



[developerWorks 中国网站 : Java : 所有的文章](#)

developerWorks

使用 JFreeChart 来创建基于 web 的图表

刘冬 (winter.lau@163.com)

珠海市创我科技发展有限公司软件工程师

2003 年 9 月

WWW 的发展使得基于因特网的应用程序不再局限于静态或者简单的动态内容提供。传统的一些以软件包形式发布应用程序例如报表系统等都在逐渐搬到因特网上。但是这两者之间有着天壤之别，虽然对于数据获取、业务处理等方面基本类似，但是最大的差别在于用户界面。为了能在 web 浏览器上显示要求用户界面使用 HTML 以及图片的方式来展现数据，而传统的一些利用操作系统本身的控件来开发的用户界面无法适应琳琅满目的客户端，因此在这里也变得无能为力。回到本文的题目上来，为了创建一个可以在 web 浏览器上查看到图表一般有两种做法：第一种就是使用 applet 利用 java 本身对图形的支持来显示一个图表；第二种就是直接在 web 服务器端生成好图表图片文件后发送给浏览器。第一种方式显然对于客户端要求太高，随着现在主流浏览器放弃对 JAVA 的支持后，这种方式只适合一些局域网的应用，而对于因特网的环境就显得不太适合。因此我们下面将介绍一个 JAVA 的图表引擎 JFreeChart 用来产生基于 WEB 的图表。

一、JFreeChart 项目简介

JFreeChart 是开放源代码站点 SourceForge.net 上的一个 JAVA 项目，它主要用来各种各样的图表，这些图表包括：饼图、柱状图(普通柱状图以及堆栈柱状图)、线图、区域图、分布图、混合图、甘特图以及一些仪表盘等等。这些不同式样的图表基本上可以满足目前的要求。为了减少篇幅本文主要介绍前面三种类型的图表，读者可以触类旁通去开发其他样式的图表。下面几个是 JFreeChart 产生的这三种类型图表的结果：

内容：

- [一、JFreeChart 项目简介](#)
- [二、解读 JFreeChart 的源码结构](#)
- [三、使用 JFreeChart 生成各种样式的图表](#)
- [四、将生成的图表移到浏览器上](#)

高级主题

参考资料

关于作者

Java 专区中还有：

- [教学](#)
- [工具与产品](#)
- [代码与组件](#)
- [所有文章](#)
- [实用技巧](#)

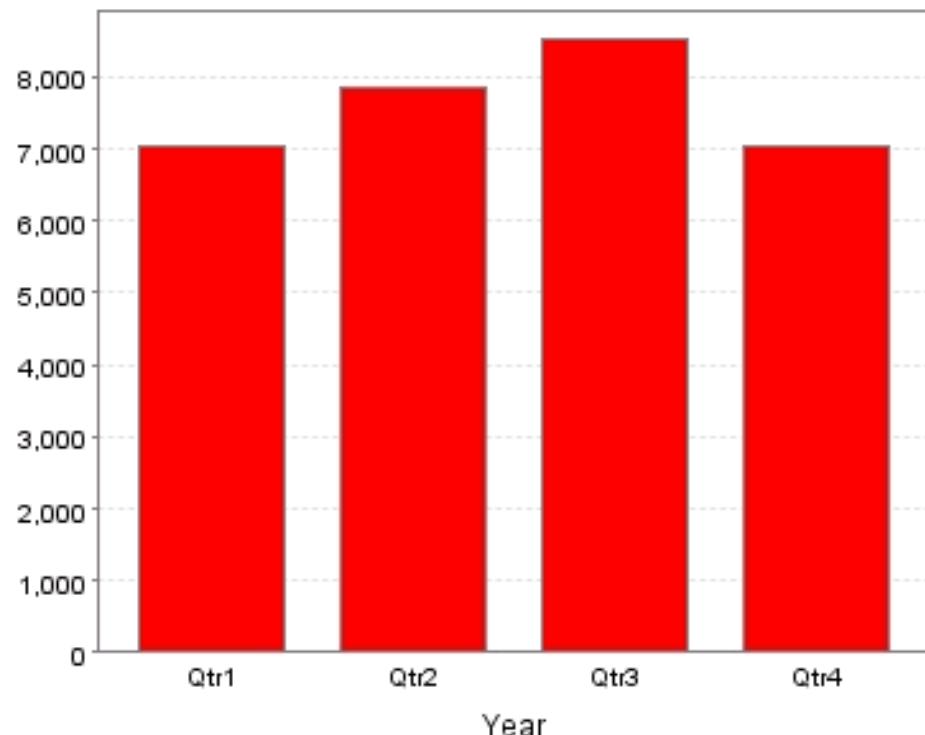


图1

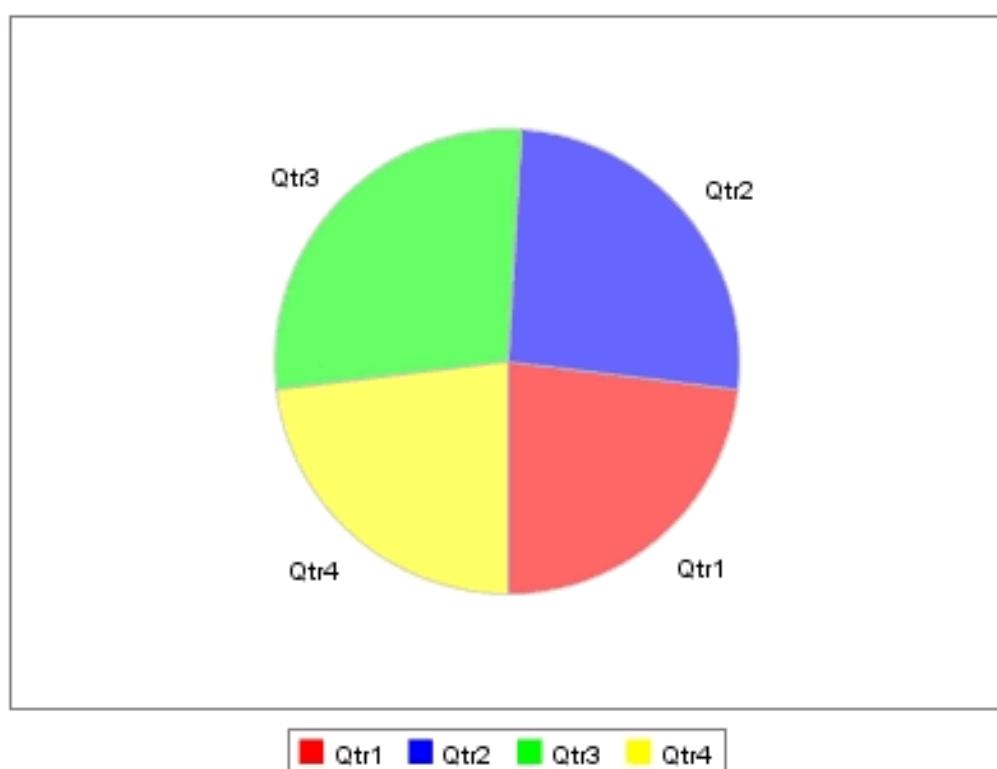


图2

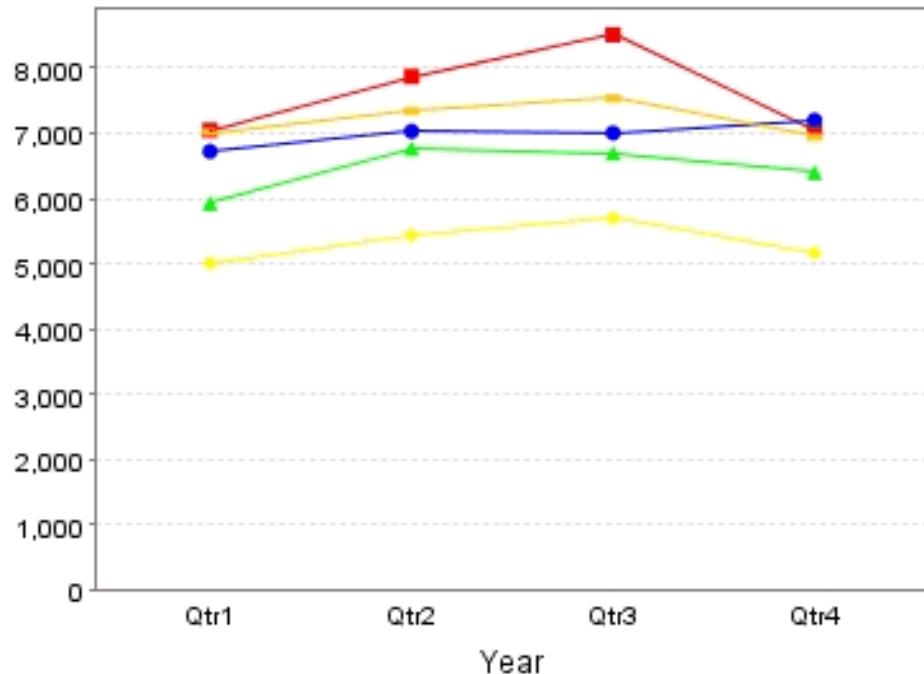


图3

上面的三个图都是表示四个季度的某个产品的销量信息。在继续下面小节之前必须先准备好开发环境，因为是基于WEB浏览器的图表展现，因此需要一个Servlet引擎或者是J2EE应用服务器（例如WebSphere，Tomcat等）。WEB环境的搭建就不赘述了，读者根据喜好自行安装。JFreeChart引擎本身需要到SourceForge.net上下载，地址如下：

JFreeChart主页：<http://www.jfree.org/jfreechart/index.html>
JFreeChart下载页面：<http://sourceforge.net/projects/jfreechart/>

下载的时候需要注意的是必须下载两个文件：JFreeChart以及Jcommon。目前最新配套版本是：JFreeChart 0.9.11
Jcommon 0.8.6

这里有点笔者在开发中遇见的问题需要注意的是：在使用Eclipse开发的时候会报一个莫名其妙的错误，错误可能指向某个类文件的第一行。遇到这样的问题一般是因为没有把Jcommon的jar包设置到项目的类路径中的缘故。具体的原因不祥。

二、解读JFreeChart的源码结构

在开始使用JFreeChart之前我们有必要先大概了解一下JFreeChart本身的结构以及它所带一些例子程序，这样有助于我们下一步自行开发。下载JFreeChart包后已经带有非常丰富的例子，因为JFreeChart这个项目本身的使用文档非常少，因此学习它最好的办法就是学习它所带的例子源码。在包org.jfree.chart.demo中有几十个文件用于展示JFreeChart所能支持的所有图表的结果。如果你的JDK是比较新的情况下可能在运行这些例子时会有问题，现象如下：

```

java.lang.UnsatisfiedLinkError: initDDraw
at sun.awt.windows.Win32OffScreenSurfaceData.initDDraw(Native Method)
at sun.awt.windows.Win32OffScreenSurfaceData.<clinit>(Win32OffScreenSurfaceData.java:141)
at sun.awt.Win32GraphicsDevice.<clinit>(Win32GraphicsDevice.java:58)
at sun.awt.Win32GraphicsEnvironment.makeScreenDevice(Win32GraphicsEnvironment.java:168)
at sun.java2d.SunGraphicsEnvironment.getScreenDevices(SunGraphicsEnvironment.java:240)
at sun.awt.Win32GraphicsEnvironment.getDefaultScreenDevice(Win32GraphicsEnvironment.java:61)
at java.awt.Window.init(Window.java:224)
at java.awt.Window.<init>(Window.java:268)
at java.awt.Frame.<init>(Frame.java:398)
at javax.swing.JFrame.<init>(JFrame.java:198)
at org.jfree.chart.demo.JFreeChartDemo.<init>(JFreeChartDemo.java:148)
at org.jfree.chart.demo.JFreeChartDemo.main(JFreeChartDemo.java:285)
Exception in thread "main"

```

这个错误是由于新版的 Swing 大量的使用了微软的 DirectDraw 的技术来提高画图的性能，而可能你的显卡在这时候会跟你闹点情绪或者显卡本身并不支持这样的一个技术。难道就没有办法了吗？要解决这个问题也非常简单，我们可以屏蔽掉 DirectDraw，不让 Swing 使用该技术就可以了。在运行这些代码时给虚拟机指定参数-Dsun.java2d.noddraw 即可。

这时可能你又该纳闷了，不说基于 Web 的图表嘛，怎么又扯到 Swing 上了？这是因为为了使开发者容易上手，无需配置任何运行环境，所以这些例子都是基于 GUI 方式的用于展现给开发者如果生成一个图表，我们要学习的也就是如何利用这个引擎生成图表而不是怎么来显示一个图表。当我们把生成的图表对象 Export 到一个图像文件即可在 Web 上发布。

下面我们来介绍 JFreeChart 中几个核心的对象类：

类名	类的作用以及简单描述
JFreeChart	图表对象，任何类型的图表的最终表现形式都是在该对象进行一些属性的定制。JFreeChart 引擎本身提供了一个工厂类用于创建不同类型的图表对象
XXXXXDataset	数据集对象，用于提供显示图表所用的数据。根据不同类型的图表对应着很多类型的数据集对象类
XXXXXPlot	图表区域对象，基本上这个对象决定着什么样式的图表，创建该对象的时候需要 Axis、Renderer 以及数据集对象的支持
XXXXXAxis	用于处理图表的两个轴：纵轴和横轴
XXXXXRenderer	负责如何显示一个图表对象
XXXXXURLGenerator	用于生成 Web 图表中每个项目的鼠标点击链接
XXXXXToolTipGenerator	用于生成图象的帮助提示，不同类型图表对应不同类型的工具提示类

基本上我认为 JFreeChart 项目本身的类结构的设计并不是很好，首先在创建图表的时候用到了大量的工厂方法，这样做虽然可以简化创建图表对象的代码，但是对项目本身或者开发人员来讲自行扩展一种新的图表都仍然是一件很麻烦的事情；其次除图表对象本身外其余的类过于复杂，使用者必须去了解每个类型的图表对象应该对应哪些 Axis、Plot、Renderer 类，并且必须非常熟悉这些类的构造函数中每个参数的具体含义。这些问题都大大困扰很多初学者。不过，虽然存在很多问题，但是 JFreeChart 本身仍不失为一个非常优秀的图表引擎，况且项目本身也在逐渐的发展中。

在非常简略的介绍了 JFreeChart 本身的代码结构后，下面我们开始动手试验几个常用的图表并把他们放到 web 上。

三、使用 JFreeChart 生成各种样式的图表

限于篇幅的问题我们在这里只实现两种常用的图表，其他类型图表读者可以触类旁通。我们先给出柱状图的实现，饼图的实

现再来跟柱状图进行比较。

1 柱状图

```

package lius.chart.demo;

import java.io.*;

import org.jfree.data.*;
import org.jfree.chart.*;
import org.jfree.chart.plot.*;
/**
 * 该类用于演示最简单的柱状图生成
 * @author Winter Lau
 */
public class BarChartDemo {

    public static void main(String[] args) throws IOException{

        CategoryDataset dataset = getDataSet2();
        JFreeChart chart = ChartFactory.createBarChart3D(
            "水果产量图", // 图表标题
            "水果", // 目录轴的显示标签
            "产量", // 数值轴的显示标签
            dataset, // 数据集
            PlotOrientation.VERTICAL, // 图表方向：水平、垂直
            true, // 是否显示图例(对于简单的柱状图必须是false)
            false, // 是否生成工具
            false // 是否生成URL链接
        );

        FileOutputStream fos_jpg = null;
        try {
            fos_jpg = new FileOutputStream("D:\\fruit.jpg");
            ChartUtilities.writeChartAsJPEG(fos_jpg,100,chart,400,300,null);
        } finally {
            try {
                fos_jpg.close();
            } catch (Exception e) {}
        }
    }
}

/**
 * 获取一个演示用的简单数据集对象
 * @return
 */
private static CategoryDataset getDataSet() {
    DefaultCategoryDataset dataset = new DefaultCategoryDataset();
    dataset.addValue(100, null, "苹果");
    dataset.addValue(200, null, "梨子");
    dataset.addValue(300, null, "葡萄");
    dataset.addValue(400, null, "香蕉");
    dataset.addValue(500, null, "荔枝");
    return dataset;
}

/**
 * 获取一个演示用的组合数据集对象
 * @return
 */
private static CategoryDataset getDataSet2() {
    DefaultCategoryDataset dataset = new DefaultCategoryDataset();
    dataset.addValue(100, "北京", "苹果");
    dataset.addValue(100, "上海", "苹果");
    dataset.addValue(100, "广州", "苹果");
    dataset.addValue(200, "北京", "梨子");
    dataset.addValue(200, "上海", "梨子");
    dataset.addValue(200, "广州", "梨子");
    dataset.addValue(300, "北京", "葡萄");
    dataset.addValue(300, "上海", "葡萄");
    dataset.addValue(300, "广州", "葡萄");
    dataset.addValue(400, "北京", "香蕉");
}

```

```
dataset.addValue(400, "上海", "香蕉");
dataset.addValue(400, "广州", "香蕉");
dataset.addValue(500, "北京", "荔枝");
dataset.addValue(500, "上海", "荔枝");
dataset.addValue(500, "广州", "荔枝");
return dataset;
}
}
```

程序运行结束后生成的图片文件效果如下图所示：

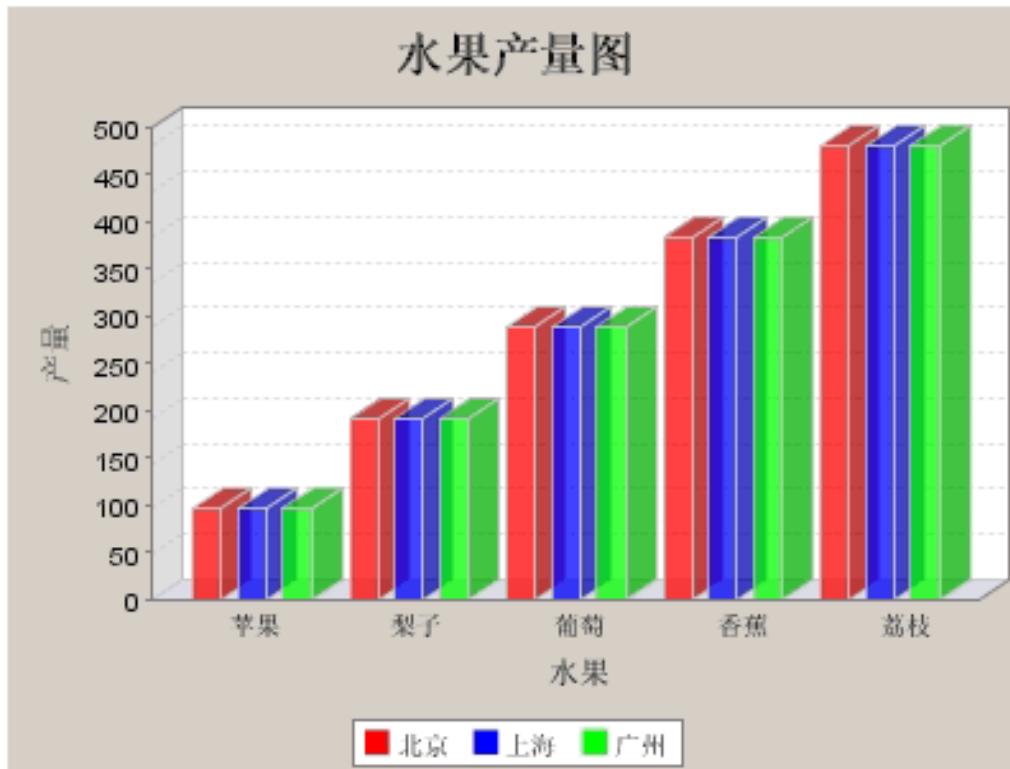


图4

如果是使用简单数据即使用 `getDataSet` 方法获取数据集时产生的图片文件如下：

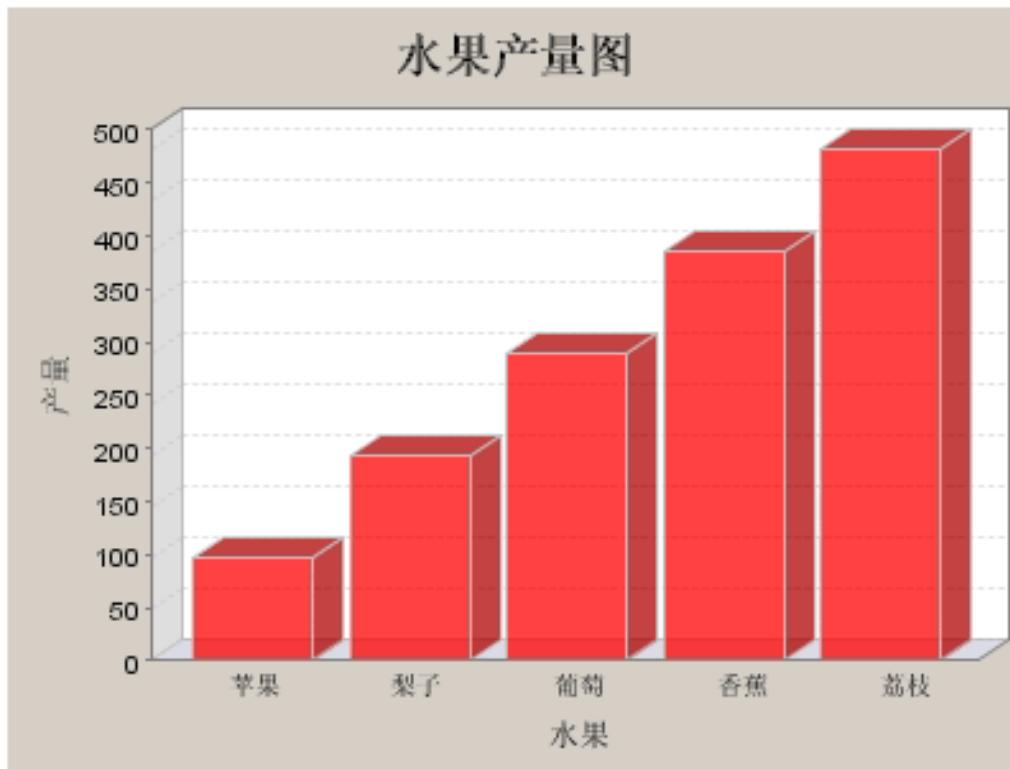


图5

2 饼图

对于饼图而言，数据集的获取用的不是同一个数据集类，另外饼图不支持同一个类别的项目中还有子项目这样的数据。我们只给出创建饼图的代码，至于写图表到一个文件则与柱状图一致，无需重复。

```

package lius.chart.demo;

import java.io.*;

import org.jfree.data.*;
import org.jfree.chart.*;
/***
 * 用于演示饼图的生成
 * @author Winter Lau
 */
public class PieChartDemo {

    public static void main(String[] args) throws IOException{
        DefaultPieDataset data = getDataSet();
        JFreeChart chart = ChartFactory.createPie3DChart("水果产量图", // 图表标题
        data,
        true, // 是否显示图例
        false,
        false
        );
        //写图表对象到文件，参照柱状图生成源码
    }
    /**
     * 获取一个演示用的简单数据集对象
     * @return
     */
    private static DefaultPieDataset getDataSet() {
        DefaultPieDataset dataset = new DefaultPieDataset();
        dataset.setValue("苹果",100);
        dataset.setValue("梨子",200);
    }
}

```

```
dataset.setValue("葡萄",300);
dataset.setValue("香蕉",400);
dataset.setValue("荔枝",500);
return dataset;
}
```

生成的饼图文件效果如下：



图6

四、将生成的图表移到浏览器上

为了将生成的图表直接传给客户端浏览器，只需要将前面两个例子中的文件流换成是通过HttpServletResponse对象获取到的输出流，详细代码清单如下：

```

package lius.chart.demo;

import java.io.IOException;
import javax.servlet.*;
import javax.servlet.http.HttpServlet;

import org.jfree.data.*;
import org.jfree.chart.*;
/***
 * 演示通过servlet直接输出图表
 * @author Winter Lau
 */
public class ChartDemoServlet extends HttpServlet {

    public void service(ServletRequest req, ServletResponse res)
        throws ServletException, IOException
    {
        res.setContentType("image/jpeg");
        DefaultPieDataset data = getDataSet();
        JFreeChart chart = ChartFactory.createPie3DChart("水果产量图",
            data,
            true,
            false,
            false
        );
        ChartUtilities.writeChartAsJPEG(res.getOutputStream(),
            100,chart,400,300,null);
    }
    /**
     * 获取一个演示用的简单数据集对象
     * @return
     */
    private static DefaultPieDataset getDataSet() {
        DefaultPieDataset dataset = new DefaultPieDataset();
        dataset.setValue("苹果",100);
        dataset.setValue("梨子",200);
        dataset.setValue("葡萄",300);
        dataset.setValue("香蕉",400);
        dataset.setValue("荔枝",500);
        return dataset;
    }
}

```

高级主题

很多情况我们不仅仅要求可以在浏览器上显示一个图表，我们更需要客户可以直接在图表上做一下交互的操作，例如获取信息提示，点击图表某个部分进行更详细信息的展示等等。例如前面生成的简单柱状图，用户需要在看到柱状图后点击某种水果例如是苹果即可看到各个地区苹果产量的情况。为此就要求该图形具有交互操作的功能。在HTML中为了让一个图像具有可交互的功能就必须给该图像定义一个Map对象。下表节选一段具有该功能的HTML代码

```

<MAP NAME="chartMap">
<AREA SHAPE="RECT" COORDS="81,15,126,254" href="?series=0&category=100" title="100 = 7,048"
      onclick="javascript:clickChart('100');return false;">
<AREA SHAPE="RECT" COORDS="143,27,188,255" href="?series=0&category=200" title="200 = 6,721"
      onclick="javascript: clickChart ('200');return false;">
<AREA SHAPE="RECT" COORDS="205,54,250,255" href="?series=0&category=300" title="300 = 5,929"
      onclick="javascript: clickChart ('300');return false;">
<AREA SHAPE="RECT" COORDS="267,85,312,255" href="?series=0&category=400" title="400 = 5,005"
      onclick="javascript: clickChart ('400');return false;">
<AREA SHAPE="RECT" COORDS="329,17,374,255" href="?series=0&category=Diet" title="Diet = 7,017" onclick="javascript:
      clickChart ('Diet');return false;">
</MAP>

```

由此就产生了一个问题：如果根据一个图像来生成对应的MAP对象。我们回头看看刚才的代码，在创建一个图表对象时候有两个参数，我们举柱状图的例子来讲这两个参数就是ChartFactory.createBarChart3D方法中的最后两个参数，这两个参数的类型都是布尔值。这两个参数意思分别是：是否创建工具提示(tooltip)以及是否生成URL。这两个参数分别对应着MAP中一个AREA的title属性以及href属性。

可是我想知道的是怎么来产生这个MAP啊！哈哈，不要着急，JFreeChart已经帮我们做好生成MAP对象的功能。为了生成MAP对象就要引入另外一个对象：ChartRenderingInfo。因为JFreeChart没有直接的方法利用一个图表对象直接生成MAP数据，它需要一个中间对象来过渡，这个对象就是ChartRenderingInfo。下图是生成MAP数据的流程图：

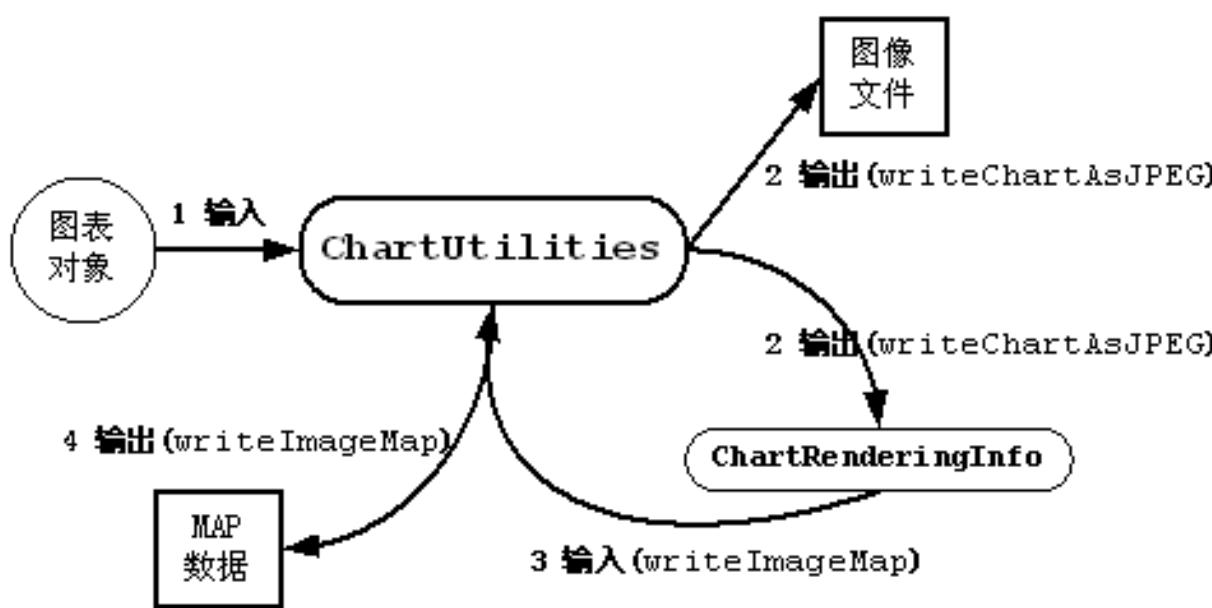


图7

如上图所示，ChartUtilities类是整个流程的核心，它周围的对象都是一些例如数据对象或者是文件等。这个流程简单描述如下：首先创建一个ChartRenderingInfo对象并在调用ChartUtilities的writeChartAsJPEG时作为最后一个参数传递进去。调用该方法结束后将产生一个图像文件以及一个填充好MAP数据的ChartRenderingInfo对象，有了这个对象我们还是没有办法获取具体的MAP数据，我们还必须借助于ChartUtilities的writeImageMap方法来将ChartRenderingInfo对象读取出来，获取MAP数据的代码片断如下：

```

PrintWriter w = null;
FileOutputStream fos_jpg = null;
FileOutputStream fos_cri = null;
try{
    //根据不同类型的图表使用不同类，以下是针对饼图的操作
    PiePlot plot = (PiePlot) chart.getPlot();
    plot.setURLGenerator(new StandardPieURLGenerator(url));
    //设置工具提示
    plot.setToolTipGenerator(new StandardPieToolTipGenerator());
    fos_jpg = new FileOutputStream("d:\\fruit.jpg");
    ChartUtilities.writeChartAsJPEG(
        fos_jpg,
        100,
        chart,
        400,
        300,
        info);
    fos_cri = new FileOutputStream("d:\\fruit.map");
    w = new PrintWriter(fos_cri);
}
  
```

```
ChartUtilities.writeImageMap(w, __mapname__, info);
w.flush();
}finally{
    try{
        w.close();
    }catch(Exception e){}
    try{
        fos_cri.close();
    }catch(Exception e){}
    try{
        fos_jpg.close();
    }catch(Exception e){}
}
```

打开文件 D:\fruit.map , 文件的内容就是要写到页面上的MAP数据。把生成的图像文件以及MAP数据文件写到页面上即可完成热点图表的功能。至于怎么结合两者之间的关系例如图像的useMap属性值必须与MAP对象的名称结合起来，必须根据实际的应用情况进行相应的处理。笔者建议把二者通过标签库封装起来，图像文件的名称以及MAP对象的名称由标签库统一进行控制，这样可以保证二者的一致性。

参考资料

[JFreeChart图表引擎首页](#)
[Cewolf自定义图表标签库](#)

关于作者

刘冬，他已经不再叫凤梨罐头了。珠海市创我科技发展有限公司软件工程师，主要从事在J2EE平台上实现商业智能（BI）方面的开发。
联系电话：0756-3377435-383
电子邮件：winter.lau@163.com

[到页首](#)

您对这篇文章的看法如何？

真棒！(5)

好材料 (4)

一般；尚可 (3)

需提高 (2)

太差！ (1)

建议？

(c) Copyright IBM Corp. 2001, (c) Copyright IBM China 2001, All Right Reserved

[关于 IBM](#) | [隐私条约](#) | [使用条款](#) | [联系 IBM](#)

JAVA>>开源综合>>使用JFreeChart创建图象 [讨论论坛]

使用JFreeChart创建图象

一：jfreechart介绍

jfreechart是一个免费创建图片的java工具.可以创建如下图形：

饼图(pie charts;)

曲线图(line charts)

柱状图(horizontal/vertical bar charts)

甘特图(Gantt charts)

XY plots and scatter plots

time series, high/low/open/close charts and candle stick charts

combination charts

Pareto charts

bubble charts

wind plots, meter charts and symbol charts

从以下地址可以看到jfreechart可以创建的图形类型:

<http://www.jfree.org/jfreechart/samples.html>

sourceforge有一个基于jfreechart的项目Cewolf可以很方便的在jsp/servlet中创建图片。jfreechart目前 (2003-05-08)版本为0.98 希望得到详细的信息或下载jfreechart请访问如下站点:

<http://www.jfree.org/jfreechart/>

二：特别说明:

jfreechart是一个开源项目，但是文档是需要40美金去购买的。还有一个很重要的问题,jfreechart如果使用中文，他使用的默认字体显示出来的中文会很模糊，你可能需要修改源代码。下面我就举几个简单的例子说明一下如何使用jfreechart创建图片在开发中有可能会导入以下的类

```

import com.jrefinery.chart.ChartFactory;
import com.jrefinery.chart.ChartUtilities;
import com.jrefinery.chart.JFreeChart;
import com.jrefinery.chart.TextTitle;
import com.jrefinery.chart.axis.NumberAxis;
import com.jrefinery.chart.plot.CategoryPlot;
import com.jrefinery.chart.plot.PiePlot;
import com.jrefinery.data.Day;
import com.jrefinery.data.DefaultCategoryDataset;
import com.jrefinery.data.DefaultPieDataset;
import com.jrefinery.data.TimeSeries;
import com.jrefinery.data.TimeSeriesCollection;
import com.jrefinery.data.TimeSeriesDataPair;

```

在0.98以后包由com.jrefinery.*改变为:org.jfree

三：创建饼图

```

//图片标题
String title = "空调2002年市场占有率";
//设定数据源
DefaultPieDataset piedata = new DefaultPieDataset();
//第一个参数为名称，第二个参数是double数
piedata.setValue("联想", 27.3);
piedata.setValue("长城", 12.2);
piedata.setValue("海尔", 5.5);
piedata.setValue("美的", 17.1);
piedata.setValue("松下", 9.0);
piedata.setValue("科龙", 19.0);
//创建JFreeChart，都使用ChartFactory来创建JFreeChart,很标准的工厂设计模式
JFreeChart chart =
ChartFactory.createPieChart(title, piedata, true, true, true);
//设定图片标题
chart.setTitle(new TextTitle(title, new Font("隶书", Font.ITALIC, 15)));
//chart.addSubtitle(new TextTitle("2002财年分析", new Font("隶书", Font.ITALIC, 12)));
//设定背景
chart.setBackgroundPaint(Color.white);
//chart.s
//饼图使用一个PiePlot
PiePlot pie = (PiePlot)chart.getPlot();
//pie.setSectionLabelType(PiePlot.NAME_AND_PERCENT_LABELS);
pie.setSectionLabelType(PiePlot.NAME_AND_VALUE_LABELS);

```

```

//设定显示格式(名称加百分比或数值)
pie.setPercentFormatString("#,###0.0#%");
//设定百分比显示格式
pie.setBackgroundPaint(Color.white);
pie.setSectionLabelFont(new Font("黑体", Font.TRUETYPE_FONT, 12));
//设定背景透明度 ( 0-1.0 之间 )
pie.setBackgroundAlpha(0.6f);
//设定前景透明度 ( 0-1.0 之间 )
pie.setForegroundAlpha(0.90f);
//输出文件到指定目录
String rfname = MathUtil.getRoundCode(12) + ".jpeg";
String fileName = "d:/test/" + rfname;
try {
//可以保存文件为jpg或png格式。
ChartUtilities.saveChartAsJPEG(new File(fileName), 100, chart, 600, 600);
//第一个参数为文件名
//第二个参数质量
//第三个参数为哪个chart创建图片
//第四个宽度
//第五个高度
} catch (IOException exz) {
System.out.print("....Can't Create image File");
}

```

其实使用JFreeChart创建图片很简单，不同的的图片类型区别在于设置数据集

四：创建曲线图

```

// create a default chart based on some sample data...
//曲线图标题
String title = "趋势分析";
//曲线图X轴提示
String domain = "月份走势";
//曲线图Y轴提示
String range = "应收余额";
//曲线图自标题
String subtitleStr = "2003财年分析";
//创建时间数据源
//每一个TimeSeries在图上是一条曲线
TimeSeries ca = new TimeSeries("用友");
for (int i = 1999; i < 2005; i++) {
    for (int mon = 0; mon < 12; mon++) {

```

```

//ca.add(new Month(mon + 1, i), new Double(500 + Math.random() * 100));
//TimeSeriesDataPair是一个时间点的数值体现
ca.add(new TimeSeriesDataPair(new Day(1, mon + 1, i),new Double(500 + Math.random() * 100)));
}

}

TimeSeries ibm = new TimeSeries("金蝶");
for (int i = 1999; i < 2005; i++) {
    for (int mon = 0; mon < 12; mon++) {
        //ibm.add(new Month(mon+1,i),new Double(400-Math.random()*100));
        ibm.add(new TimeSeriesDataPair(new Day(1, mon + 1, i),new Double(400 - Math.random() *
100)));
    }
}

TimeSeries king = new TimeSeries("东软");
for (int i = 1999; i < 2005; i++) {
    for (int mon = 0; mon < 12; mon++) {
        //ibm.add(new Month(mon+1,i),new Double(400-Math.random()*100));
        king.add(
            new TimeSeriesDataPair(new Day(1, mon + 1, i),new Double(300 - Math.random() * 100)));
    }
}

//时间曲线数据集合
TimeSeriesCollection dataset = new TimeSeriesCollection();
dataset.addSeries(ca);
dataset.addSeries(ibm);
dataset.addSeries(king);
//dataset.addSeries(jpy);
//dataset.addSeries(mav);
//时间曲线元素
JFreeChart chart =

ChartFactory.createTimeSeriesChart(title,domain,range,dataset,true,true,false);
// then customise it a little...
TextTitle subtitle = new TextTitle(subtitleStr, new Font("黑体", Font.BOLD, 12));
chart.addSubtitle(subtitle);
chart.setTitle(new TextTitle(title, new Font("隶书", Font.ITALIC, 15)));
//pie.setSeriesLabelFont(new Font("黑体", Font.BOLD, 15));
chart.setBackgroundPaint(new GradientPaint(0, 0, Color.white, 0, 1000, Color.blue));
//sysout
//输出文件到指定目录
String rfname = MathUtil.getRoundCode(22) + ".jpeg";

```

```

String fileName = "d:/test/" + rfname;
try {
//for
//System.out.println();
ChartUtilities.saveChartAsJPEG(new File(fileName), 100, chart, 600, 600);
// log.info("....Create image File:" + fileName);
} catch (IOException exz) {
System.out.print("....Can't Create image File");
}

```

五：创建柱状图

```

String title = "柱状图测试";
String domain = "单位比较";
String range = "数值";
//CategoryDataset data = DemoDatasetFactory.createCategoryDataset();
DefaultCategoryDataset data = new DefaultCategoryDataset();
for (int r = 0; r < 5; r++) {
String rowKey = "单位 [" + (r + 1) + "]";
//第一层循环：分析对象
for (int c = 0; c < 6; c++) {
//第二层循环：分析对象在时间点上的数据
String columnKey = "2001年" + (c + 1) + "月";
data.addValue(new Double(r * c + 5), rowKey, columnKey);
}
}
}

JFreeChart chart = ChartFactory.createVerticalBarChart(title, domain, range, data, true, true, false);
// then customise it a little...
chart.setBackgroundPaint(new GradientPaint(0, 0, Color.white, 1000, 0, Color.red));
chart.setTitle(new TextTitle(title, new Font("隶书", Font.ITALIC, 15)));
CategoryPlot plot = (CategoryPlot)chart.getPlot();
plot.setForegroundAlpha(0.9f);
plot.setValueLabelFont(new Font("黑体", Font.TRUETYPE_FONT, 12));
//plot.setSectionLabelFont(new Font("黑体", Font.TRUETYPE_FONT, 12));
//注意以下代码
NumberAxis verticalAxis = (NumberAxis)plot.getRangeAxis();
verticalAxis.setStandardTickUnits(NumberAxis.createIntegerTickUnits());
// 输出文件到指定目录
String rfname = MathUtil.getRoundCode(22) + "b.jpeg";
String fileName = "d:/test/" + rfname;
try {

```

```
ChartUtilities.saveChartAsJPEG(new File(fileName), 100, chart, 600, 600);
// log.info("....Create image File:" + fileName);
} catch (IOException exz) {
    System.out.print("....Cant 't Create image File");
}
```

六：结束语

个人感觉JFreeChart可以满足大部分图片创建的需要，美中不足的是：对字体的设置做的不够好，特别是使用中文的时候字体很不清晰。因为这个原因建议你自己去修改他的源代码最好使用properties文件去设置字体.还有就是文档要钱所以要多花点时间去看源代码。或多上社区.因为时间等原因我只介绍了三种图片的创建，其他类型的图片可以参考jfreechart提供的例子。

原作者：孤魂一笑(bingo_ge@hotmail.com)

来 源：ZDNet China

共有754位读者阅读过此文

上篇文章：Hibernate技术

下篇文章：用缓冲技术提高JSP应用的性能和稳定性

- 本周热门文章

- 相关文章

JFreeChart -- 柱状图

JFreeChart-饼状图-初级

使用JFreeChart创建图象

[网站首页](#) | [网站简介](#) | [网站留言](#) | [关于我们](#)

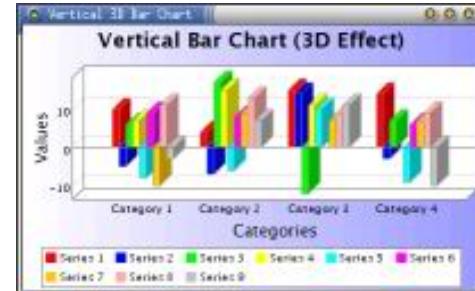
JAVA>>开源综合>>JFreeChart-饼状图-初级 [讨论论坛]

JFreeChart-饼状图-初级

作者：王恩建 来源：<http://www.sentom.net>

申明：未经作者同意，谢绝转载

JFreeChart，开放源码的Java图表生成组件（基于LGPL Licence），可以创建饼图、曲线图、柱状图、甘特图，生成PNG,JPG等图片格式文件。当前最新版本为0.9.11，右边是一张JFreeChart生成的统计图。本文讲解如何在Web利用JFreeChart生成饼状统计图。



一 安装配置

当前最新版本为0.9.11

下载jfreechart-0.9.11.zip

下载jfreechart-0.9.11.tar.gz

解压缩文件，jfreechart-0.9.11目录下的jfreechart-0.9.11.jar和 lib/jcommon-0.8.6.jar,lib/gnujaxp.jar都是开发运行需要的文件，把这三个文件放置到应用的lib目录下。

然后在web.xml文件中增加以下内容：

```
<servlet>
<servlet-name>DisplayChart</servlet-name>
<servlet-class>org.jfree.chart.servlet.DisplayChart</servlet-class>
</servlet>
```

安装配置完成，现在可以写程序生成我们需要的统计图了。

二 开始，在JSP中生成饼型统计图

```

<%@ page contentType="text/html; charset=GBK"%>
<%@ page import="org.jfree.data.DefaultPieDataset"%>
<%@ page import="org.jfree.chart.JFreeChart"%>
<%@ page import="org.jfree.chart.plot.PiePlot"%>
<%@ page import="org.jfree.chart.ChartRenderingInfo"%>
<%@ page import="org.jfree.chart.servlet.ServletUtilities"%>
<%@ page import="org.jfree.chart.urls.StandardPieURLGenerator"%>
<%@ page import="org.jfree.chart.entity.StandardEntityCollection"%>
<%
DefaultPieDataset data = new DefaultPieDataset();
data.setValue("六月",500);
data.setValue("七月",580);
data.setValue("八月",828);

PiePlot plot = new PiePlot(data);

JFreeChart chart = new JFreeChart("",JFreeChart.DEFAULT_TITLE_FONT, plot, true);
chart.setBackgroundPaint(java.awt.Color.white); //可选，设置图片背景色
chart.setTitle("www.SenTom.net 网站访问统计表"); //可选，设置图片标题

ChartRenderingInfo info = new ChartRenderingInfo(new StandardEntityCollection());
//500是图片长度，300是图片高度

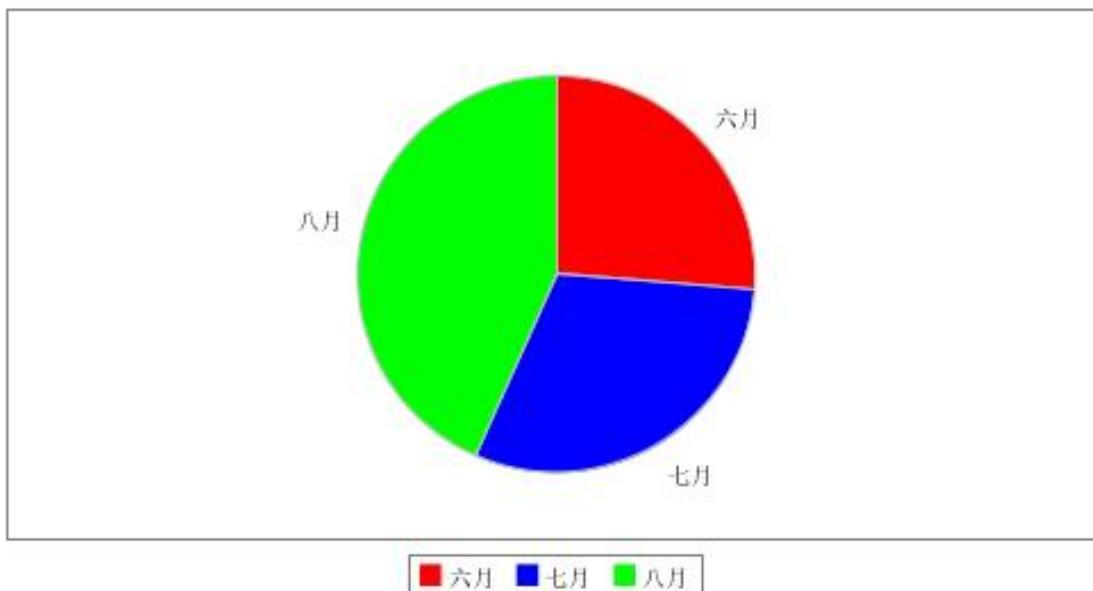
String filename = ServletUtilities.saveChartAsPNG(chart, 500, 300, info, session);
String graphURL = request.getContextPath() + "/servlet/DisplayChart?filename=" + filename;

%>
<HTML>
<HEAD>
    <TITLE>www.sentom.net</TITLE>
</HEAD>
<BODY>
<P ALIGN="CENTER">
    ">
</P>
</BODY>
</HTML>

```

就这么简单，生成的饼状图如下

www.SenTom.net 网站访问统计表



三 高级 , JSP+JAVABEAN生成饼状统计图

WebChart.java

```

/*
 * Created on 2003-9-9
 * http://www.sentom.net
 * To change the template for this generated file go to
 * Window>Preferences>Java>Code Generation>Code and Comments
 */
package net.sentom.chart;

import java.awt.Insets;
import java.awt.Font;
import java.io.PrintWriter;
import javax.servlet.http.HttpSession;
import org.jfree.data.*;
import org.jfree.chart.*;
import org.jfree.chart.plot.*;
import org.jfree.chart.entity.*;
import org.jfree.chart.urls.*;
import org.jfree.chart.servlet.*;
import org.jfree.chart.labels.StandardPieToolTipGenerator;

/**
 * @author sentom
 *
 * To change the template for this generated type comment go to
 * Window>Preferences>Java>Code Generation>Code and Comments
 */
public class WebChart {

```

```

private DefaultPieDataset data = new DefaultPieDataset();

public void setValue(String key,double value){
    data.setValue(key,value);
}

public String generatePieChart(String title, HttpSession session, PrintWriter pw) {
    String filename = null;
    try {
        //创建chart对象
        PiePlot plot = new PiePlot(data);
        plot.setInsets(new Insets(0, 5, 5, 5));
        //在统计图片上建连结
        plot.setURLGenerator(new StandardPieURLGenerator("link.jsp","section"));
        plot.setToolTipGenerator(new StandardPieToolTipGenerator());
        JFreeChart chart = new JFreeChart("", JFreeChart.DEFAULT_TITLE_FONT,
plot, true);

        chart.setBackgroundPaint(java.awt.Color.white);//设置图片的背景色

        Font font = new Font("黑体",Font.CENTER_BASELINE,20);//设置图片标题的字体
和大小
        TextTitle _title = new TextTitle(title);
        _title.setFont(font);
        chart.setTitle(_title);

        //把生成的图片放到临时目录
        ChartRenderingInfo info = new ChartRenderingInfo(new
StandardEntityCollection());
        //500是图片长度 , 300是图片高度
        filename = ServletUtilities.saveChartAsPNG(chart, 500, 300, info,
session);

        ChartUtilities.writeImageMap(pw, filename, info);
        pw.flush();

    } catch (Exception e) {
        System.out.println("Exception - " + e.toString());
        e.printStackTrace(System.out);
        filename = "public_error_500x300.png";
    }
    return filename;
}
}

```

JSP文件

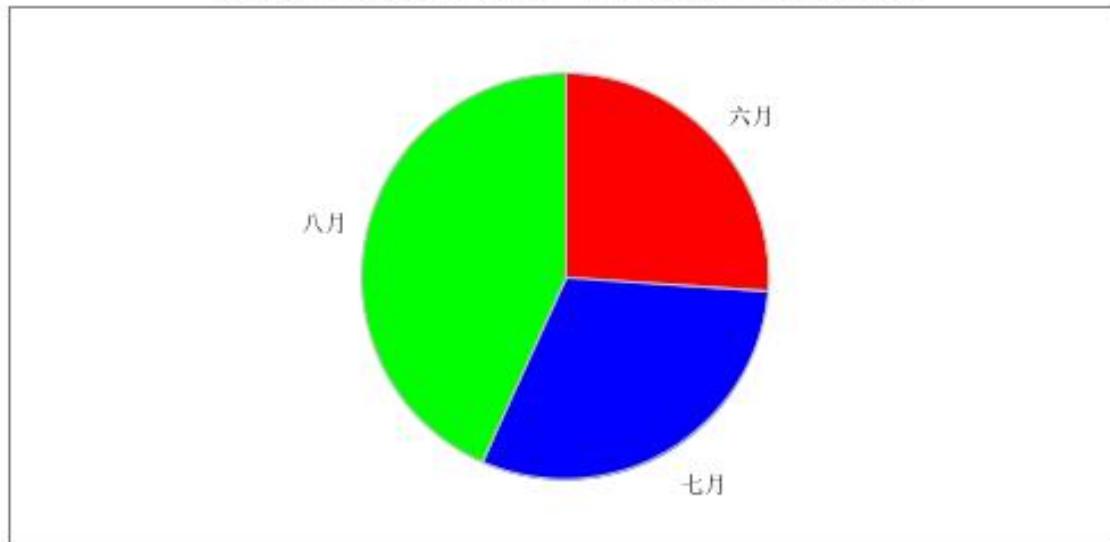
```
<%@ page contentType="text/html; charset=GBK"%>
<%@ page import = "java.io.PrintWriter" %>
<%@ page import="net.sentom.chart.WebChart"%>
<%
WebChart chart = new WebChart();
chart.setValue("六月",500);
chart.setValue("七月",580);
chart.setValue("八月",828);

String filename = chart.generatePieChart("www.SenTom.net 网站访问统计表", session, new
PrintWriter(out));
String graphURL = request.getContextPath() + "/servlet/DisplayChart?filename=" + filename;

%>
<HTML>
<HEAD>
    <TITLE>www.sentom.net</TITLE>
</HEAD>
<BODY>
<P ALIGN="CENTER">
    ">
</P>
</BODY>
</HTML>
```

生成的饼状图如下

www.SenTom.net 网站访问统计表



四 参考资源

<http://sourceforge.net/projects/jfreechart>
<http://www.jfree.org/jfreechart/index.html>
<http://www.jfree.org/phpBB2/viewforum.php?f=3>

原作者：王恩建
来 源：<http://www.sentom.net>
共有1229位读者阅读过此文

上篇文章：已经没有了

下篇文章：[Apache jakarta 各项目主要用途及简单说明](#)

- 本周热门文章

- 相关文章

[JFreeChart -- 柱状图](#)
[JFreeChart-饼状图-初级](#)
[使用JFreeChart创建图象](#)

JAVA>>开源综合>>JFreeChart-饼状图-中级 [讨论论坛]

JFreeChart-饼状图-中级

作者：王恩建 来源：<http://www.sentom.net>

申明：未经作者同意，谢绝转载

上次简单介绍了 JFreeChart 安装和如何生成简单的饼形图，详情请参考《JFreeChart-饼状图-初级》。上一次介绍的比较简单，生成的饼形图也比较粗糙。这一次，将介绍 JFreeChart 在生成饼形图能用到的其他特性以及生成 3D 饼形图。

— org.jfree.chart.plot.PiePlot

这个类提供了很多 set 方法用于控制生成饼型图，这些方法都是可选的。

setURLGenerator(PieURLGenerator generator)	在图片上建立连接，就是图片不同部分连接不同的资源；
setSectionLabelType(int type)	指定 section 标签的类型，共有 7 种类型。如果不指定，默认是 NAME_LABELS。 PiePlot.NO_LABELS PiePlot.NAME_LABELS PiePlot.VALUE_LABELS PiePlot.PERCENT_LABELS PiePlot.NAME_AND_VALUE_LABELS PiePlot.NAME_AND_PERCENT_LABELS PiePlot.VALUE_AND_PERCENT_LABELS。
setDefaultOutlinePaint(java.awt.Paint paint)	指定 section 轮廓线的颜色，如果不想要就直接传递 null 就可以了；
setDefaultOutlineStroke(java.awt.Stroke stroke)	指定 section 轮廓线的厚度；
setRadius(double percent) 和 setExplodePercent(int section, double percent)	抽离 section，就是把某一 section 从饼形图剥离出来，需要两个方法一起使用；
setStartAngle(double angle)	设置第一个 section 开始位置，默认是从 12 点钟方向开始；

setPaint(int section, java.awt.Paint paint)	指定 section 的颜色；
setDirection(int direction)	指定 section 顺序，默认是顺时针方向。 顺时针:PiePlot.CLOCKWISE 逆时针 :PiePlot.ANTICLOCKWISE

下面我给一个完整的范例，包括 WebChart.java 和 pie.jsp 两个文件；

WebChart.java

```
/*
 * Created on 2003-9-9
 * http://www.sentom.net
 * To change the template for this generated file go to
 * Window>Preferences>Java>Code Generation>Code and Comments
 */
package net.sentom.chart;

import java.awt.Insets;
import java.awt.Color;
import java.awt.Font;
import java.awt.BasicStroke;
import java.io.PrintWriter;
import javax.servlet.http.HttpSession;
import org.jfree.data.*;
import org.jfree.chart.*;
import org.jfree.chart.plot.*;
import org.jfree.chart.entity.*;
import org.jfree.chart.urls.*;
import org.jfree.chart.servlet.*;
import org.jfree.chart.labels.StandardPieToolTipGenerator;

/**
 * @author sentom
 *
 * To change the template for this generated type comment go to
 * Window>Preferences>Java>Code Generation>Code and Comments
 */
public class WebChart {
    private DefaultPieDataset data = new DefaultPieDataset();

    public void setValue(String key,double value){
        data.setValue(key,value);
    }
}
```

```
public String generatePieChart(String title, HttpSession session, PrintWriter pw) {  
    String filename = null;  
    try {  
        JFreeChart chart = ChartFactory.createPieChart("饼型图", // chart title  
                                                       data, // data  
                                                       true, // include legend  
                                                       true,  
                                                       false  
        );  
        //设置图片的背景色  
        chart.setBackgroundPaint(java.awt.Color.white);  
        //设置图片标题的字体和大小  
        Font font = new Font("黑体",Font.CENTER_BASELINE,20);  
        TextTitle _title = new TextTitle(title);  
        _title.setFont(font);  
        chart.setTitle(_title);  
        //chart.setAntiAlias(false);  
  
        PiePlot plot = (PiePlot) chart.getPlot();  
        plot.setInsets(new Insets(5, 5, 5, 5));  
        //在统计图片上建连结  
        plot.setURLGenerator(new StandardPieURLGenerator("link.jsp","section"));  
        //指定 section 标签的的类型  
        plot.setSectionLabelType(PiePlot.NAME_AND_VALUE_LABELS);  
        //指定 section 轮廓线的颜色  
        plot.setDefaultOutlinePaint(new Color(0x99, 0x99, 0xFF));  
        //指定 section 轮廓线的厚度  
        plot.setDefaultOutlineStroke(new BasicStroke(0));  
        plot.setRadius(0.70);  
        //抽离一个 section 出来  
        plot.setExplodePercent(1, 1.00);  
        //设置第一个 section 的开始位置 ,默认是12点钟方向  
        plot.setStartAngle(270);  
        //指定 section 的色彩  
        plot.setPaint(1, new Color(0x99, 0x99, 0xFF));  
        //指定 section 按逆时针方向依次显示 ,默认是顺时针方向  
        plot.setDirection(PiePlot.ANTICLOCKWISE);  
        //指定显示的饼图上圆形还椭圆形  
        //plot.setCircular(false);  
        plot.setToolTipGenerator(new StandardPieToolTipGenerator());  
  
        //把生成的图片放到临时目录  
        ChartRenderingInfo info = new ChartRenderingInfo(new
```

```

StandardEntityCollection());
//500是图片长度，300是图片高度
filename = ServletUtilities.saveChartAsPNG(chart, 500, 300, info,
session);

ChartUtilities.writeImageMap(pw, filename, info);
pw.flush();

} catch (Exception e) {
    System.out.println("Exception - " + e.toString());
    e.printStackTrace(System.out);
    filename = "public_error_500x300.png";
}
return filename;
}
}
}

```

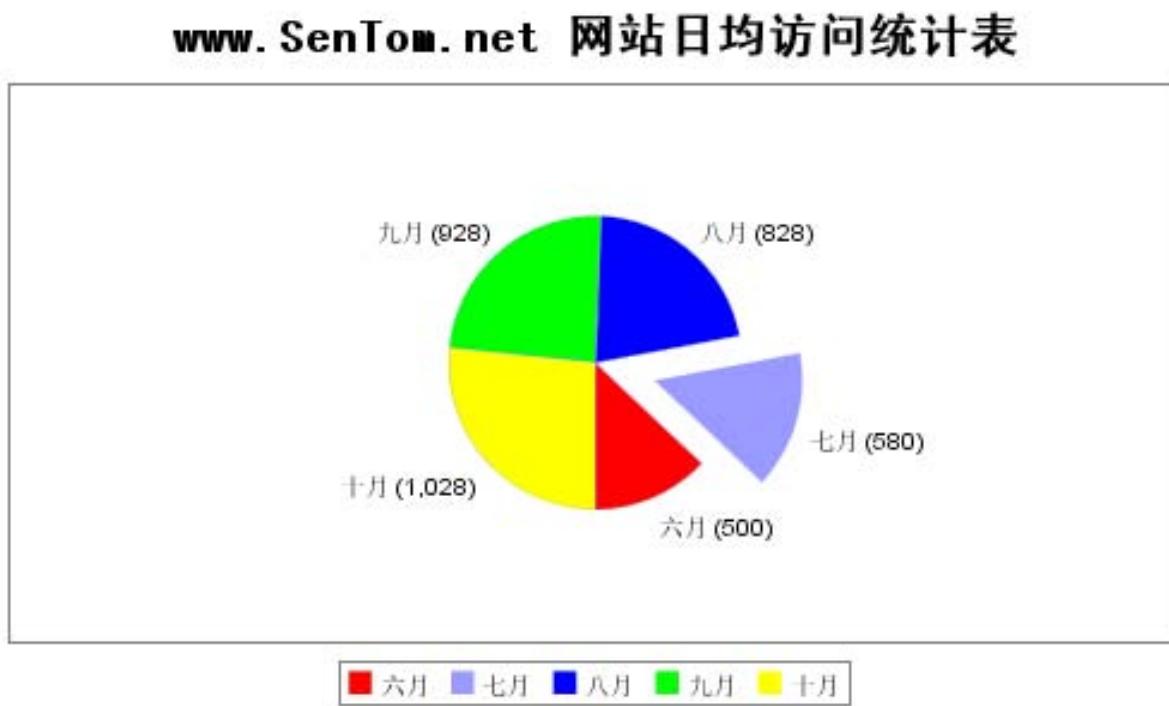
pie.jsp

```

<%@ page contentType="text/html;charset=GBK"%>
<%@ page import = "java.io.PrintWriter" %>
<%@ page import="net.sentom.chart.WebChart"%>
<%
WebChart chart = new WebChart();
chart.setValue("六月",500);
chart.setValue("七月",580);
chart.setValue("八月",828);
chart.setValue("九月",928);
chart.setValue("十月",1028);

String filename = chart.generatePieChart("www.SenTom.net 网站日均访问统计表", session, new
PrintWriter(out));
String graphURL = request.getContextPath() + "/servlet/DisplayChart?filename=" + filename;
%>
<HTML>
<HEAD>
    <TITLE>www.sentom.net</TITLE>
</HEAD>
<BODY>
<P ALIGN="CENTER">
    ">
</P>
</BODY>
</HTML>
</BODY>
</HTML>
```

运行结果如下：



二 org.jfree.chart.plot.Pie3DPlot

使用该类指定 3D 饼形图的属性。Pie3DPlot 是上面讲到的 PiePlot 的子类，所以呢，上面介绍的方法 在 Pie3DPlot 中都适用。这里不在重复，只介绍一个上面没有介绍的 setForegroundAlpha(float alpha) 方法，设置图片的透明度。

对上面 WebChar.java 程序做一点改动：

```
/*
 * Created on 2003-9-9
 * http://www.sentom.net
 * To change the template for this generated file go to
 * Window>Preferences>Java>Code Generation>Code and Comments
 */
package net.sentom.chart;

import java.awt.Insets;
import java.awt.Color;
```

```
import java.awt.Font;
import java.awt.BasicStroke;
import java.io.PrintWriter;
import javax.servlet.http.HttpSession;
import org.jfree.data.*;
import org.jfree.chart.*;
import org.jfree.chart.plot.*;
import org.jfree.chart.entity.*;
import org.jfree.chart.urls.*;
import org.jfree.chart.servlet.*;
import org.jfree.chart.labels.StandardPieToolTipGenerator;

/**
 * @author sentom
 *
 * To change the template for this generated type comment go to
 * Window>Preferences>Java>Code Generation>Code and Comments
 */
public class WebChart {
    private DefaultPieDataset data = new DefaultPieDataset();

    public void setValue(String key,double value){
        data.setValue(key,value);
    }

    public String generatePieChart(String title, HttpSession session, PrintWriter pw) {
        String filename = null;
        try {
            JFreeChart chart = ChartFactory.createPie3DChart("饼型图", // chart title
                    data, // data
                    true, // include legend
                    true,
                    false
            );
            //设置图片的背景色
            chart.setBackgroundPaint(java.awt.Color.white);
            //设置图片标题的字体和大小
            Font font = new Font("黑体",Font.CENTER_BASELINE,20);
            TextTitle _title = new TextTitle(title);
            _title.setFont(font);
            chart.setTitle(_title);
            //chart.setAntiAlias(false);

            PiePlot plot = (PiePlot) chart.getPlot();
            plot.setInsets(new Insets(5, 5, 5, 5));
        }
    }
}
```

```

//在统计图片上建连结
plot.setURLGenerator(new StandardPieURLGenerator("link.jsp","section"));
//指定 section 标签的的类型
plot.setSectionLabelType(PiePlot.NAME_LABELS);
//指定 section 轮廓线的颜色
plot.setDefaultValuePaint(null);
//指定 section 轮廓线的厚度
plot.setDefaultOutlineStroke(new BasicStroke(0));
//设置第一个 section 的开始位置 , 默认是12点钟方向
plot.setStartAngle(270);
//指定 section 的色彩
plot.setPaint(1, new Color(0x99, 0x99, 0xFF));
//指定 section 按逆时针方向依次显示 , 默认是顺时针方向
plot.setDirection(PiePlot.ANTICLOCKWISE);
//指定显示的饼图上圆形还椭圆形
//plot.setCircular(false);
plot.setToolTipGenerator(new StandardPieToolTipGenerator());
//指定图片的透明度
plot.setForegroundAlpha(0.5f);

//把生成的图片放到临时目录
ChartRenderingInfo info = new ChartRenderingInfo(new

StandardEntityCollection());
//500是图片长度 , 300是图片高度
filename = ServletUtilities.saveChartAsPNG(chart, 500, 300, info,
session);

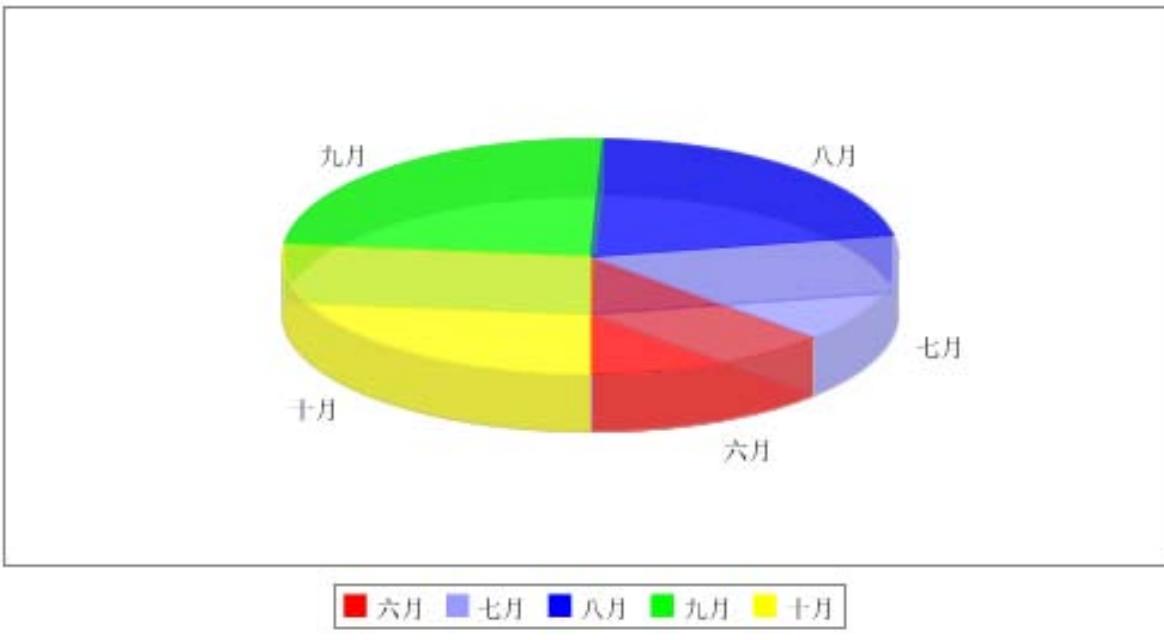
ChartUtilities.writeImageMap(pw, filename, info);
pw.flush();

} catch (Exception e) {
System.out.println("Exception - " + e.toString());
e.printStackTrace(System.out);
filename = "public_error_500x300.png";
}
return filename;
}
}

```

pie.jsp文件不变 , 运行的结果如下 :

www.SenTom.net 网站日均访问统计表



三 结论

用 JFreeChart 创建饼形图讲完了，不过都是基于 Web，欢迎有使用过 JFreeChart 的朋友不吝指教，我的 email 是 sentom@tom.com。接来下会继续介绍 JFreeChart 如何创建其他的类型的图表。

原作者：王恩建
来 源：<http://www.sentom.net>
共有1025位读者阅读过此文

上篇文章：第32期-制作条码(barcode)

下篇文章：第33期-StrutsMenu-Jakarta

- 本周热门文章

- 相关文章

- CLI学习笔记
- 第35期-Tomcat Admin Tree -Jakarta Struts系列(III)
- 第33期-StrutsMenu-Jakarta
- JFreeChart-饼状图-中级
- 第32期-制作条码(barcode)
- 第31期-Jakarta Commons Jexl
- 第30期-Jakarta Commons Betwixt

JAVA>>开源综合>>JFreeChart -- 柱状图 [讨论论坛]

JFreeChart -- 柱状图

作者：王恩建 来源：<http://www.sentom.net>

申明：未经作者同意，谢绝转载

不知道 JFreeChart 项目组出于什么目的，他们提供的 sample code 里面的例子给人感觉就是乱，同样的结果可以用不同方式、使用不的方法。再加上网上流传的 JFreeChart 旧的使用方法，使刚接触 JFreeChart 的朋友进到 JFreeChart 中不花一些时间很难理出一个头绪来。

前面已经介绍过 JFreeChart 生成饼状图，可能由于上面提到的原因，有些朋友来信和留言希望介绍 JFreeChart 如何生成柱状图。抄袭别人一句话：Help others as well as to help myself。这次我们介绍使用 JFreeChart 生成柱状图，首先从一个最简单的例子开始。

一 最简单的例子

为了降低门槛，让大家心理有个底，先介绍一个简单的不能再简单的例子，图片中的各类属性都采用默认值。

```
<%@ page contentType="text/html; charset=GBK"%>
<%@ page import="org.jfree.chart.ChartFactory,
                 org.jfree.chart.JFreeChart,
                 org.jfree.chart.plot.PlotOrientation,
                 org.jfree.chart.servlet.ServletUtilities,
                 org.jfree.data.DefaultCategoryDataset"%>
<%
DefaultCategoryDataset dataset = new DefaultCategoryDataset();
dataset.addValue(300, "广州", "苹果");
dataset.addValue(200, "广州", "梨子");
dataset.addValue(500, "广州", "葡萄");
dataset.addValue(340, "广州", "芒果");
dataset.addValue(280, "广州", "荔枝");

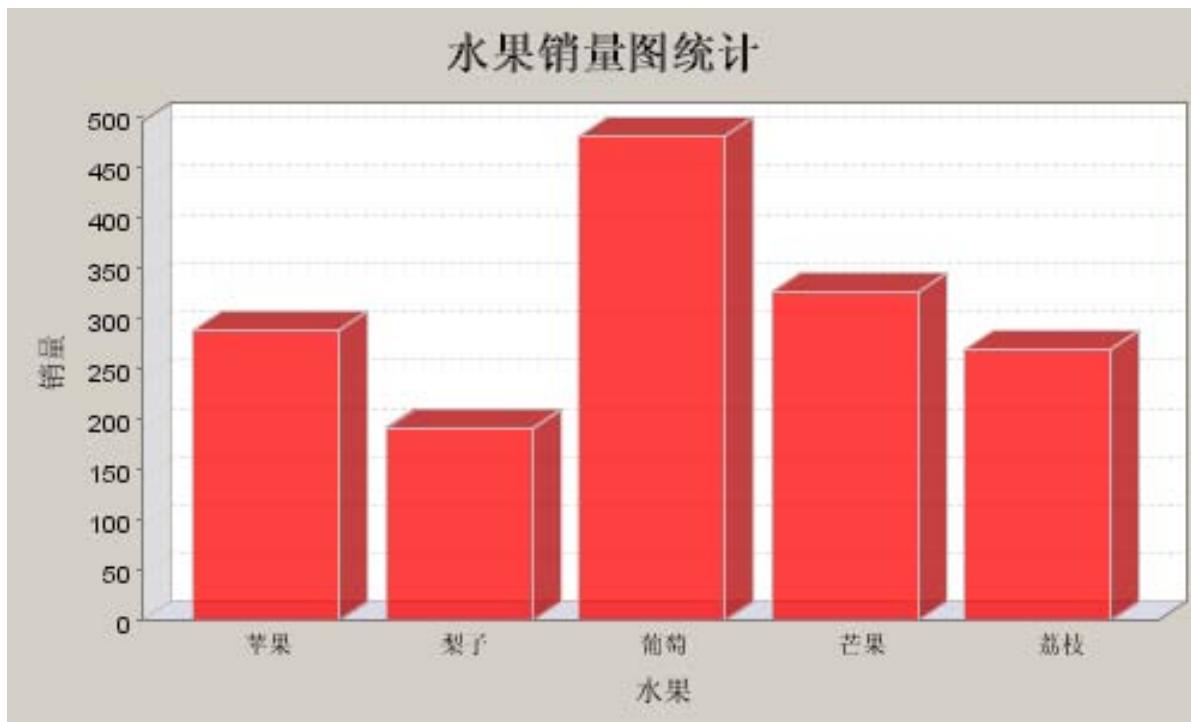
JFreeChart chart = ChartFactory.createBarChart3D("水果销量统计图",
                                                 "水果",
                                                 "销量",
                                                 dataset,
                                                 PlotOrientation.VERTICAL,
                                                 false,
```

```

        false,
        false);

String filename = ServletUtilities.saveChartAsPNG(chart, 500, 300, null,
session);
String graphURL = request.getContextPath() + "/servlet/DisplayChart?
filename=" + filename;
%>
">
```

这个 JSP 程序运行的结果如下图



图一

二 柱状图高级特性

上面的程序简单，但生成的柱状图也很简单。更多的时候，我们可能需要不同的效果。org.jfree.chart.ChartFactory 这个工厂类有 createBarChart, createStackedBarChart, createBarChart3D, createStackedBarChart3D 这几个工厂方法创建不同类型的柱状图。关于这四个方法的 JFreeChart 的 Java Doc API 文档有详细说明，比较重要的是 PlotOrientation.VERTICAL 让平行柱垂直显示，而 PlotOrientation.HORIZONTAL 则让平行柱水平显示。

几个对柱状图影响较大的几个类，它们分别是：

org.jfree.chart.axis.CategoryAxis
 org.jfree.chart.axis.ValueAxis
 org.jfree.chart.renderer.BarRenderer
 org.jfree.chart.renderer.BarRenderer3D

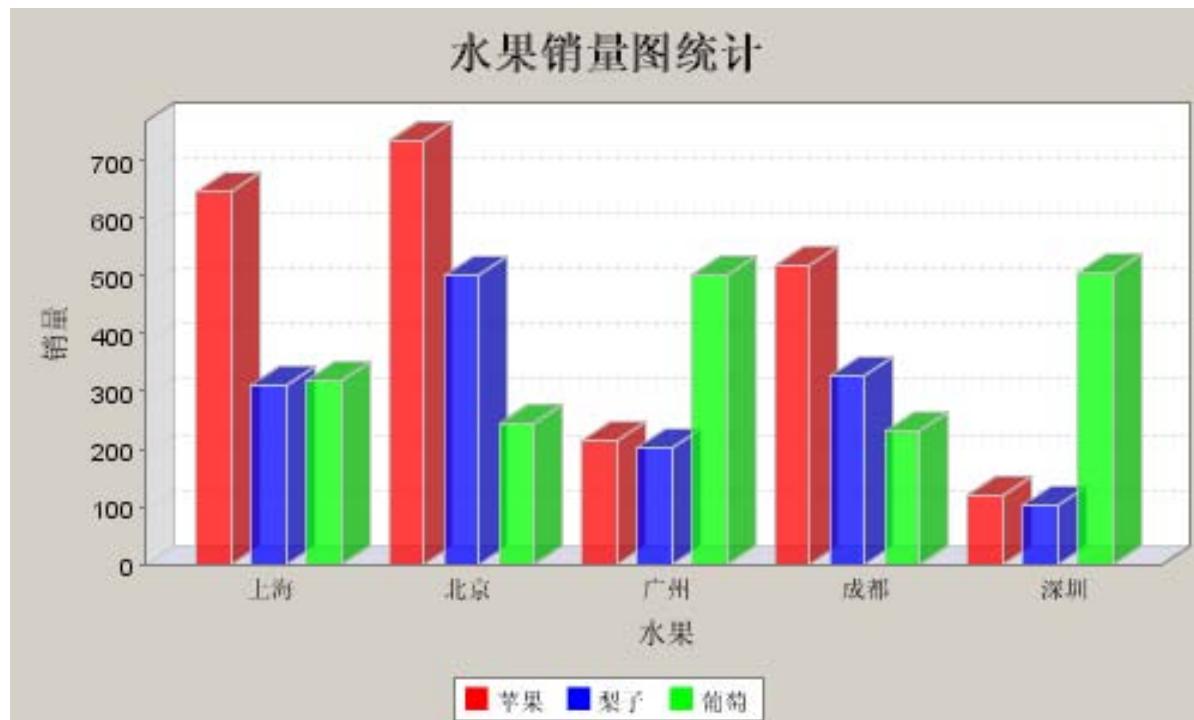
我们还是以实例来说明这几个类，先来假设一个需要统计的数据表：

	北京	上海	广州	成都	深圳
苹果	672	766	223	540	126
梨子	325	521	210	340	106
葡萄	332	256	523	240	526

根据上表数据，首先构造 CategoryDataset，这里不再使用上面简单例子里面的 DefaultCategoryDataset 类，而是 DatasetUtilities 更有效的构造 CategoryDataset，如下列代码：

```
double[][] data = new double[][] {{672, 766, 223, 540, 126}, {325, 521, 210, 340, 106}, {332, 256, 523, 240, 526}};
String[] rowKeys = {"苹果", "梨子", "葡萄"};
String[] columnKeys = {"北京", "上海", "广州", "成都", "深圳"};
CategoryDataset dataset = DatasetUtilities.createCategoryDataset(rowKeys, columnKeys, data);
```

用上面的 dataset 生成的 3D 柱状图



图二

org.jfree.chart.axis.CategoryAxis

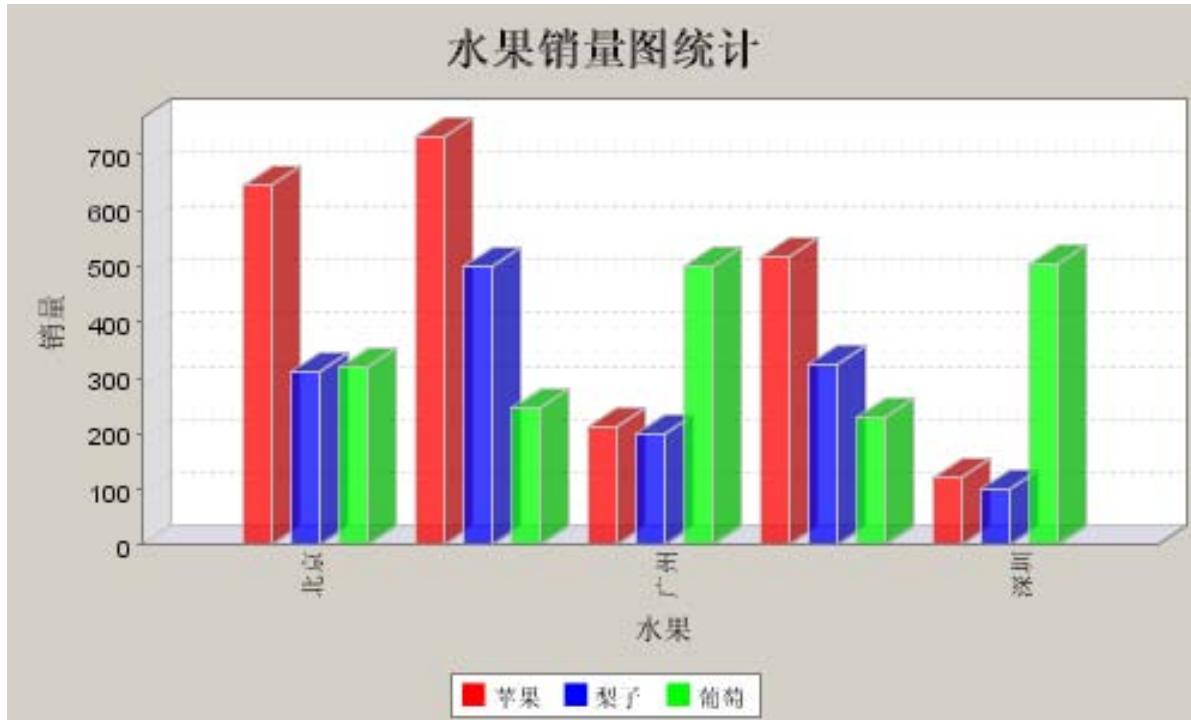
```
CategoryAxis domainAxis = plot.getDomainAxis();
//设置 columnKey 是否垂直显示
```

```

domainAxis.setVerticalCategoryLabels(true);
//设置距离图片左端距离
domainAxis.setLowerMargin(0.1);
//设置距离图片右端距离
domainAxis.setUpperMargin(0.1);
//设置 columnKey 是否间隔显示
domainAxis.setSkipCategoryLabelsToFit(true);
plot.setDomainAxis(domainAxis);

```

上面代码产生的效果如下图，注意与图二的区别。



图三

[org.jfree.chart.axis.ValueAxis](#)

```

ValueAxis rangeAxis = plot.getRangeAxis();
//设置最高的一个柱与图片顶端的距离
rangeAxis.setUpperMargin(0.15);
//设置最低的一个柱与图片底端的距离
//rangeAxis.setLowerMargin(0.15);
plot.setRangeAxis(rangeAxis);

```

上面代码产生的效果如下图，注意与图二的区别。

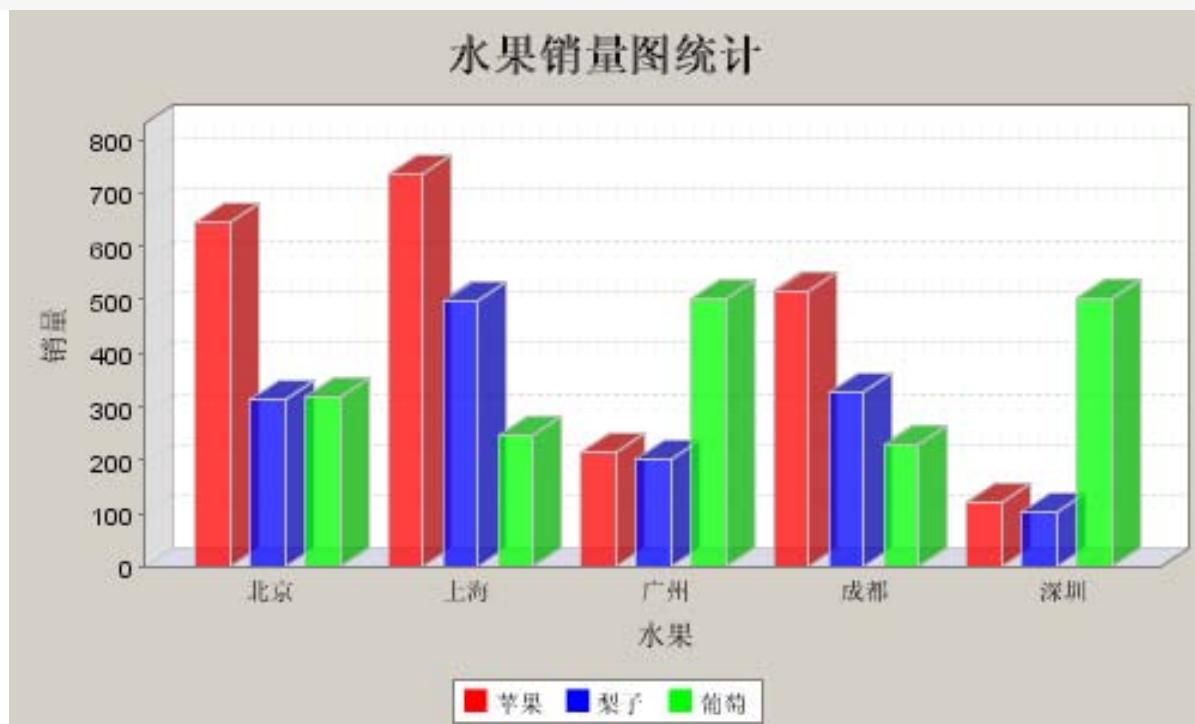


图 四

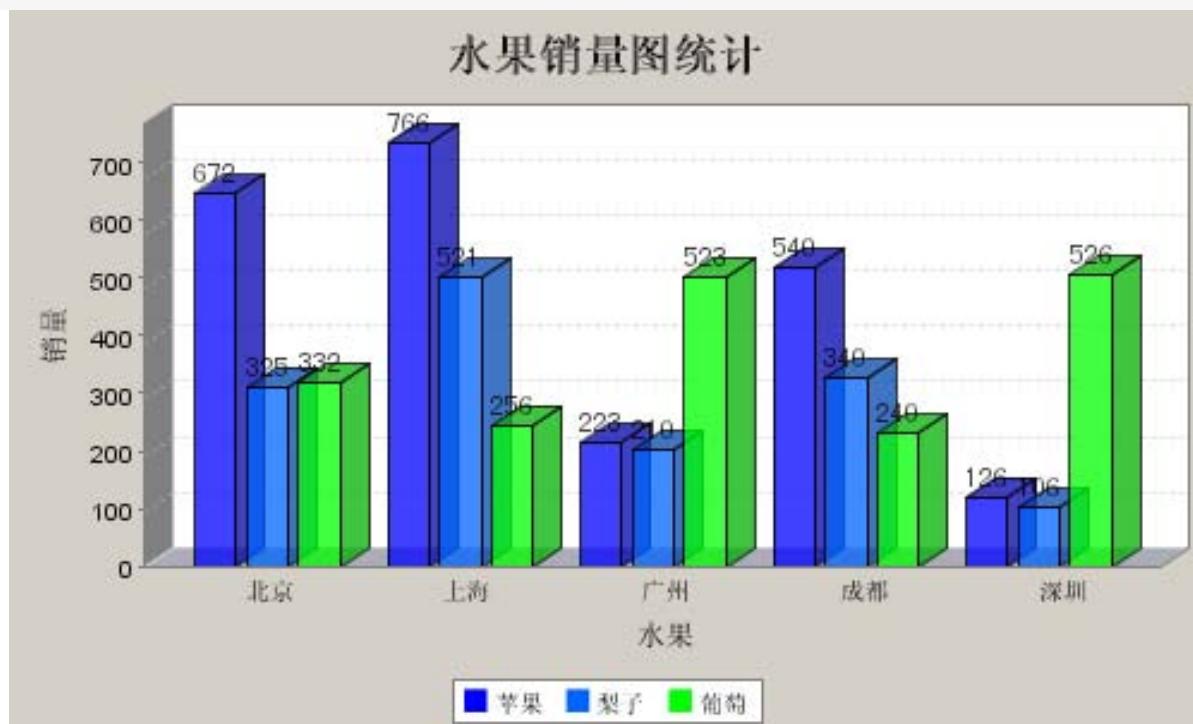
[org.jfree.chart.renderer.BarRenderer3D](#)

```

BarRenderer3D renderer = new BarRenderer3D();
renderer.setBaseOutlinePaint(Color.BLACK);
//设置 Wall 的颜色
renderer.setWallPaint(Color.gray);
//设置每种水果代表的柱的颜色
renderer.setSeriesPaint(0, new Color(0, 0, 255));
renderer.setSeriesPaint(1, new Color(0, 100, 255));
renderer.setSeriesPaint(2, Color.GREEN);
//设置每种水果代表的柱的 Outline 颜色
renderer.setSeriesOutlinePaint(0, Color.BLACK);
renderer.setSeriesOutlinePaint(1, Color.BLACK);
renderer.setSeriesOutlinePaint(2, Color.BLACK);
//设置每个地区所包含的平行柱的之间距离
renderer.setItemMargin(0.1);
//显示每个柱的数值，并修改该数值的字体属性
renderer.setItemLabelGenerator(new StandardCategoryItemLabelGenerator());
renderer.setItemLabelFont(new Font("黑体", Font.PLAIN, 12));
renderer.setItemLabelsVisible(true);

```

上面代码产生的效果如下图，注意与图二的区别。



图五

补充两个有用的方法

补充 org.jfree.chart.plot.CategoryPlot 的两个方法，这两个方法对所有类型的图表都有作用，因为在前面没有介绍，这里补充一下。

```
//设置地区、销量的显示位置
plot.setDomainAxisLocation(AxisLocation.TOP_OR_RIGHT);
plot.setRangeAxisLocation(AxisLocation.BOTTOM_OR_RIGHT);
```

上面代码产生的效果如下图，注意与图二的区别。

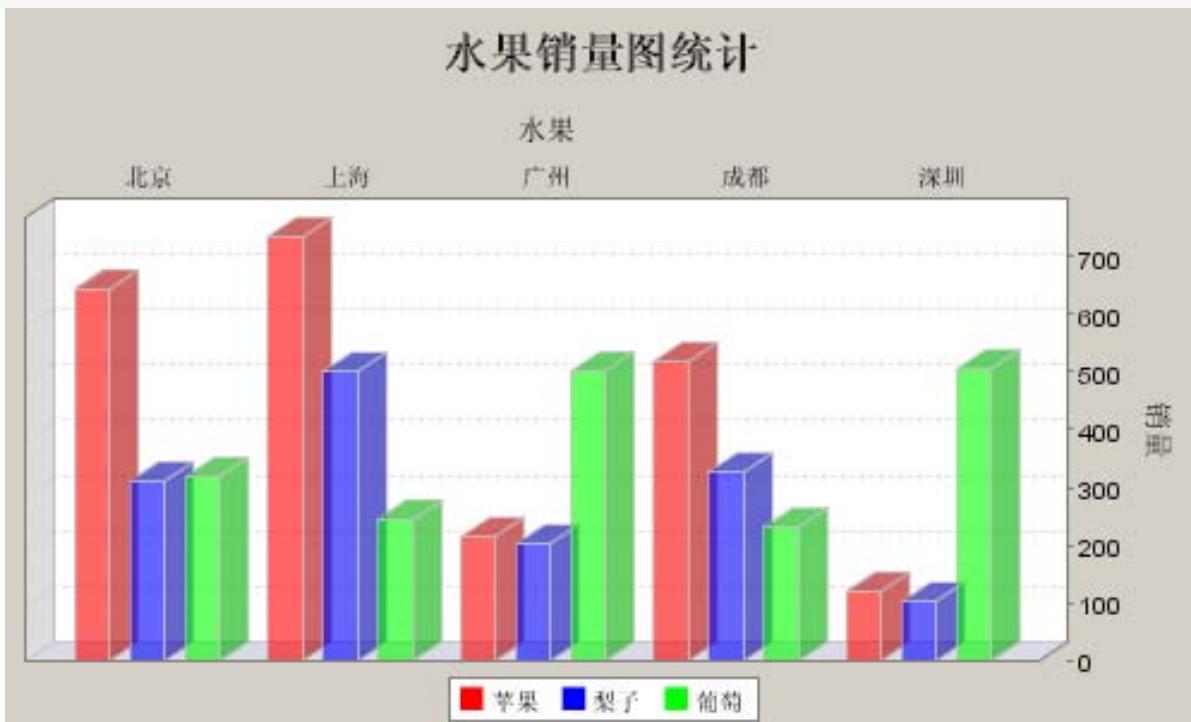


图 六

三 完整范例

前面都是一些代码片段，现在把这些片段组合成一个完整范例。

```

<%@ page contentType="text/html; charset=GBK"%>
<%@ page import="java.awt.Color,
                java.awt.Font,
                org.jfree.chart.ChartFactory,
                org.jfree.chart.JFreeChart,
                org.jfree.chart.plot.PlotOrientation,
                org.jfree.chart.servlet.ServletUtilities,
                org.jfree.data.CategoryDataset,
                org.jfree.data.DatasetUtilities,
                org.jfree.chart.plot.CategoryPlot,
                org.jfree.chart.axis.CategoryAxis,
                org.jfree.chart.axis.ValueAxis,
                org.jfree.chart.renderer.BarRenderer3D,
                org.jfree.chart.labels.StandardCategoryItemLabelGenerator,
                org.jfree.chart.axis.AxisLocation"%>

<%
double[][] data = new double[][] {{672, 766, 223, 540, 126},{325, 521, 210, 340, 106},{332, 256, 523, 240, 526}};
String[] rowKeys = {"苹果", "梨子", "葡萄"};
String[] columnKeys = {"北京", "上海", "广州", "成都", "深圳"};
CategoryDataset dataset = DatasetUtilities.createCategoryDataset(rowKeys,
    columnKeys, data);

```

```

JFreeChart chart = ChartFactory.createBarChart3D("水果销量图统计",
        null,
        null,
        dataset,
        PlotOrientation.VERTICAL,
        true, false, false);
chart.setBackgroundPaint(Color.WHITE);
CategoryPlot plot = chart.getCategoryPlot();

CategoryAxis domainAxis = plot.getDomainAxis();
domainAxis.setVerticalCategoryLabels(false);
plot.setDomainAxis(domainAxis);

ValueAxis rangeAxis = plot.getRangeAxis();
//设置最高的一个 Item 与图片顶端的距离
rangeAxis.setUpperMargin(0.15);
//设置最低的一个 Item 与图片底端的距离
rangeAxis.setLowerMargin(0.15);
plot.setRangeAxis(rangeAxis);

BarRenderer3D renderer = new BarRenderer3D();
renderer.setBaseOutlinePaint(Color.BLACK);
//设置 Wall 的颜色
renderer.setWallPaint(Color.gray);
//设置每种水果代表的柱的颜色
renderer.setSeriesPaint(0, new Color(0, 0, 255));
renderer.setSeriesPaint(1, new Color(0, 100, 255));
renderer.setSeriesPaint(2, Color.GREEN);
//设置每个地区所包含的平行柱的之间距离
renderer.setItemMargin(0.1);
//显示每个柱的数值，并修改该数值的字体属性
renderer.setItemLabelGenerator(new StandardCategoryItemLabelGenerator());
renderer.setItemLabelsVisible(true);
plot.setRenderer(renderer);

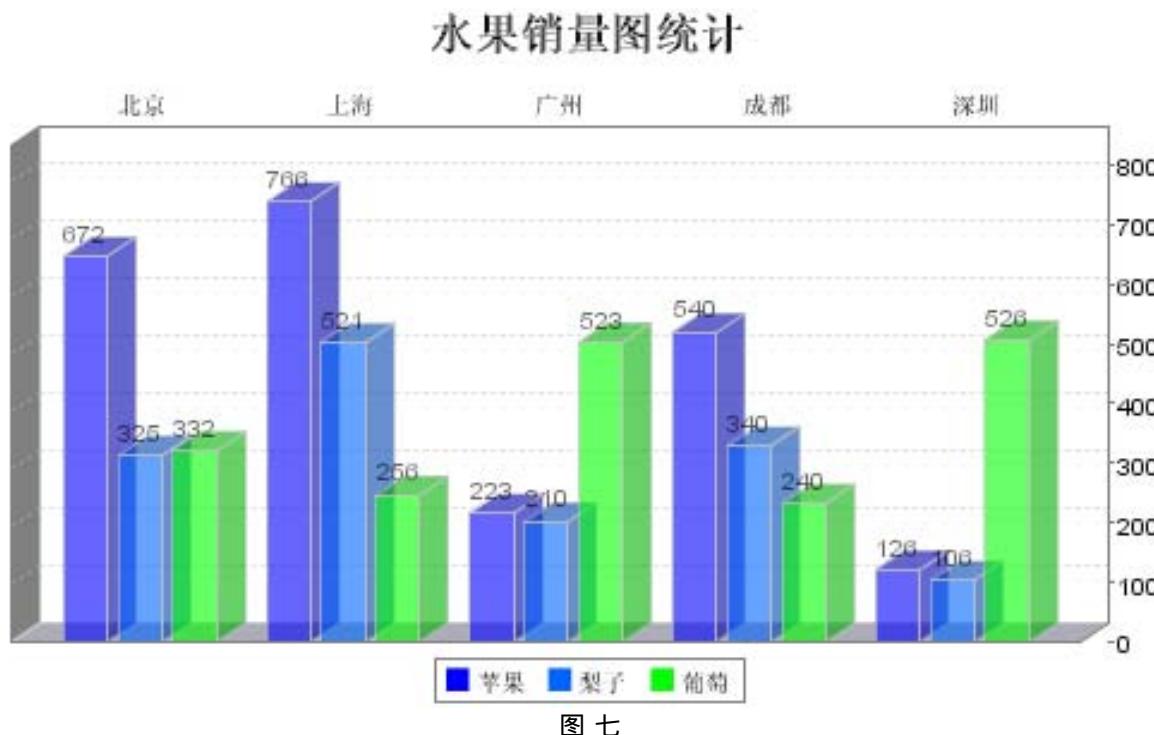
//设置柱的透明度
plot.setForegroundAlpha(0.6f);
//设置地区、销量的显示位置
plot.setDomainAxisLocation(AxisLocation.TOP_OR_RIGHT);
plot.setRangeAxisLocation(AxisLocation.BOTTOM_OR_RIGHT);

String filename = ServletUtilities.saveChartAsPNG(chart, 500, 300, null,
        session);
String graphURL = request.getContextPath() + "/servlet/DisplayChart?
filename=" + filename;
%>

```

```
">
```

看看程序运行的结果吧：



图七

三 总结

我只介绍了少量的方法，更多的请参考 JFreeChart 的 Java Doc API 文档和 Sample Code。文中的有些关于图表的术语不知道该如何正确的表达，如果你有任何关于本文的想法，可以与我联系 wayne@sentom.net.

原作者：王恩建
来 源：<http://www.sentom.net>
共有1371位读者阅读过此文

上篇文章：CLI学习笔记

下篇文章：用 Eclipse 、 Spindle 和 Tomcat 轻松搭建 Tapstry 开发环境

- 本周热门文章

- 相关文章

JFreeChart -- 柱状图
JFreeChart-饼状图-初级
使用JFreeChart创建图象