1. **柱形图,饼状图,折线图 示例**
2. **用struct将JFreeChart显示到客户端(页面)**
3. **JFreeChart Api**

**package** jfreechart;

**import** java.awt.BasicStroke;

**import** java.awt.Color;

**import** java.awt.Font;

**import** java.io.FileOutputStream;

**import** java.text.DecimalFormat;

**import** java.text.NumberFormat;

**import** org.jfree.chart.ChartFactory;

**import** org.jfree.chart.ChartUtilities;

**import** org.jfree.chart.JFreeChart;

**import** org.jfree.chart.axis.AxisLocation;

**import** org.jfree.chart.axis.CategoryAxis;

**import** org.jfree.chart.axis.CategoryLabelPositions;

**import** org.jfree.chart.axis.NumberAxis;

**import** org.jfree.chart.labels.ItemLabelAnchor;

**import** org.jfree.chart.labels.ItemLabelPosition;

**import** org.jfree.chart.labels.StandardCategoryItemLabelGenerator;

**import** org.jfree.chart.labels.StandardPieSectionLabelGenerator;

**import** org.jfree.chart.labels.StandardPieToolTipGenerator;

**import** org.jfree.chart.plot.CategoryPlot;

**import** org.jfree.chart.plot.PiePlot;

**import** org.jfree.chart.plot.PlotOrientation;

**import** org.jfree.chart.renderer.category.BarRenderer3D;

**import** org.jfree.chart.renderer.category.CategoryItemRenderer;

**import** org.jfree.chart.title.LegendTitle;

**import** org.jfree.chart.title.TextTitle;

**import** org.jfree.data.category.DefaultCategoryDataset;

**import** org.jfree.data.general.DefaultPieDataset;

**import** org.jfree.ui.TextAnchor;

**import** org.jfree.util.Rotation;

**public** **class** Jfreechart {

//柱状图

**public** **static** **void** barCharts (){

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");

dataset.addValue(**null**, "广州", "橘子");

/\*以二维数组的形式

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","武汉","深圳"};

CategoryDataset dataset=DatasetUtilities.createCategoryDataset(rowKeys, columnKeys,data);

\*/

JFreeChart chart=ChartFactory.*createBarChart3D*("水果销售统计图", "水果", "销量", dataset,

PlotOrientation.*VERTICAL*, //竖直显示

**true**, **false**, **false**);

//重新设置图标标题,改变字体

chart.setTitle(**new** TextTitle("水果销售统计",**new** Font("黑体",Font.*BOLD*,20)));

//取得统计图标的第一个图例

LegendTitle legend=chart.getLegend(0);

//修改图例的字体,必须把显示图片设置为ture,否则会包空指针异常

legend.setItemFont(**new** Font("宋体",Font.*BOLD*,20));

//获取柱状图的Plot对象

CategoryPlot plot=chart.getCategoryPlot();

//设定图表数据显示部分背景色

plot.setBackgroundPaint(Color.*PINK*);

//数据区的方向

plot.setOrientation(PlotOrientation.*VERTICAL*);

//分类轴的位置（参数常量在org.jfree.chart.axis.AxisLocation类中定义）

plot.setDomainAxisLocation(AxisLocation.*BOTTOM\_OR\_RIGHT*);

//分类轴网格是否可见(竖的分割线)

plot.setDomainGridlinesVisible(**false**);

//数据图例和数据区的位置（参数常量在org.jfree.chart.axis.AxisLocation类中定义）

plot.setRangeAxisLocation(AxisLocation.*TOP\_OR\_LEFT*);

plot.setRangeGridlinesVisible(**true**);

//取得横轴

CategoryAxis categoryAxis=plot.getDomainAxis();

//设置横轴显示标签的字体

categoryAxis.setLabelFont(**new** Font("隶书",Font.*BOLD*,20));

//分类标签以45度角倾斜

categoryAxis.setCategoryLabelPositions(CategoryLabelPositions.*UP\_45*);

//分类标签显示的字体

categoryAxis.setTickLabelFont(**new** Font("宋体",Font.*BOLD*,20));

//取得纵轴

NumberAxis numberAxis=(NumberAxis) plot.getRangeAxis();

//设置纵轴显示标签的字体

numberAxis.setLabelFont(**new** Font("隶书",Font.*BOLD*,20));

//数据轴是否反向（默认为false）

numberAxis.setInverted(**false**);

//数据轴的数据标签是否自动确定（默认为true）

numberAxis.setAutoTickUnitSelection(**true**);

//设置最高的一个柱与图片顶端的距离

numberAxis.setUpperMargin(0.2);

//numberAxis.setLowerMargin(0.2);

//设置显示小于最高一个柱的值100所有柱子,纵轴从柱子最大值-100开始显示

numberAxis.setFixedAutoRange(400);

//设置整数显示

numberAxis.setStandardTickUnits(NumberAxis.*createIntegerTickUnits*());

numberAxis.setNegativeArrowVisible(**true**);

//设置最大值

numberAxis.setUpperBound(1000);

//设置最小值,如果小于的话不显示

numberAxis.setLowerBound(150);

//设置百分比显示

//numberAxis.setNumberFormatOverride(new DecimalFormat("0%"));

//numberAxis.setNumberFormatOverride(new DecimalFormat("0.00%"));

//设置最小显示数,小于的话会显示在中间(正负)

//numberAxis.setAutoRangeMinimumSize(1);

plot.setNoDataMessage("没有使用数据");

plot.setNoDataMessagePaint(Color.*blue*);

//柱子对象

BarRenderer3D renderer=**new** BarRenderer3D();

//设置柱子的宽度

renderer.setMaximumBarWidth(0.05);

//设置柱子的高度

renderer.setMinimumBarLength(0.2);

//设置柱子的颜色

renderer.setSeriesPaint(0,Color.*GREEN*);

//设置每种水果代表的柱的 边框颜色

renderer.setSeriesOutlinePaint(0, Color.*BLACK*);

//设置柱子边框可见

renderer.setDrawBarOutline(**true**);

//设置柱子默认的边框颜色,必须设置边框可见才起效

renderer.setBaseOutlinePaint(Color.*BLACK*);

//设置柱子的纵横背景色

renderer.setWallPaint(Color.*BLUE*);

//设置平行柱的之间距离

renderer.setItemMargin(0.5);

//显示每个柱子的数值,并修改改数值的字体属性

renderer.setBaseItemLabelGenerator(**new** StandardCategoryItemLabelGenerator());

renderer.setBaseItemLabelFont(**new** Font("隶书",Font.*BOLD*,22));

renderer.setBaseItemLabelsVisible(**true**);

//设置显示值的位置

renderer.setBasePositiveItemLabelPosition(**new** ItemLabelPosition(

ItemLabelAnchor.*OUTSIDE12*, TextAnchor.*BASELINE\_CENTER*));

renderer.setItemLabelAnchorOffset(12D);// 设置柱形图上的文字偏离值

//将修改后的属性值保存到图中,这一步很重要,否则上面对颜色的设置都无效

plot.setRenderer(renderer);

//设置柱子的透明度,0.8相当于80%的透明度

plot.setForegroundAlpha(0.8f);

*drawToOut*(chart);

}

//饼图

**public** **static** **void** pieCharts(){

//与柱状图和折线图不同

DefaultPieDataset dataset = **new** DefaultPieDataset();

dataset.setValue("管理人员",10.02D);

dataset.setValue("市场人员",20.23D);

dataset.setValue("开发人员",60.02D);

dataset.setValue("OEM人员",10.02D);

dataset.setValue("其他人员",5.11D);

//createPieChart 2D; createPieChart3D 3D

JFreeChart chart=ChartFactory.*createPieChart*("公司组织架构图", dataset, **true**, **true**, **false**);

//设置标题字体,为了防止中文乱码

chart.setTitle(**new** TextTitle("公司组织架构统计图",**new** Font("黑体",Font.*ITALIC*,22)));

//取得统计图标的第一个图例

LegendTitle legend=chart.getLegend(0);

//修改图例的字体,必须把显示图片设置为ture,否则会包空指针异常

legend.setItemFont(**new** Font("宋体",Font.*BOLD*,20));

//取得图表显示对象(与柱状图和折线图不同)

PiePlot plot=(PiePlot) chart.getPlot();

//设置区块标签的字体==为了防止中文乱码：必须设置字体

plot.setLabelFont(**new** Font("隶书",Font.*BOLD*,22));

//图形边框颜色

plot.setBaseSectionOutlinePaint(Color.*BLUE*);

//图形边框粗细

plot.setBaseSectionOutlineStroke(**new** BasicStroke(0.5f));

//设置饼状图的绘制方向,可以按顺时针方向绘制,也可以按逆时针方向绘制

plot.setDirection(Rotation.*ANTICLOCKWISE*);//逆时针,Rotation.CLOCKWISE顺时针

//设置绘制角度(图形旋转角度)

plot.setStartAngle(70);

//设置突出显示的数据块

//plot.setExplodePercent(1, 0.5D);

//plot.setExplodePercent("One", 0.5D);

//扇区分离显示,对3D图不起效

plot.setExplodePercent(dataset.getKey(0), 0.1d);

//分类标签与图的连接线颜色

plot.setLabelLinkPaint(Color.*BLUE*);

//分类标签边框颜色

plot.setLabelOutlinePaint(Color.*black*);

//分类标签阴影颜色

plot.setLabelShadowPaint(Color.*RED*);

//指定分类饼的颜色

plot.~~setSectionPaint~~(1, Color.*BLACK*);

//饼状图标签显示百分比 :自定义,{0}表示选项,{1}表示数值,{2}表示所占比例,小数点后两位

plot.setLabelGenerator(**new** StandardPieSectionLabelGenerator(

"{0}:{1}\r\n{2}",NumberFormat.*getNumberInstance*(),**new** DecimalFormat("0.00%")));

//图例显示百分比

plot.setLegendLabelGenerator(**new** StandardPieSectionLabelGenerator("{0}={2}"));

//指定显示的拼图为:圆形(true),还是椭圆形(false)

plot.setCircular(**true**);

//没有数据的时候显示的内容

plot.setNoDataMessage("没有可用的数据...");

//设置鼠标悬停提示

plot.setToolTipGenerator(**new** StandardPieToolTipGenerator());

//设置热点链接

//plot.setURLGenerator(new StandardPieURLGenerator("detail.jsp"));

*drawToOut*(chart);

}

**public** **static** **void** lineCharts(){

DefaultCategoryDataset dataSet=**new** DefaultCategoryDataset();

dataSet.addValue(0.0, "冰箱", "1月");

dataSet.addValue(4.2, "冰箱", "2月");

dataSet.addValue(3.9, "冰箱", "3月");

dataSet.addValue(1.0, "彩电", "1月");

dataSet.addValue(5.2, "彩电", "2月");

dataSet.addValue(7.9, "彩电", "3月");

dataSet.addValue(2.0, "洗衣机", "1月");

dataSet.addValue(9.2, "洗衣机", "2月");

dataSet.addValue(8.9, "洗衣机", "3月");

JFreeChart chart=ChartFactory.*createLineChart*("折线图",

"时间",

"销售额(百万)",

dataSet,

PlotOrientation.*VERTICAL*,

**true**,

**true**,

**false**);

// 获取一个图例

LegendTitle legendTitle = chart.getLegend(0);

// 设置图例字体

legendTitle.setItemFont(**new** Font("宋体", Font.*BOLD*, 18));

CategoryPlot plot = (CategoryPlot) chart.getPlot();

// 取得横轴

CategoryAxis categoryAxis = plot.getDomainAxis();

// 设置横轴的字体

categoryAxis.setLabelFont(**new** Font("宋体", Font.*BOLD*, 18));

// 设置分类标签字体

categoryAxis.setTickLabelFont(**new** Font("宋体", Font.*BOLD*, 12));

// 取得纵轴

NumberAxis numberAxis = (NumberAxis) plot.getRangeAxis();

// 设置纵轴的字体

numberAxis.setLabelFont(**new** Font("宋体", Font.*BOLD*, 14));

numberAxis.setStandardTickUnits(NumberAxis.*createIntegerTickUnits*());//显示整数

CategoryItemRenderer xylineandshaperenderer = plot.getRenderer();

xylineandshaperenderer.setBaseItemLabelsVisible(**true**);

xylineandshaperenderer

.setBasePositiveItemLabelPosition(**new** ItemLabelPosition(

ItemLabelAnchor.*OUTSIDE2*, TextAnchor.*BASELINE\_RIGHT*));

xylineandshaperenderer

.setBaseItemLabelGenerator(**new** StandardCategoryItemLabelGenerator());

*drawToOut*(chart);

}

**public** **static** **void** drawToOut(JFreeChart chart){

FileOutputStream fos;

**try** {

fos = **new** FileOutputStream("d://1.jpg");

ChartUtilities.*writeChartAsJPEG*(fos, chart, 1400, 800);

} **catch** (Exception e) {

e.printStackTrace();

}

System.*out*.println("成功");

}

**public** **static** **void** main(String[] args) {

Jfreechart.*barCharts*();

}

}

**----------------------------用struct将JFreeChart显示到客户端(页面)--------------------------------------------------**

public String showVoteChart()  
{  
   if("support".equals(voteType))  
   {  
    supportCount++;  
   }  
   else if("blackball".equals(voteType))  
   {  
    blackballCount++;  
   }  
   //supportCount = 0;//快速清零，不用重启系统  
   //blackballCount = 0;//快速清零，不用重启系统  
   this.chart = getChart();  
   return SUCCESS;  
}

/\*\*  
\* @return the supportCount  
\*/  
public static int getSupportCount()  
{  
   return supportCount;  
}

/\*\*  
\* @return the blackballCount  
\*/  
public static int getBlackballCount()  
{  
   return blackballCount;  
}

/\*\*  
\* @return the voteType  
\*/  
public String getVoteType()  
{  
   return voteType;  
}

/\*\*  
\* @param supportCount the supportCount to set  
\*/  
public static void setSupportCount(int supportCount)  
{  
   VoteChartAction.supportCount = supportCount;  
}

/\*\*  
\* @param blackballCount the blackballCount to set  
\*/  
public static void setBlackballCount(int blackballCount)  
{  
   VoteChartAction.blackballCount = blackballCount;  
}

/\*\*  
\* @param voteType the voteType to set  
\*/  
public void setVoteType(String voteType)  
{  
   this.voteType = voteType;  
}

}

=============================================

struts 配置

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

<package name="chart" extends="jfreechart-default">   
   <action name="show\_vote\_chart" class="com.guosen.weblogger.web.action.VoteChartAction" method="showVoteChart">  
    <result name="success" type="chart">  
     <param name="width">420</param>  
     <param name="height">250</param>   
    </result>  
   </action>  
</package>

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

页面代码

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

<div style="width:420px;height:250px;padding-left:100px;">  
    <iframe id="votechart" name="votechart" src="${base}/chart/show\_vote\_chart.action" frameborder="0" marginheight="0" marginwidth="0" scrolling="no" style="height:100%;width:100%"></iframe>  
   </div>  
   <div style="width:420px;height:25px;padding-left:100px;text-align:center;font-size:16px;">  
    <a href="#this" onclick="changeChart('support')">支持</a>&nbsp;&nbsp;  
    <a href="#this" onclick="changeChart('blackball')">反对</a>  
   </div>  
   <script type="text/javascript">  
    function changeChart(type)  
    {  
     if(type == "support")  
     {  
      coos.$id("votechart").src = "${base}/chart/show\_vote\_chart.action?voteType=support";  
     }  
     else if(type == "blackball")  
     {  
      coos.$id("votechart").src = "${base}/chart/show\_vote\_chart.action?voteType=blackball";  
     }  
     
    }  
   </script>  
   <hr />

---------------------------JFreeChart API---------------------------------

JFreeChart类：  
void setAntiAlias(boolean flag)    字体模糊边界  
void setBackgroundImage(Image image)   背景图片  
void setBackgroundImageAlignment(int alignment)  背景图片对齐方式（参数常量在org.jfree.ui.Align类中定义）  
void setBackgroundImageAlpha(float alpha)  背景图片透明度（0.0～1.0）  
void setBackgroundPaint(Paint paint)   背景色  
void setBorderPaint(Paint paint)   边界线条颜色  
void setBorderStroke(Stroke stroke)   边界线条笔触  
void setBorderVisible(boolean visible)   边界线条是否可见

-----------------------------------------------------------------------------------------------------------

TextTitle类：  
void setFont(Font font)     标题字体  
void setPaint(Paint paint)    标题字体颜色  
void setText(String text)    标题内容

-----------------------------------------------------------------------------------------------------------

StandardLegend(Legend)类：  
void setBackgroundPaint(Paint paint)   图示背景色  
void setTitle(String title)    图示标题内容  
void setTitleFont(Font font)    图示标题字体  
void setBoundingBoxArcWidth(int arcWidth)  图示边界圆角宽  
void setBoundingBoxArcHeight(int arcHeight)  图示边界圆角高  
void setOutlinePaint(Paint paint)   图示边界线条颜色  
void setOutlineStroke(Stroke stroke)   图示边界线条笔触  
void setDisplaySeriesLines(boolean flag)  图示项是否显示横线（折线图有效）  
void setDisplaySeriesShapes(boolean flag)  图示项是否显示形状（折线图有效）  
void setItemFont(Font font)    图示项字体  
void setItemPaint(Paint paint)    图示项字体颜色  
void setAnchor(int anchor)    图示在图表中的显示位置（参数常量在Legend类中定义）

-----------------------------------------------------------------------------------------------------------

Axis类：  
void setVisible(boolean flag)    坐标轴是否可见  
void setAxisLinePaint(Paint paint)   坐标轴线条颜色（3D轴无效）  
void setAxisLineStroke(Stroke stroke)   坐标轴线条笔触（3D轴无效）  
void setAxisLineVisible(boolean visible)  坐标轴线条是否可见（3D轴无效）  
void setFixedDimension(double dimension)  （用于复合表中对多坐标轴的设置）  
void setLabel(String label)    坐标轴标题  
void setLabelFont(Font font)    坐标轴标题字体  
void setLabelPaint(Paint paint)    坐标轴标题颜色  
void setLabelAngle(double angle)`   坐标轴标题旋转角度（纵坐标可以旋转）  
void setTickLabelFont(Font font)   坐标轴标尺值字体  
void setTickLabelPaint(Paint paint)   坐标轴标尺值颜色  
void setTickLabelsVisible(boolean flag)   坐标轴标尺值是否显示  
void setTickMarkPaint(Paint paint)   坐标轴标尺颜色  
void setTickMarkStroke(Stroke stroke)   坐标轴标尺笔触  
void setTickMarksVisible(boolean flag)   坐标轴标尺是否显示

ValueAxis(Axis)类：  
void setAutoRange(boolean auto)    自动设置数据轴数据范围  
void setAutoRangeMinimumSize(double size)  自动设置数据轴数据范围时数据范围的最小跨度  
void setAutoTickUnitSelection(boolean flag)  数据轴的数据标签是否自动确定（默认为true）  
void setFixedAutoRange(double length)   数据轴固定数据范围（设置100的话就是显示MAXVALUE到MAXVALUE-100那段数据范围）  
void setInverted(boolean flag)    数据轴是否反向（默认为false）  
void setLowerMargin(double margin)   数据轴下（左）边距  
void setUpperMargin(double margin)   数据轴上（右）边距  
void setLowerBound(double min)    数据轴上的显示最小值  
void setUpperBound(double max)    数据轴上的显示最大值  
void setPositiveArrowVisible(boolean visible)  是否显示正向箭头（3D轴无效）  
void setNegativeArrowVisible(boolean visible)  是否显示反向箭头（3D轴无效）  
void setVerticalTickLabels(boolean flag)  数据轴数据标签是否旋转到垂直  
void setStandardTickUnits(TickUnitSource source) 数据轴的数据标签（可以只显示整数标签，需要将AutoTickUnitSelection设false）

NumberAxis(ValueAxis)类：  
void setAutoRangeIncludesZero(boolean flag)  是否强制在自动选择的数据范围中包含0  
void setAutoRangeStickyZero(boolean flag)  是否强制在整个数据轴中包含0，即使0不在数据范围中  
void setNumberFormatOverride(NumberFormat formatter) 数据轴数据标签的显示格式  
void setTickUnit(NumberTickUnit unit)   数据轴的数据标签（需要将AutoTickUnitSelection设false）

DateAxis(ValueAxis)类：  
void setMaximumDate(Date maximumDate)   日期轴上的最小日期  
void setMinimumDate(Date minimumDate)   日期轴上的最大日期  
void setRange(Date lower,Date upper)   日期轴范围  
void setDateFormatOverride(DateFormat formatter) 日期轴日期标签的显示格式  
void setTickUnit(DateTickUnit unit)   日期轴的日期标签（需要将AutoTickUnitSelection设false）  
void setTickMarkPosition(DateTickMarkPosition position) 日期标签位置（参数常量在org.jfree.chart.axis.DateTickMarkPosition类中定义）

CategoryAxis(Axis)类：  
void setCategoryMargin(double margin)   分类轴边距  
void setLowerMargin(double margin)   分类轴下（左）边距  
void setUpperMargin(double margin)   分类轴上（右）边距  
void setVerticalCategoryLabels(boolean flag)  分类轴标题是否旋转到垂直  
void setMaxCategoryLabelWidthRatio(float ratio)  分类轴分类标签的最大宽度

-----------------------------------------------------------------------------------------------------------

Plot类：  
void setBackgroundImage(Image image)   数据区的背景图片  
void setBackgroundImageAlignment(int alignment)  数据区的背景图片对齐方式（参数常量在org.jfree.ui.Align类中定义）  
void setBackgroundPaint(Paint paint)   数据区的背景图片背景色  
void setBackgroundAlpha(float alpha)   数据区的背景透明度（0.0～1.0）  
void setForegroundAlpha(float alpha)   数据区的前景透明度（0.0～1.0）  
void setDataAreaRatio(double ratio)   数据区占整个图表区的百分比  
void setOutLinePaint(Paint paint)   数据区的边界线条颜色  
void setOutLineStroke(Stroke stroke)   数据区的边界线条笔触  
void setNoDataMessage(String message)   没有数据时显示的消息  
void setNoDataMessageFont(Font font)   没有数据时显示的消息字体  
void setNoDataMessagePaint(Paint paint)   没有数据时显示的消息颜色

CategoryPlot(Plot)类：  
void setDataset(CategoryDataset dataset)  数据区的2维数据表  
void setColumnRenderingOrder(SortOrder order)  数据分类的排序方式  
void setAxisOffset(Spacer offset)   坐标轴到数据区的间距  
void setOrientation(PlotOrientation orientation) 数据区的方向（PlotOrientation.HORIZONTAL或PlotOrientation.VERTICAL）  
void setDomainAxis(CategoryAxis axis)   数据区的分类轴  
void setDomainAxisLocation(AxisLocation location) 分类轴的位置（参数常量在org.jfree.chart.axis.AxisLocation类中定义）  
void setDomainGridlinesVisible(boolean visible)  分类轴网格是否可见  
void setDomainGridlinePaint(Paint paint)  分类轴网格线条颜色  
void setDomainGridlineStroke(Stroke stroke)  分类轴网格线条笔触  
void setRangeAxis(ValueAxis axis)   数据区的数据轴  
void setRangeAxisLocation(AxisLocation location) 数据轴的位置（参数常量在org.jfree.chart.axis.AxisLocation类中定义）  
void setRangeGridlinesVisible(boolean visible)  数据轴网格是否可见  
void setRangeGridlinePaint(Paint paint)   数据轴网格线条颜色  
void setRangeGridlineStroke(Stroke stroke)  数据轴网格线条笔触  
void setRenderer(CategoryItemRenderer renderer)  数据区的表示者（详见Renderer组）  
void addAnnotation(CategoryAnnotation annotation) 给数据区加一个注释  
void addRangeMarker(Marker marker,Layer layer)  给数据区加一个数值范围区域

PiePlot(Plot)类：  
void setDataset(PieDataset dataset)   数据区的1维数据表  
void setIgnoreNullValues(boolean flag)   忽略无值的分类  
void setCircular(boolean flag)    饼图是否一定是正圆  
void setStartAngle(double angle)   饼图的初始角度  
void setDirection(Rotation direction)   饼图的旋转方向  
void setExplodePercent(int section,double percent) 抽取的那块（1维数据表的分类下标）以及抽取出来的距离（0.0～1.0），3D饼图无效  
void setLabelBackgroundPaint(Paint paint)  分类标签的底色  
void setLabelFont(Font font)    分类标签的字体  
void setLabelPaint(Paint paint)    分类标签的字体颜色  
void setLabelLinkMargin(double margin)   分类标签与图的连接线边距  
void setLabelLinkPaint(Paint paint)   分类标签与图的连接线颜色  
void setLabelLinkStroke(Stroke stroke)   分类标签与图的连接线笔触  
void setLabelOutlinePaint(Paint paint)   分类标签边框颜色  
void setLabelOutlineStroke(stroke stroke)   分类标签边框笔触  
void setLabelShadowPaint(Paint paint)   分类标签阴影颜色  
void setMaximumLabelWidth(double width)   分类标签的最大长度（0.0～1.0）  
void setPieIndex(int index)    饼图的索引（复合饼图中用到）  
void setSectionOutlinePaint(int section,Paint paint) 指定分类饼的边框颜色  
void setSectionOutlineStroke(int section,Stroke stroke) 指定分类饼的边框笔触  
void setSectionPaint(int section,Paint paint)  指定分类饼的颜色  
void setShadowPaint(Paint paint)   饼图的阴影颜色  
void setShadowXOffset(double offset)   饼图的阴影相对图的水平偏移  
void setShadowYOffset(double offset)   饼图的阴影相对图的垂直偏移  
void setLabelGenerator(PieSectionLabelGenerator generator) 分类标签的格式，设置成null则整个标签包括连接线都不显示  
void setToolTipGenerator(PieToolTipGenerator generator)  MAP中鼠标移上的显示格式  
void setURLGenerator(PieURLGenerator generator)   MAP中钻取链接格式

PiePlot3D(PiePlot)类：  
void setDepthFactor(double factor)   3D饼图的Z轴高度（0.0～1.0）

MultiplePiePlot(Plot)类：  
void setLimit(double limit)    每个饼图之间的数据关联（详细比较复杂）     
void setPieChart(JFreeChart pieChart)   每个饼图的显示方式（见JFreeChart类个PiePlot类）

-----------------------------------------------------------------------------------------------------------

AbstractRenderer类：  
void setItemLabelAnchorOffset(double offset)     数据标签的与数据点的偏移  
void setItemLabelsVisible(boolean visible)     数据标签是否可见  
void setItemLabelFont(Font font)      数据标签的字体  
void setItemLabelPaint(Paint paint)      数据标签的字体颜色  
void setItemLabelPosition(ItemLabelPosition position)    数据标签位置  
void setPositiveItemLabelPosition(ItemLabelPosition position)   正数标签位置  
void setNegativeItemLabelPosition(ItemLabelPosition position)   负数标签位置  
void setOutLinePaint(Paint paint)      图形边框的线条颜色  
void setOutLineStroke(Stroke stroke)      图形边框的线条笔触  
void setPaint(Paint paint)       所有分类图形的颜色  
void setShape(Shape shape)       所有分类图形的形状（如折线图的点）  
void setStroke(Stroke stroke)       所有分类图形的笔触（如折线图的线）  
void setSeriesItemLabelsVisible(int series,boolean visible)   指定分类的数据标签是否可见  
void setSeriesItemLabelFont(int series,Font font)    指定分类的数据标签的字体  
void setSeriesItemLabelPaint(int series,Paint paint)    指定分类的数据标签的字体颜色  
void setSeriesItemLabelPosition(int series,ItemLabelPosition position)  数据标签位置  
void setSeriesPositiveItemLabelPosition(int series,ItemLabelPosition position) 正数标签位置  
void setSeriesNegativeItemLabelPosition(int series,ItemLabelPosition position) 负数标签位置  
void setSeriesOutLinePaint(int series,Paint paint)    指定分类的图形边框的线条颜色  
void setSeriesOutLineStroke(int series,Stroke stroke)    指定分类的图形边框的线条笔触  
void setSeriesPaint(int series,Paint paint)     指定分类图形的颜色  
void setSeriesShape(int series,Shape shape)     指定分类图形的形状（如折线图的点）  
void setSeriesStroke(int series,Stroke stroke)     指定分类图形的笔触（如折线图的线）

AbstractCategoryItemRenderer(AbstractRenderer)类：  
void setLabelGenerator(CategoryLabelGenerator generator)   数据标签的格式  
void setToolTipGenerator(CategoryToolTipGenerator generator)   MAP中鼠标移上的显示格式  
void setItemURLGenerator(CategoryURLGenerator generator)   MAP中钻取链接格式  
void setSeriesLabelGenerator(int series,CategoryLabelGenerator generator) 指定分类的数据标签的格式  
void setSeriesToolTipGenerator(int series,CategoryToolTipGenerator generator) 指定分类的MAP中鼠标移上的显示格式  
void setSeriesItemURLGenerator(int series,CategoryURLGenerator generator) 指定分类的MAP中钻取链接格式

BarRenderer(AbstractCategoryItemRenderer)类：  
void setDrawBarOutline(boolean draw)      是否画图形边框  
void setItemMargin(double percent)      每个BAR之间的间隔  
void setMaxBarWidth(double percent)      每个BAR的最大宽度  
void setMinimumBarLength(double min)      最短的BAR长度，避免数值太小而显示不出  
void setPositiveItemLabelPositionFallback(ItemLabelPosition position)  无法在BAR中显示的正数标签位置  
void setNegativeItemLabelPositionFallback(ItemLabelPosition position)  无法在BAR中显示的负数标签位置

BarRenderer3D(BarRenderer)类：  
void setWallPaint(Paint paint)    3D坐标轴的墙体颜色

StackedBarRenderer(BarRenderer)类：  
没有特殊的设置

StackedBarRenderer3D(BarRenderer3D)类：  
没有特殊的设置

GroupedStackedBarRenderer(StackedBarRenderer)类：  
void setSeriesToGroupMap(KeyToGroupMap map)  将分类自由的映射成若干个组（KeyToGroupMap.mapKeyToGroup(series,group)）

LayeredBarRenderer(BarRenderer)类：  
void setSeriesBarWidth(int series,double width)  设定每个分类的宽度（注意设置不要使某分类被覆盖）

WaterfallBarRenderer(BarRenderer)类：  
void setFirstBarPaint(Paint paint)   第一个柱图的颜色  
void setLastBarPaint(Paint paint)   最后一个柱图的颜色  
void setPositiveBarPaint(Paint paint)   正值柱图的颜色  
void setNegativeBarPaint(Paint paint)   负值柱图的颜色

IntervalBarRenderer(BarRenderer)类：  
需要传IntervalCategoryDataset作为数据源

GanttBarRenderer(IntervalBarRenderer)类：  
void setCompletePaint(Paint paint)   完成进度颜色  
void setIncompletePaint(Paint paint)   未完成进度颜色  
void setStartPercent(double percent)   设置进度条在整条中的起始位置（0.0～1.0）  
void setEndPercent(double percent)   设置进度条在整条中的结束位置（0.0～1.0）

StatisticBarRenderer(BarRenderer)类：  
需要传StatisticCategoryDataset作为数据源

LineAndShapeRenderer(AbstractCategoryItemRenderer)类：  
void setDrawLines(boolean draw)    是否折线的数据点之间用线连  
void setDrawShapes(boolean draw)   是否折线的数据点根据分类使用不同的形状  
void setShapesFilled(boolean filled)   所有分类是否填充数据点图形  
void setSeriesShapesFilled(int series,boolean filled) 指定分类是否填充数据点图形  
void setUseFillPaintForShapeOutline(boolean use) 指定是否填充数据点的Paint也被用于画数据点形状的边框

LevelRenderer(AbstractCategoryItemRenderer)类：  
void setItemMargin(double percent)   每个分类之间的间隔  
void setMaxItemWidth(double percent)   每个分类的最大宽度

CategoryStepRenderer(AbstractCategoryItemRenderer)类：  
void setStagger(boolean shouldStagger)   不同分类的图是否交错

MinMaxCategoryRenderer(AbstractCategoryItemRenderer)类：  
void setDrawLines(boolean drawLines)   是否在每个分类线间画连接线  
void setGroupPaint(Paint groupPaint)   一组图形连接线的颜色  
void setGroupStroke(Stroke groupStroke)   一组图形连接线的笔触  
void setMaxIcon(Icon maxIcon)    最大值的ICON  
void setMinIcon(Icon minIcon)    最小值的ICON  
void setObjectIcon(Icon objectIcon)   所有值的ICON

AreaRender(AbstractCategoryItemRenderer)类：  
没有特殊的设置

StackedAreaRender(AreaRender)类：  
没有特殊的设置