1. 基本説明

Java からグラフを実現するソフト・ライブラリは商用も含めて多く存在しますが、本資料で紹介する JFreeChart はオープンソースで、十分に実用に耐えうる完成度を持っています。雑誌、Web サイト等でも紹介されていますが、私が実際に触った印象では、サンプルソースが公開されていない事もあり、シンプルなグラフを出すのは簡単ですが、一歩進んで表現方法、見栄えに凝ろうとすると敷居が高くなると感じました。そのような理由から、本資料では、基本説明＆頻繁に使用されると思われるグラフについてのカスタマイズ方法に重点を置いて説明しようと思います。

1.1. JFreeChart

JFreeChart(http://www.jfree.org/JFreeChart/)は Java から利用可能なオープンソースのグラフ作成のライブラリで、LGPL で配布されています。

＃残念なことにサンプルソースとマニュアルは有料

1.2. インストール

インストールはダウンロードしたアーカイブを展開します。(以下、展開したディレクトリを$JFREE_CHART とします)

以下、ソースのコンパイル、Javadoc の生成は必要に応じて行います。

(a) ソースのコンパイル

$JFREE_CHART¥ant に移動し、以下のコマンドを実行します。

```
ant compile
```

(b) Javadoc の生成

$JFREE_CHART¥ant に移動し、以下のコマンドを実行します。

```
ant javadoc
```

1.3. サンプル

JFreeChart でどのようなチャートが作成できるのか、サンプルを実行してみましょう。

JFreeChart を解凍したディレクトリに移動し、以下のコマンドを実行することでサンプルが起動します。

```
java –jar JFreeChart-0.9.21-demo.jar
```

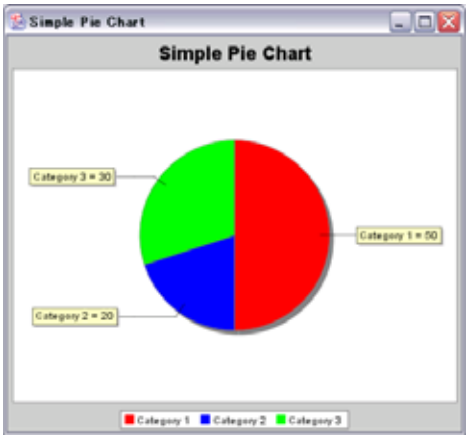
サンプルを見れば、表現力豊かなチャートが作成できることが分かります。

2. チャート作成

JFreeChart でチャートを作成する場合は以下の 3 つの処理を行う必要があります。

1. 出力チャートに合わせたデータセットを作成する。
2. 出力チャートに合わせた JFreeChart オブジェクトを生成する
3. チャートを出力する

では、以下のような簡単な円グラフを作成してみましょう。



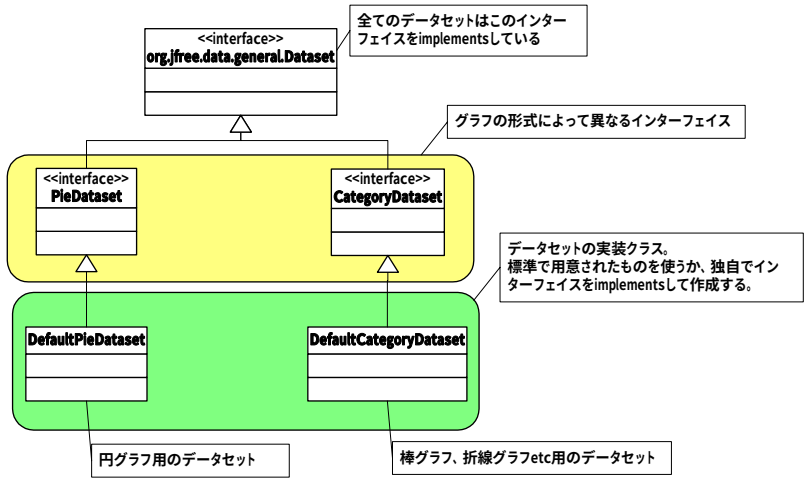
ソースコード

```
package JFreeChart;

import org.JFreeChart.ChartFactory;
import org.JFreeChart.ChartFrame;
import org.JFreeChart.JFreeChart;
import org.jfree.data.general.DefaultPieDataset;

/**
 * 円グラフのサンプル
 *
 * @author A.YOKOI
 */
public class PieChartSample {
    public static void main(String[] args) {

        // ①データセットの作成
        DefaultPieDataset data = new DefaultPieDataset();
        data.setValue("Category 1", 50);
        data.setValue("Category 2", 20);
        data.setValue("Category 3", 30);

        // ②JFreeChartオブジェクトの生成
        JFreeChart chart = ChartFactory.createPieChart(
            "Simple Pie Chart",
            data,
            true,
            true,
            false
        );

        // ③チャートの出力
        ChartFrame frame = new ChartFrame("Simple Pie Chart", chart);
        frame.pack();
        frame.setVisible(true);
    }
}
```

これぐらいの簡単なグラフであれば、ソースコードを見れば処理内容がすぐに把握できるかと思います。
複雑なグラフの場合でも基本的な流れはこの 3 つのステップは変わりません。
次章以降では、各ステップの詳細について見ていきます。

2.1. データセットの作成

データセットはグラフに表示する値の集合ですので、出力するグラフの形式によって値の保持の仕方も変わってきます。そのため出力形式に応じて、使用するデータセット(独自で作成する場合は implements するインターフェイス)を選択する必要があります。JDBC～Dataset を使用すると JDBC で取得したデータをそのままデータセットとして使用することが出来ます。



グラフ種別	利用可能データセット	備考
円グラフ(Pie Chart)	PieDataset JDBCPieDataset	
棒グラフ(Bar Chart)	CategoryDataset IntervalXYDataset JDBCCategoryDataset JDBCXYDataset	
折れ線グラフ(Line Chart)	CategoryDataset XYDataset JDBCCategoryDataset JDBCXYDataset	
時系列グラフ(Time Series Chart)	XYDataset JDBCXYDataset	

2.2. JFreeChart オブジェクトの生成

出力するグラフの形式に合わせて ChartFactory から JFreeChart オブジェクトを生成します。

例えば、ChartFactory .createBarChart()では棒グラフ用、ChartFactory .createBarChart3D()では棒グラフ(3D 表示)用の JFreeChart オブジェクトが取得できます。

また、JFreeChart オブジェクトを操作することで、表示形式のカスタマイズが可能です。カスタマイズ方法の説明を『3 グラフのカスタマイズ』に記述しましたので、詳細はそちらを参照してください。

2.3. グラフの出力

利用目的によって様々な形式で出力することが出来ます。

(1) org.JFreeChart.ChartPanel

サンプルソースは全てこの出力形式を使用しています。

org.JFreeChart.ChartPanel は JPanel を継承しているため、そのまま Java アプリケーションに組み込むことが出来ます。

(2) JPEG,PNG

JPEG、PNG のイメージファイルとして出力

- ChartUtilities.saveChartAsJPEG()
- ChartUtilities.saveChartAsPNG()

(3) PDF

iText(http://www.lowagie.com/iText)と組み合わせることで PDF に出力することが出来ます。

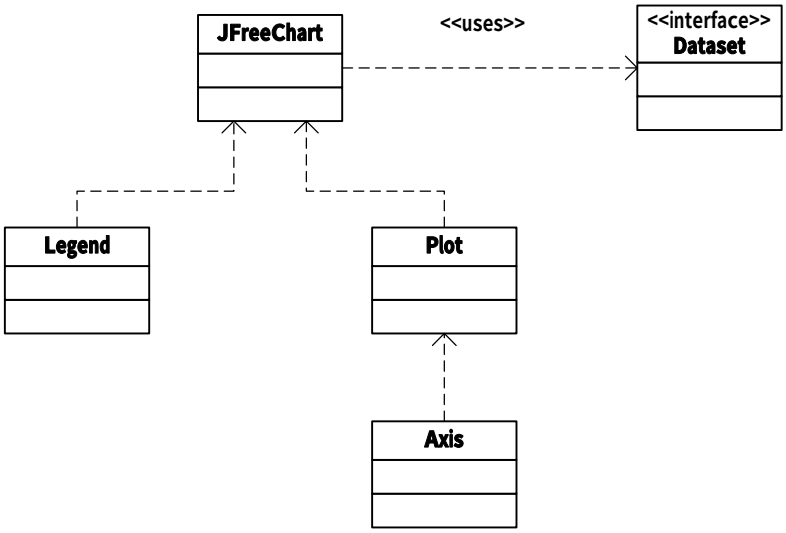
(4) SVG (Scalable Vector Graphics)

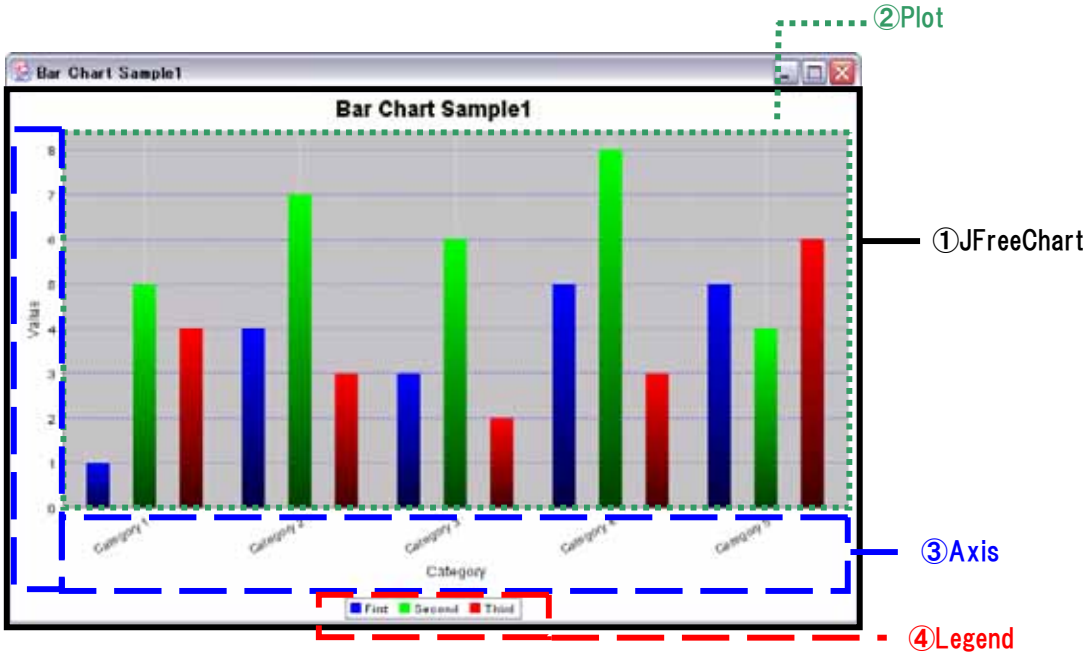
Batic(http://xml.apache.org/batik)と組み合わせることで、SVG フォーマットで出力することが出来ます。

3. グラフのカスタマイズ

JFreeChart ではグラフの出力関するさまざまなカスタマイズが可能です。

カスタマイズする内容によって以下のように操作するオブジェクトが異なります。





以降では代表的なカスタマイズ属性を紹介します。

(1) org.JFreeChart.JFreeChart

グラフ全体にかかわるカスタマイズは主に JFreeChart のオブジェクトに対して行います。

- グラフの境界線
  - chart.setBorderPaint()
  - chart.setBorderStroke()
- タイトルの表示位置
  - chart.getTitle().setPosition();
- 背景色
  - chart.setBackgroundPaint();
- 背景イメージ
  - chart.setBackgroundImage() // イメージサイズの自動調節
  - chart.setBackgroundImageAlignment();// イメージサイズの自動調節なし

(2) org.JFreeChart.plot.Plot

グラフの描画領域に関する設定は主にPlotのオブジェクトに対して行います。Plotオブジェクトは以下のように取得します。

Plot plot = chart. getPlot ();

- 背景色
  - plot.setBackgroundPaint();
- 背景イメージ
  - plot.setBackgroundImage();// イメージサイズの自動調節
  - plot.setBackgroundImageAlignment();// イメージサイズの自動調節なし

(3) org.JFreeChart.axis.Axis

グラフの軸に関する設定は主に Axis に対して行います。

Axis には CategoryAxis と ValueAxis が存在し、それぞれ横軸、縦軸(設定次第では逆)に対応しています。

CategoryAxisは以下のように取得します。

```
CategoryPlot plot = chart.getCategoryPlot();
CategoryAxis domainAxis = plot.getDomainAxis();
```

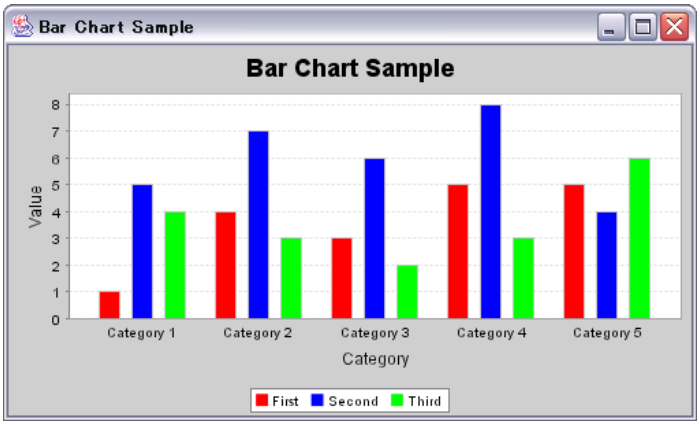
- フォント
  - domainAxis.setLabelFont();
- 描画色
  - setLabelPaint();

3.1. 棒グラフのカスタマイズ

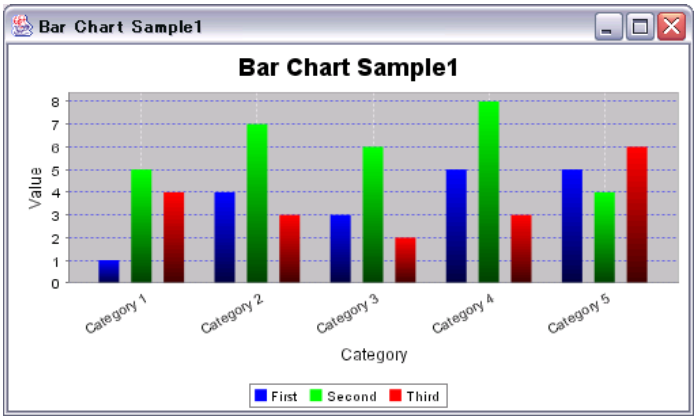
本章では、カスタマイズの例として棒グラフの表示方法を変更してみます。

以下の例では、背景色、描画エリアのプロット、バーの表示色、マージン設定などを変更しています。各カスタマイズの方法はサンプルのソースを参考にしてください。

(1) カスタマイズ前(BarChartSample)



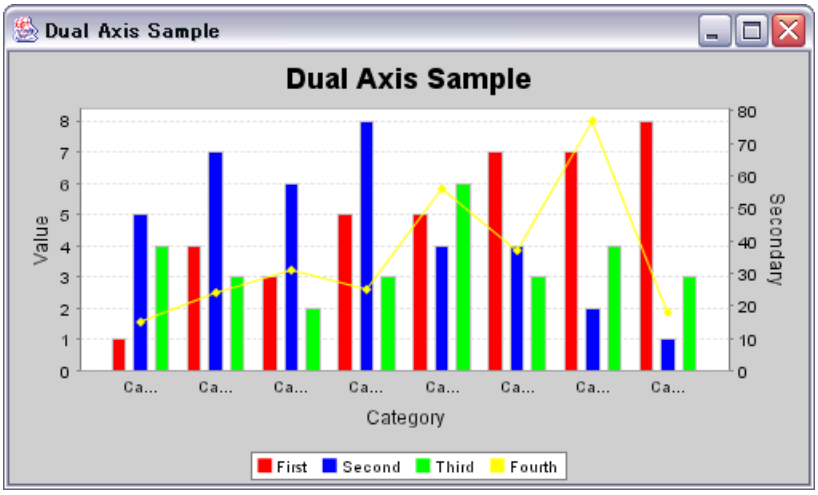
(2) カスタマイズ後(BarChartSample1)



3.2. 複数データセット、複数軸のグラフ

通常 JFreeChart で作成したグラフは、1 つの軸、1 つのデータセットを持ちますが、それらを複数持たせることで以下のような複数軸、複数データセットを持つグラフを生成することが出来ます

基本的な流れは、今まで見てきたように通常のグラフを作成後、軸とデータソースを追加する処理を行います。詳細はサンプルソース(DualAxisSample)を参照してください。



4. 最後に

グラフの基本的な作成・カスタマイズ方法について説明しましたが、いかがでしたでしょうか？ 目に見えて出力が確認できるので、本資料で紹介した以外の使用方法についても楽しみながら学べるのではないかと思います。

時間と要望があれば今回サンプルを作成した以外のグラフについても記述を追加していこうと思います。

記述内容について、お気づきの点、質問等ありましたら下記までご連絡ください。

開発部
横井 朗
yokoi@bbreak.co.jp