4)

```
// title
chart.setTitle(new TextTitle("Line Chart Demo", new
    Font("Sans", Font.BOLD, 18)));

// subtitle
chart.addSubtitle(new TextTitle("Number of Classes By
    Release", new Font("Sans", Font.BOLD, 14)));
TextTitle source = new TextTitle(
    "Source: Java In A Nutshell (4th Edition) "
    + "by David Flanagan (O'Reilly)",
    new Font("Sans", Font.PLAIN, 12));
source.setPosition(RectangleEdge.BOTTOM);
source.setHorizontalAlignment(HorizontalAlignment.RIGHT);
chart.addSubtitle(source);
```


Sample Program (5)

```
// colors
chart.setBackgroundPaint(Color.white);
CategoryPlot plot = (CategoryPlot) chart.getPlot();
plot.setBackgroundPaint(Color.lightGray);
plot.setRangeGridlinePaint(Color.white);

// customise the range axis...
NumberAxis rangeAxis = (NumberAxis) plot.getRangeAxis();
rangeAxis.setLabelFont(new Font("Sans", Font.BOLD, 14));
rangeAxis.setTickLabelFont(new Font("Sans", Font.PLAIN, 12));
rangeAxis.setStandardTickUnits(
    NumberAxis.createIntegerTickUnits());
```

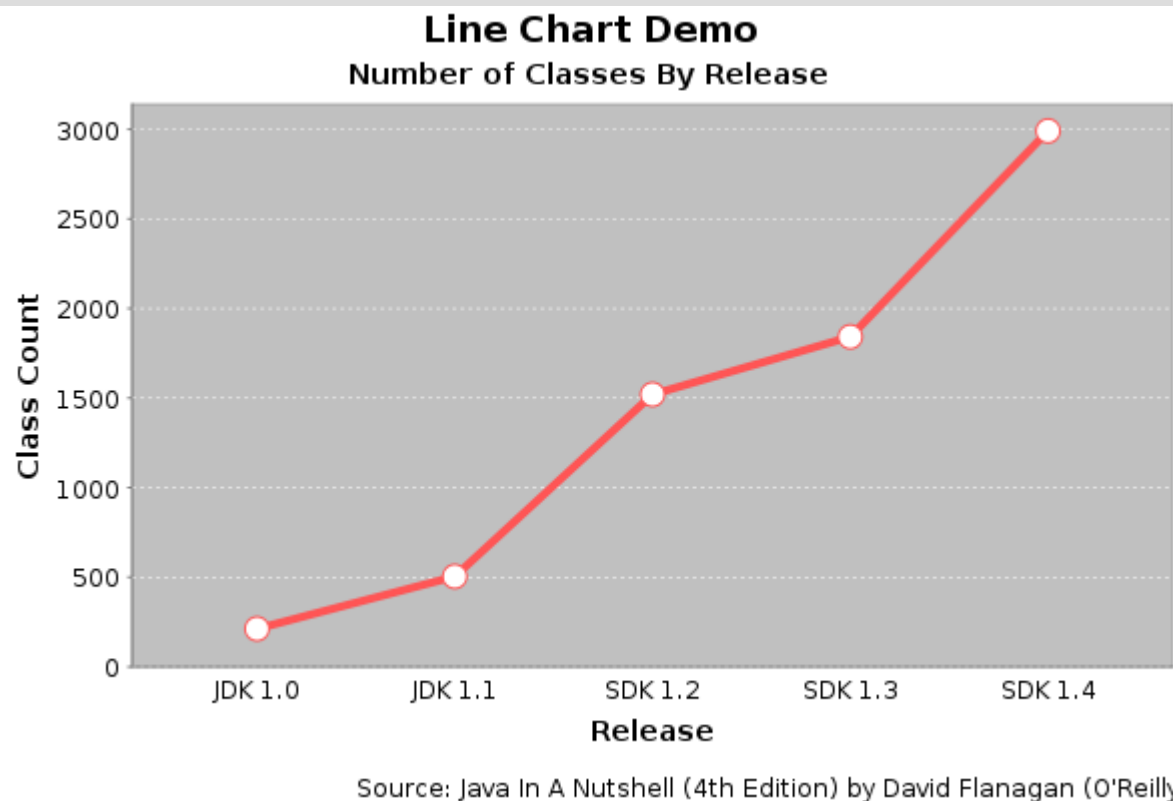
Sample Program (6)

```
// customise the renderer...
LineAndShapeRenderer renderer
    = (LineAndShapeRenderer) plot.getRenderer();
renderer.setShapesVisible(true);
renderer.setDrawOutlines(true);
renderer.setUseFillPaint(true);
renderer.setFillPaint(Color.white);
renderer.setSeriesStroke(0, new BasicStroke(4.0f));
renderer.setSeriesShape(0,
    new Ellipse2D.Double(-6, -6, 12, 12));

CategoryAxis domainAxis = plot.getDomainAxis();
domainAxis.setLabelFont(new Font("Sans", Font.BOLD, 14));
domainAxis.setTickLabelFont(new Font("Sans", Font.PLAIN, 12));

return chart;
}
```

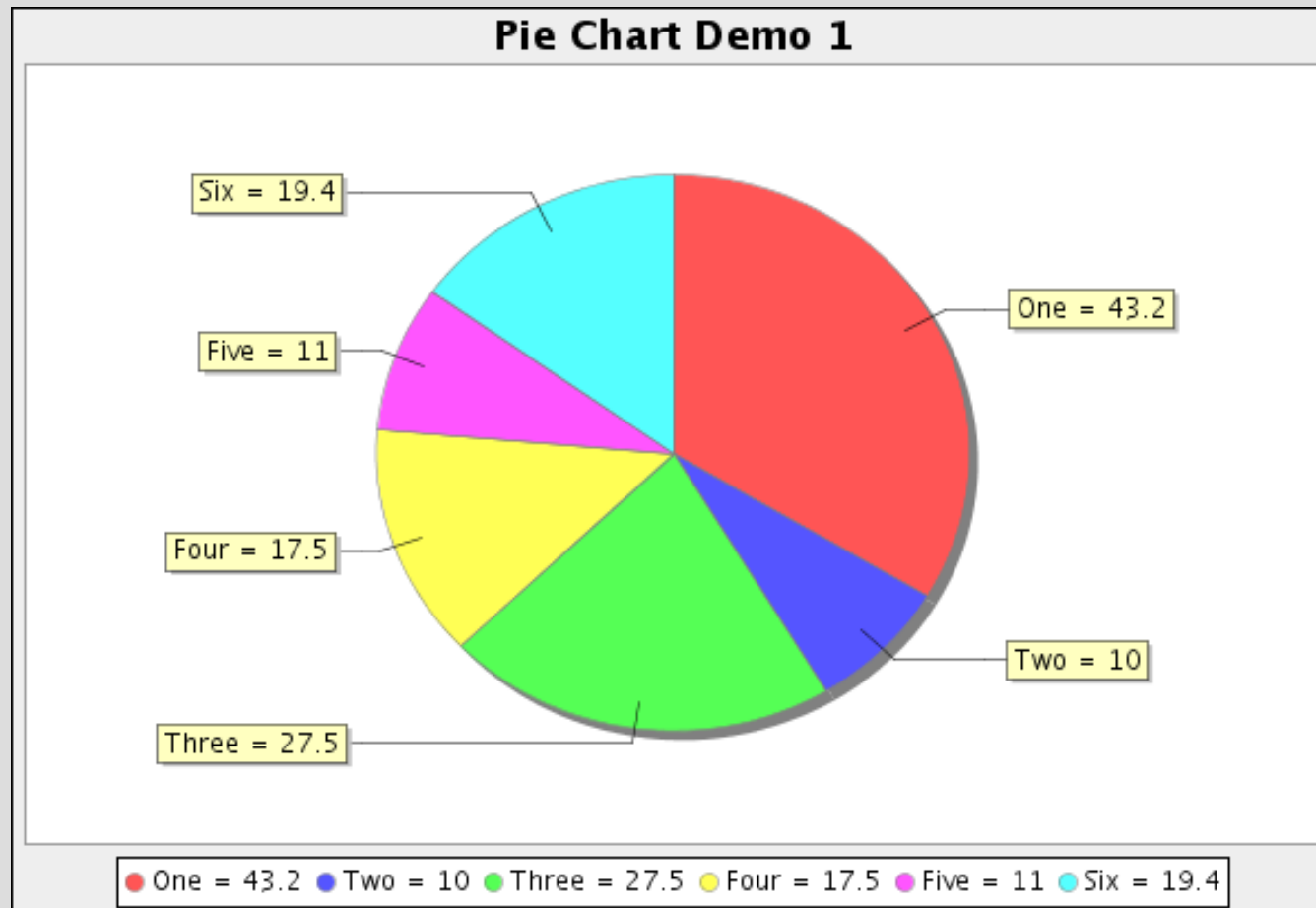
Sample Program (7)



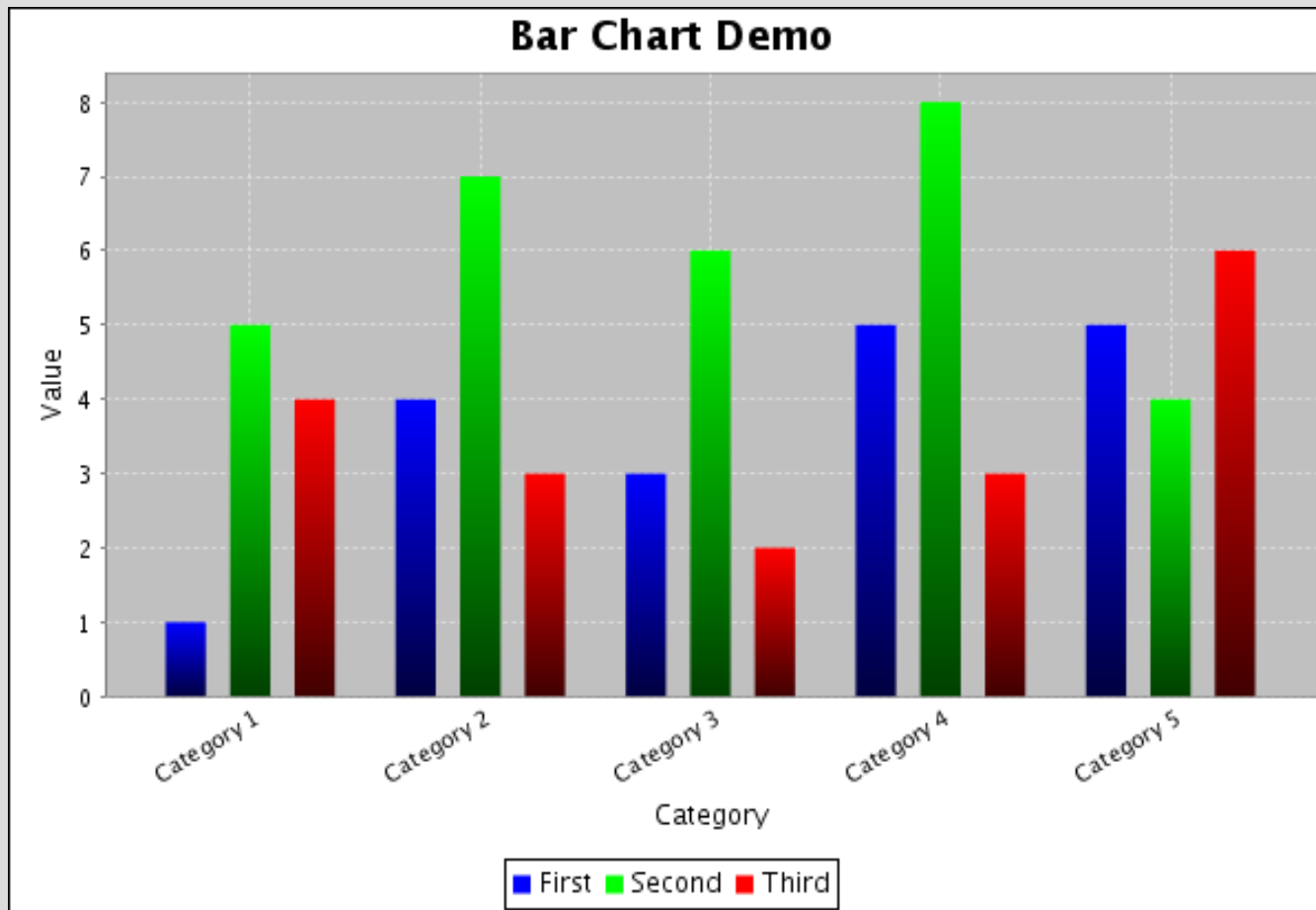
More Examples

- all generated with a Free software stack:
 - JamVM 1.4.1;
 - GNU Classpath (recent CVS);
 - Cairo 1.0.2 (via cairo-java and custom CairoGraphics2D);
 - JFreeChart 1.0.0;
- PNG output;

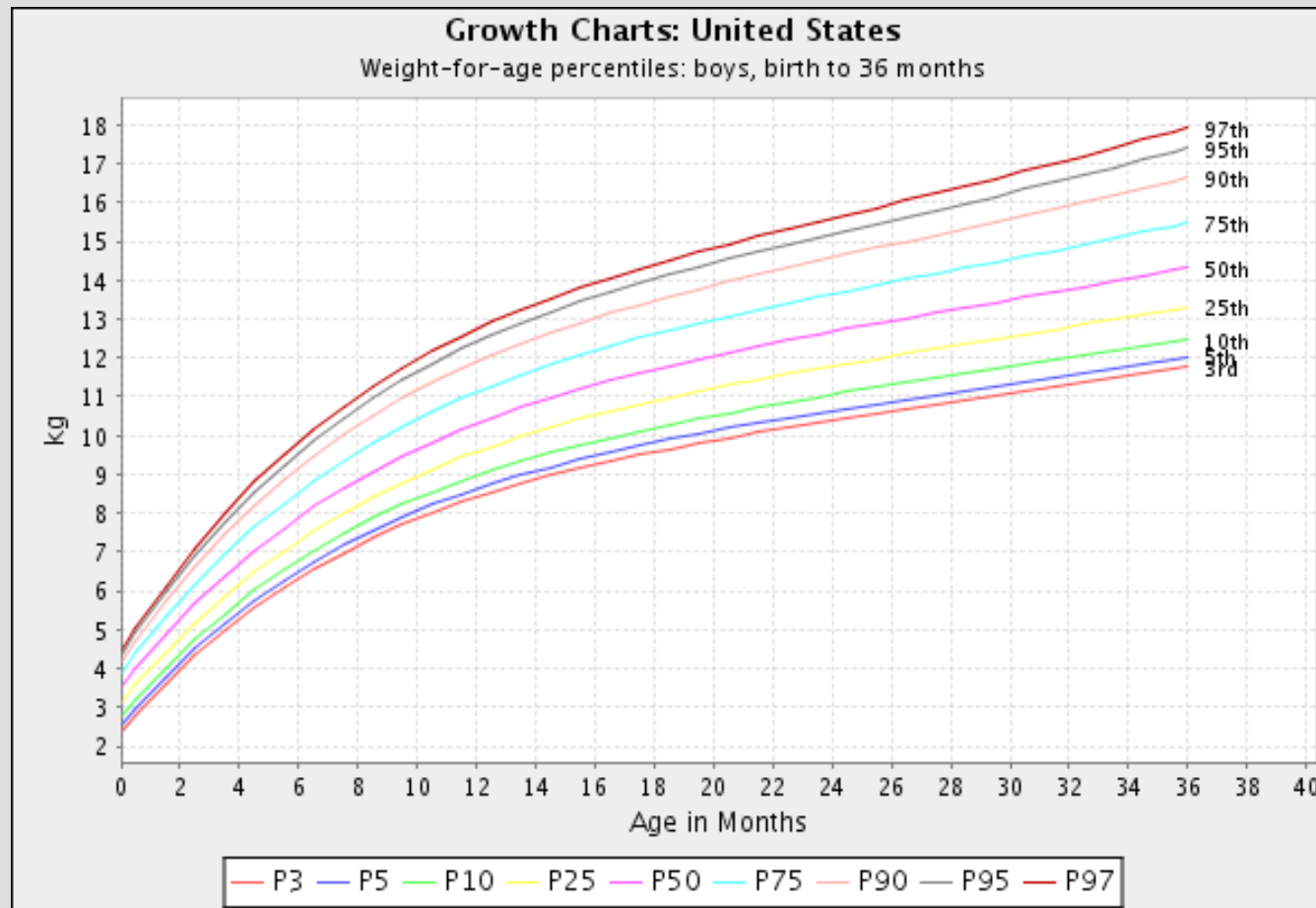
Sample : Pie Chart



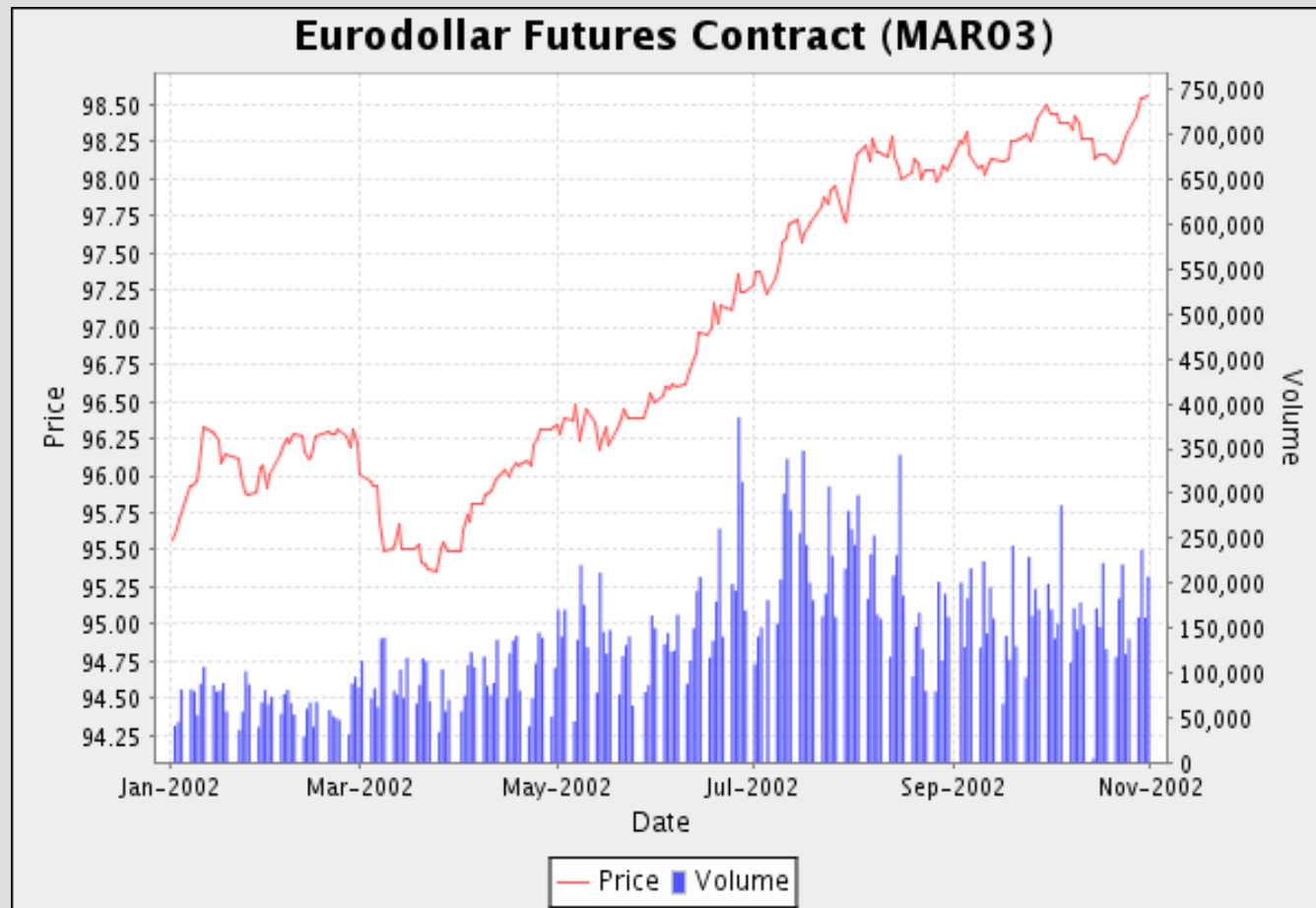
Sample : Bar Chart



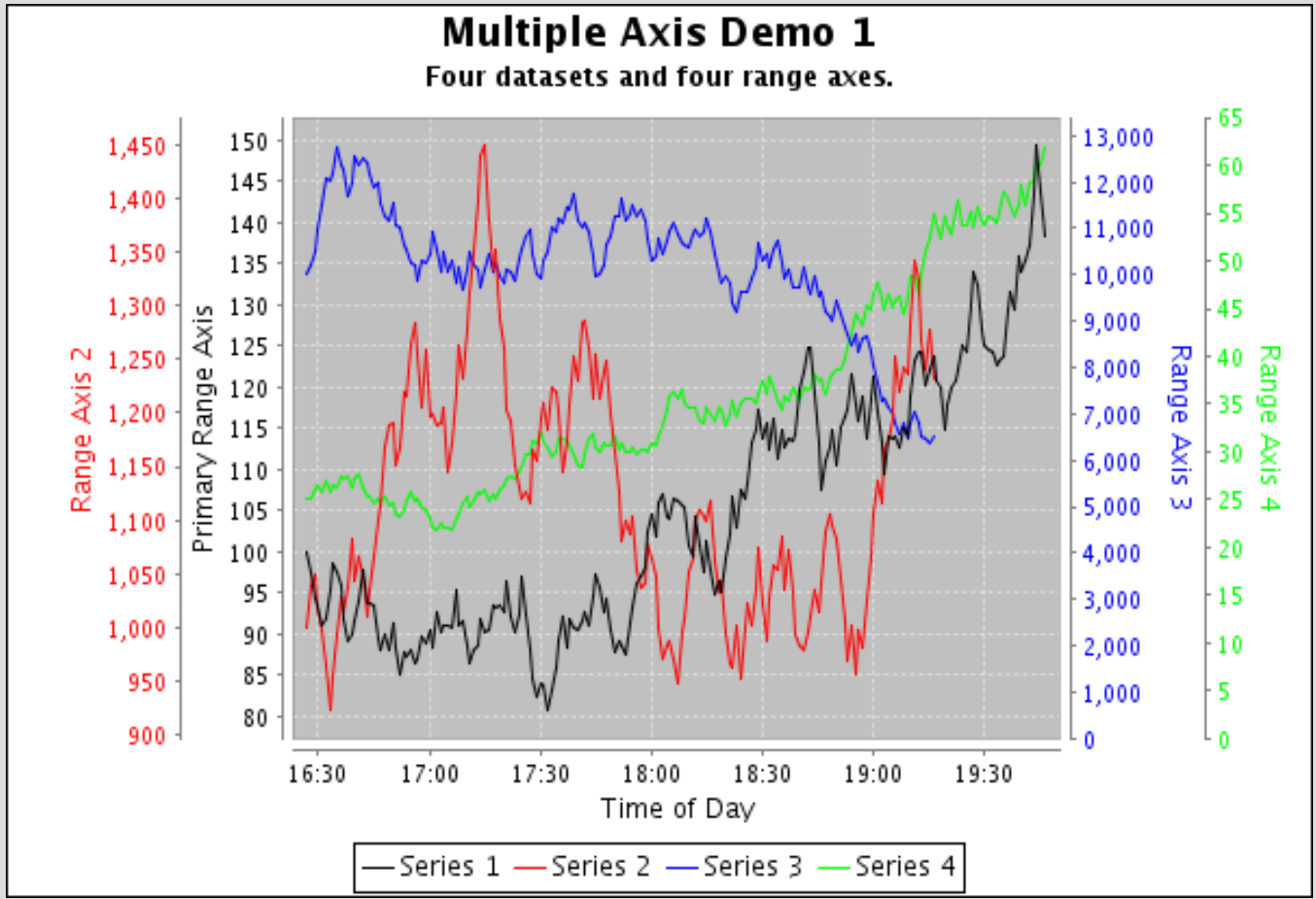
Sample : Line Chart



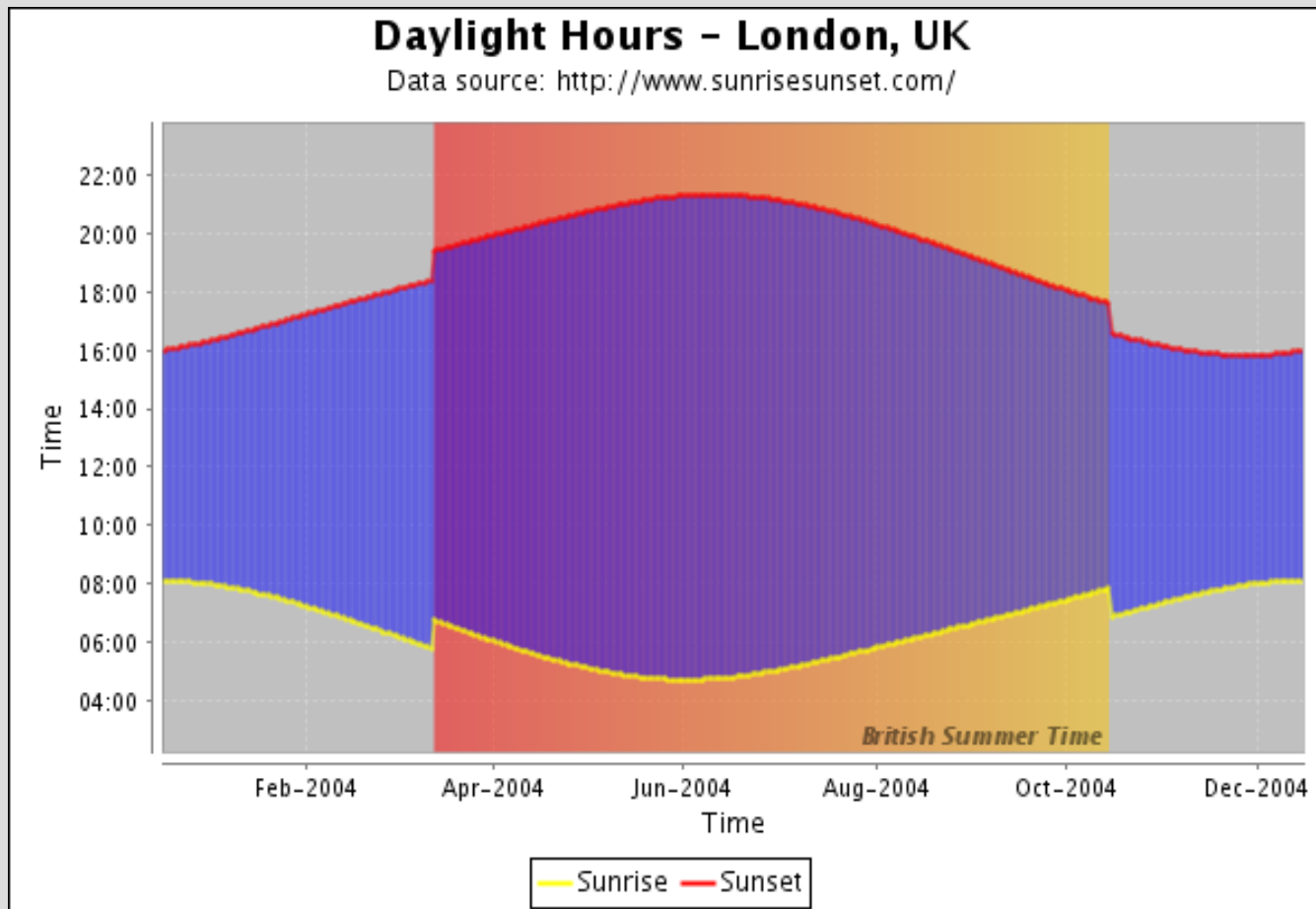
Sample: Financial Chart



Sample : Chart with Multiple Axes



Sample : Something different



IV. A Real World Application

- StatCVS;
- an analysis for Mauve;

StatCVS

- generates CVS logfile summary reports;
 - <http://statcvs.sourceforge.net/>
 - GNU LGPL
 - output to HTML, including charts generated by JFreeChart;

StatCVS for Mauve (1)

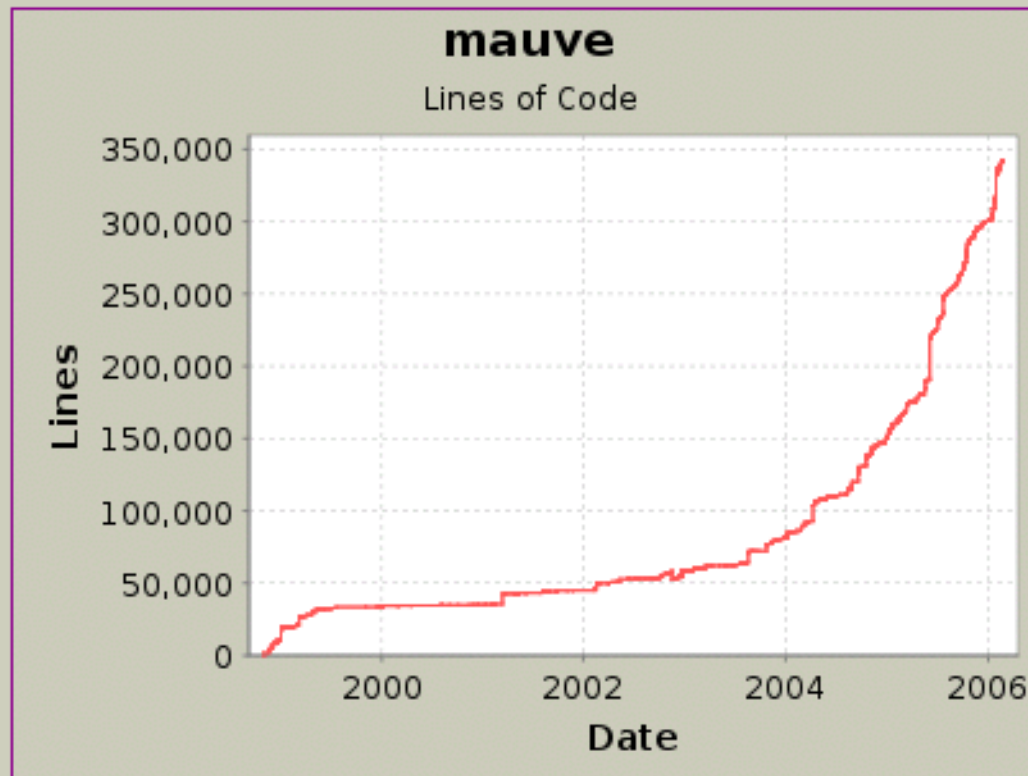
- use Mauve CVS for an example;
- Mauve: a free test suite for the standard Java™ APIs:
 - <http://sources.redhat.com/mauve/>
 - GNU General Public Licence (GPL);
 - can be used to test ANY implementation of the Java APIs, but is primarily developed and used by GNU Classpath hackers;

StatCVS for Mauve (2)

- Use a modified version of StatCVS:
 - <http://www.object-refinery.com/classpath/statcvs.html>
- Summary of changes:
 - update to use JFreeChart 1.0.0;
 - incorporate small modification to use `CairoImage` rather than `java.awt.BufferedImage`;
- Generate a log file:
 - `cvs log > logfile.log`
- Run it...

StatCVS for Mauve (3)

Lines of Code



Total Lines Of Code: 342430 (2006-02-23 03:03)

The Future

- implementation of `Graphics2D` in GNU Classpath, will enable the use of `JFreeChart`:
 - in Swing-based client applications (`ChartPanel` extends `JPanel`);
 - to create and save `BufferedImage` instances;
- extend / test external `Graphics2D` implementations:
 - `CairoGraphics2D`;
 - `EPSGraphics2D` (jibble.org);
 - `PdfGraphics2D` (`iText`);
 - `SWTGraphics2D`?

Questions

- Questions?