JFREE CHART

chart java chart library

tutorialspoint

SIMPLYEASYLEARNING

www.tutorialspoint.com

https://www.facebook.com/tutorialspointindia

https://twitter.com/tutorialspoint

About the Tutorial

This tutorial describes various ways to incorporate **JFreeChart** in Java-based standalone and web-based applications. The tutorial is categorized into various chapters to provide a comprehensive and easy understanding of JFreeChart programming with Java applications.

Audience

This reference has been prepared for beginners to help them understand the basic-to-advanced concepts related to JFreeChart library.

Prerequisites

Before you start practicing the examples given in the tutorial, it is expected that you have a basic understanding of Java programming with JDK1.6 or later versions, Swing, file management and database management in Java.

Copyright & Disclaimer

We strive to update the contents of our website and tutorials as timely and as precisely as possibly; however, the contents may contain inaccuracies or errors. Tutorials Point (I) provides no guarantee regarding the accuracy, timeliness, or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

1. JFREECHART – OVERVIEW

A chart is a graphical representation of information. There are various tools available, which can be used to create different types of charts. The **JFreeChart** project was founded in February 2000, by David Gilbert. Today, it is the most widely used charting library among Java developers.

This tutorial will help you understand what exactly JFreeChart is, why is it required, and the various ways to create different types of charts within a Java-based application or independently.

What is JFreeChart?

JfreeChart is an open source library developed in Java. It can be used within Java based applications to create a wide range of charts. By using JFreeChart, we can create all the major type of 2D and 3D charts such as pie chart, bar chart, line chart, XY chart and 3D charts.

Why JFreeChart?

JFreeChart is open source and 100% free, which permits usage in the commercial applications without any cost. We have enlisted here some more points in favor of why you should use JFreeChart:

- It comes with well documented APIs, which makes it quite easy to understand.
- It supports a wide range of chart types such as Pie Chart, Line Chart, Bar Chart, Area Chart and 3D charts.
- JFreeChart is easy to extend and can be used in both, the client-side, as well as the server-side applications.
- It supports multiple output formats like PNG, JPEG, PDF, SVG etc.
- It allows extensive customizations of charts.

Consider a situation where you are developing an application and you need to show the data in the form of charts, and the data itself is populated dynamically. In such case, displaying the data in the form of charts using JFreeChart programming is very simple.

2. JFREECHART – INSTALLATION

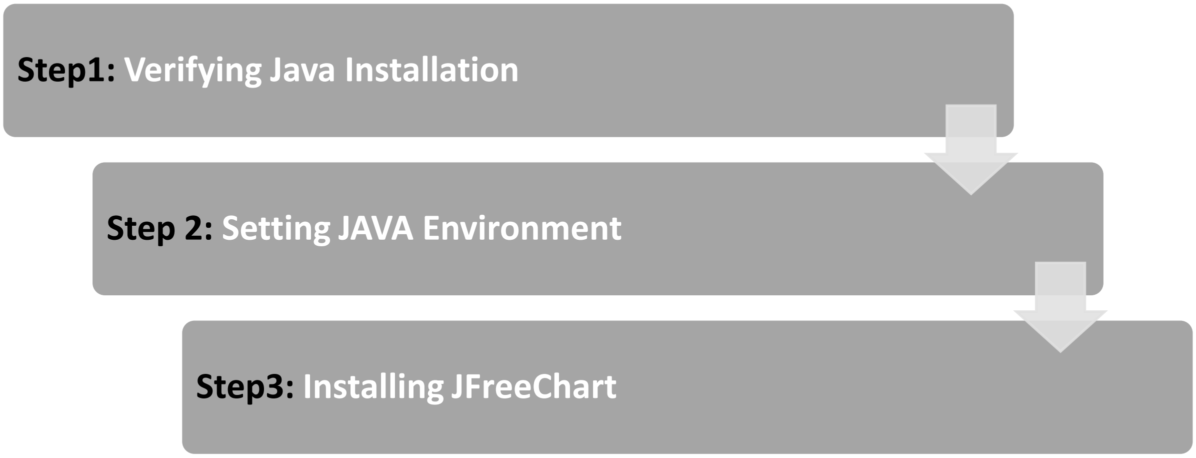
JFreeChart is popular for its efficient chart creation and user-friendly installation setup. This chapter describes the process of setting up JFreeChart on Windows and Linux. User administration is needed while installing JFreeChart.

System Requirements

JDK	1.5 or above
Memory	2GB RAM
Disk Space	No minimum requirement
Operating System Version	Linux or Windows

Installing JFreeChart

To install JFreeChart, there are three following steps viz…



Step1: Verifying Java Installation

Step 2: Setting JAVA Environment

Step3: Installing JFreeChart

Step 1: Verifying Java Installation

To verify Java installation, open the console and execute the following **java** command:

OS	Task	Command
Windows	Open command console	C:>java -version
Linux	Open command terminal	$java -version

Once Java installation is done properly, then you should get the following output for both the operating systems:

OS	Description
Windows	Java version "1.7.0_60" Java (TM) SE Run Time Environment (build 1.7.0_60-b19) Java HotSpot(TM) 64-bit Server VM (build 24.60-b09,mixed mode)
Linux	java version "1.7.0_25" OpenJDK Runtime Environment (rhel-2.3.10.4.el6_4-x86_64) OpenJDK 64-Bit Server VM (build 23.7-b01, mixed mode)

If you do not have Java installed, then install the Java Software Development Kit (SDK) from the link:

http://www.oracle.com/technetwork/java/javase/downloads/index.html

We assume that you have installed Java 1.7.0_60 version before proceeding for this tutorial.

Step 2: Setting JAVA Environment

Set the **JAVA_HOME** environment variable to point to the base directory location where Java is installed on your machine. For example,

OS	Description
Windows	Set Environmental variable JAVA_HOME to C:\ProgramFiles\java\jdk1.7.0_60
Linux	export JAVA_HOME=/usr/local/java-current

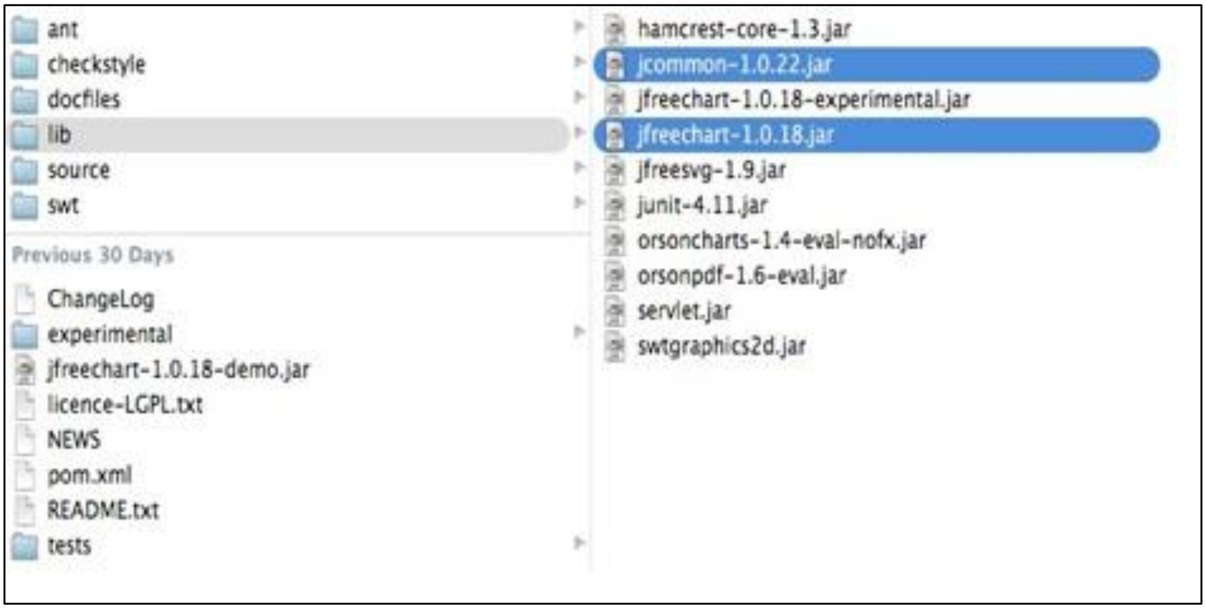
Append Java compiler location to System Path.

OS	Description
Windows	Append the String; C:\Program Files\Java\jdk1.7.0_60\bin to the end of the system variable PATH.
Linux	export PATH=$PATH:$JAVA_HOME/bin/

Verify the command **java -version** from the command prompted as explained above.

Step 3: Installing JFreeChart

Download the latest version of JFreeChart.zip from the link **http://www.jfree.org/jfreechart/download/.** Unzip the downloaded file at any location from where required libraries can be linked into your Java program. The following image shows the structure of the directories and files:



ant
checkstyle
docfiles
lib
source
swt
Previous 30 Days
ChangeLog
experimental
jfreechart-1.0.18-demo.jar
licence-LGPL.txt
NEWS
pom.xml
README.txt
tests

hamcrest-core-1.3.jar
jcommon-1.0.22.jar
jfreechart-1.0.18-experimental.jar
jfreechart-1.0.18.jar
jfreesvg-1.9.jar
junit-4.11.jar
orsoncharts-1.4-eval-nofx.jar
orsonpdf-1.6-eval.jar
servlet.jar
swtgraphics2d.jar

Add complete path of jfreechart-1.0.18.jar and jcommon-1.0.22.jar files to the CLASSPATH as shown below:

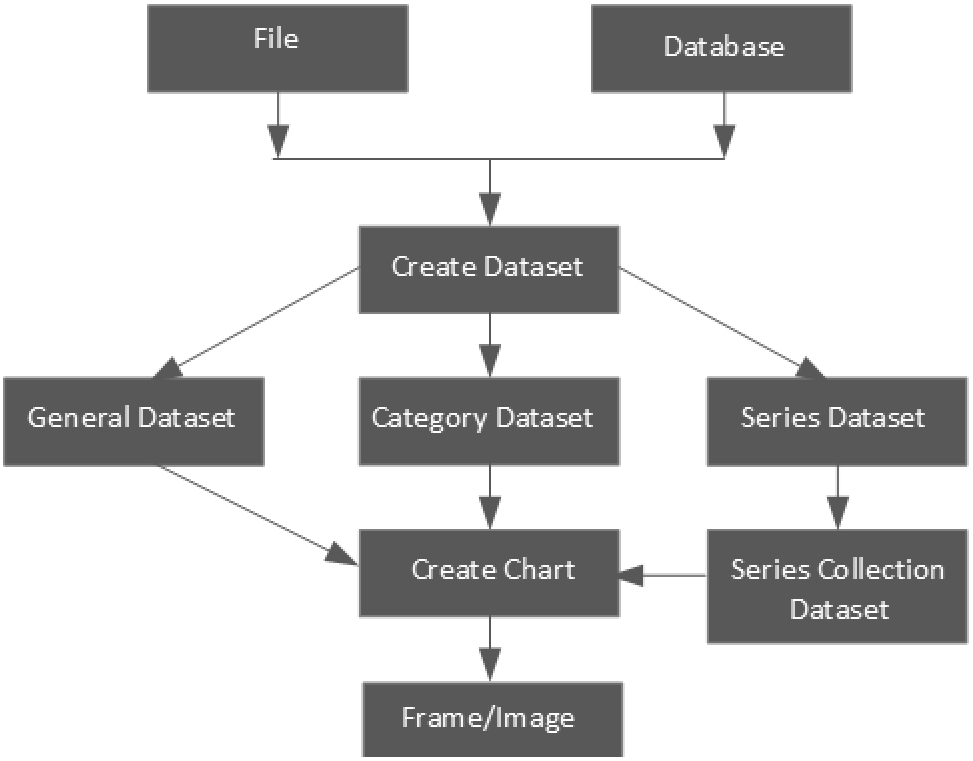
OS	Description
Windows	Append the Strings “C:\ jfreechart-1.0.18\lib\ jfreechart-1.0.18.jar” and “C:\ jfreechart-1.0.18\lib\ jcommon-1.0.22.jar” to the end of the user variable CLASSPATH
Linux	Export CLASSPATH=$CLASSPATH: /usr/share/jfreechart-1.0.18/lib/jfreechart-1.0.18.jar: /usr/share/jfreechart-1.0.18/lib/jcommon-1.0.22.jar

3. JFREECHART – ARCHITECTURE

This chapter explains basic class level and application level architectures of JFreeChart to give you an idea about how JFreeChart interacts with different classes and how it fits in your Java based application.

Class Level Architecture

The class level architecture explains how various classes from the library interact with each other to create various types of charts.



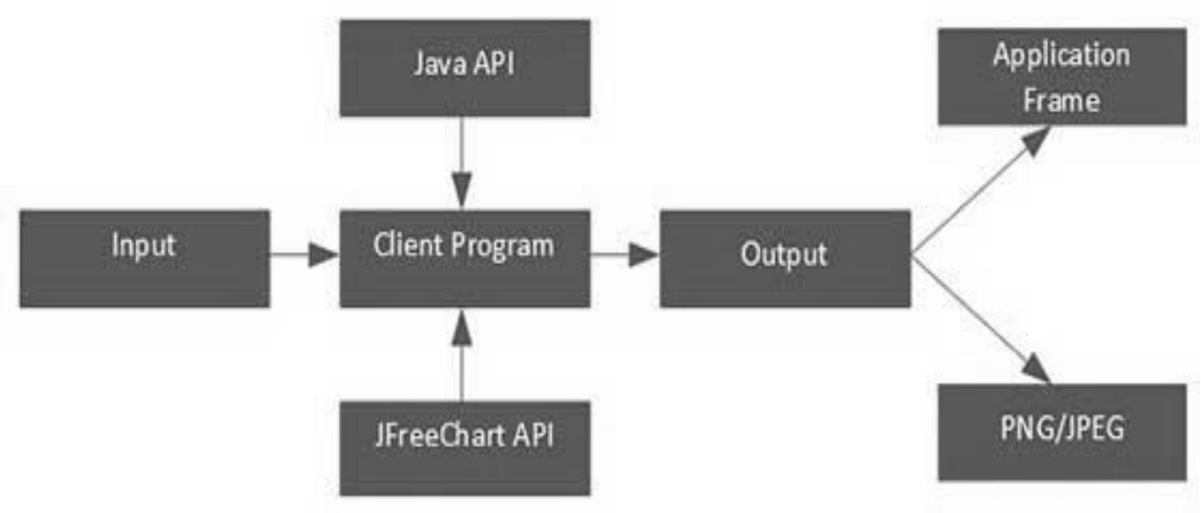
Following is the detail of the units used in the above block diagram:

Units	Description
File	The source having user input to be used for creating a dataset in the file.
Database	The source having user input to be used for creating a dataset in the database.

Create Dataset	Accepts the dataset and stores the dataset into dataset object.
General Dataset	This type of dataset is mainly used for pie charts.
Category Dataset	This type of dataset is used for bar chart, line chart, etc.
Series Dataset	This type of dataset is used for storing series of data and construct line charts.
Series Collection Dataset	The different categories of series datasets are added to series collection dataset. This type of dataset is used for XYLine Charts.
Create Chart	This is the method, which is executed to create final chart.
Frame/Image	The chart is displayed on a Swing Frame or an image is created.

Application Level Architecture

The application level architecture explains where JFreeChart library sits inside a Java Application.



The client program receives user data and then it uses standard Java and JFreeChart APIs based on requirements to generate the output in the form of either a frame, which can be displayed directly inside the application or independently in the image formats such as JPEG or PNG.

4. JFREECHART – REFERENCED APIS

In this chapter, we will discuss about some of the important packages, classes, and methods from JFreeChart library. These packages, classes, and methods are the most frequently used while creating a variety of charts using JFreeChart library.

ChartFactory Class

ChartFactory is an abstract class under the **org.jfree.chart** package. It provides a collection of utility methods for generating standard charts.

Following is a list of few of the important methods:

Class Constructor

S.N.	Description
1	**ChartFactory()** Default constructor of ChartFactory class.

Class Methods

S.N.	Methods & Description
1	**createPieChart(java.lang.String title, PieDataset dataset, boolean legend, boolean tooltips, boolean urls)** This method creates a pie chart with default settings. It returns JfreeChart type object.
2	**createPieChart3D(java.lang.String title, PieDataset dataset, boolean legend, boolean tooltips, boolean urls** This method creates a 3D pie chart using the specified dataset.

3	**createBarChart(java.lang.String title, java.lang.String categoryAxisLabel, java.lang.String valueAxisLabel, CategoryDataset dataset, PlotOrientation orientation, boolean legend, boolean tooltips, boolean urls)** The argument java.lang.String categoryAxisLabel is the label for values placed on X-axis. The argument java.lang.String valueAxisLabel is the label for values placed on Y-axis. This method creates a bar chart.
4	**createBarChart3D(java.lang.String title, java.lang.String categoryAxisLabel, java.lang.String valueAxisLabel, CategoryDataset dataset, PlotOrientation orientation, boolean legend, boolean tooltips, boolean urls)** This Method Creates a bar chart with a 3D effect. It returns JfreeChart type object.
5	**createLineChart(java.lang.String title, java.lang.String categoryAxisLabel, java.lang.String valueAxisLabel, CategoryDataset dataset, PlotOrientation orientation, boolean legend, boolean tooltips, boolean urls)** This method creates a line chart with default settings.
6	**createLineChart3D(java.lang.String title, java.lang.String categoryAxisLabel, java.lang.String valueAxisLabel, CategoryDataset dataset, PlotOrientation orientation, boolean legend, boolean tooltips, boolean urls)** This method creates a line chart with 3D effect.
7	**createXYLineChart(java.lang.String title, java.lang.String xAxisLabel, java.lang.String yAxisLabel, XYDataset dataset, PlotOrientation orientation, boolean legend, boolean tooltips, boolean urls)** This method creates a line chart based on XYDataset with default settings.

ChartFrame Class

ChartFrame class under the **org.jfree.chart** package, provides all frame related functions and utilities. ChartFrame class inherits functionalities from parent classes such as Frame, Window, Container, and Component classes.

Class Constructor

S.N.	Constructor and Description
1	**ChartFrame (java.lang.Frame String, JfreeChart chart)** It constructs a frame.
2	**Chart Frame (java.lang.Frame String, JfreeChart chart, boolean scrollpane)** It constructs a frame.

Class Method

S.N.	Method and Description
1	**getChartPanel()** This method returns the chart panel for a frame.

ChartPanel Class

ChartPanel class from the **org.jfree.chart** package is used as a swing GUI component for displaying JfreeChart object.

Class Constructor

S.N.	Constructor and Description
1	**ChartPanel(JFreeChart chart)** This constructor constructs a panel that displays the specified chart.
2	**ChartPanel(JFreeChart chart, boolean useBuffer)** This constructor constructs a panel containing a chart.
3	**ChartPanel(JFreeChart chart, boolean properties, boolean save, boolean print, boolean zoom, boolean tooltips)** This constructor constructs a JFreeChart panel.

Class Method

S.N.	Method and Description
1	**setPreferredSize(java.awt.Dimension)** This method is used to set the frame size using java.awt. Dimension class object as an argument. This method is taken from javax.swing.JComponent.

ChartUtilities Class

CharUtilites class from the **org.jfree.chart** package provides a collection of utility methods of JFreeCharts including methods for converting charts into image file format such as PNG, JPEG, and creating HTML image maps.

Class Constructor

S.N.	Constructor and Description
1	**ChartUtilities()** This is a default constructor of a class

Class Method

S.N.	Method and Description
1	**saveChartAsPNG(java.io.File file, JfreeChart chart, int width, int height)** This method converts and saves a chart to the specified file in PNG format.
2	**saveChartAsJPEG(java.io.File file, JfreeChart chart, int width, int height)** This method converts and saves a chart to the specified file in JPEG format.

JFreeChart Class

JFreeChart class is the core class under the **org.jfree.chart** package. This class provides JFreeChart method to create bar charts, line charts, pie charts, and xy plots including time series data.

Class Constructor

S.N.	Constructor and Description
1	**JfreeChart(Plot plot)** This constructor creates a new chart based on the supplied plot.
2	**JfreeChart(java.lang.String title, java.awt.Font titleFont, Plot plot, boolean createLegend)** This constructor creates a new chart with the given title and plot.
3	**JfreeChart(java.lang.String title, Plot plot)** This constructor creates a new chart with the given title and plot.

Class Method

S.N.	Method and Description
1	**getXYPlot()** This method Returns the plot chart as **XYPlot**. Using XYPolt, we can do some utility operations on xy charts.

PiePlot Class

This class is a part of **org.jfree.chart.plot** package and extends Plot class from the same package. This class provides methods to create Pie Plots.

Class Constructor

S.N.	Constructor and Description

1	**PiePlot()** It creates a new plot.
2	**PiePlot(PieDataset dataset)** It creates a plot that draws a pie chart for the specified dataset.

Class Method

S.N.	Method and Description
1	**setStartAngle(double angle)** This Method sets the starting angle and sends a PlotChangeEvent to all registered listeners

PiePlot3D Class

PiePlot3D class is a subclass of PiePlot class under the same package. Hence, this class has the same features as PiePlot class, except it is used to create 3D plots.

Class Constructor

S.N.	Constructor and Description
1	**PiePlot3D()** This constructor creates a new instance with no dataset.
2	**PiePlot3D(PieDataset dataset)** This constructor creates a pie chart with three dimensional effect using a specified dataset.

Class Method

S.N.	Method and Description
1	**setForegroundAlpha(float alpha)** It sets the alpha-transparency for the plot and sends a PlotChangeEvent to all registered listeners. This is taken from one of the parent Plot classes.

2	**setInteriorGap(double percent)** It sets the interior gap and sends a PlotChangeEvent to all registered listeners. This controls the space between the edges of the pie plot and the plot area itself (i. e., the region where the section labels appear). This method is taken from the parent class PiePlot.

PlotOrientation Class

This is a serialized class available in **org.jfree.chart.plot** package and it is used to show the orientation of a 2D plot. The orientation can either be **vertical** or **horizontal**. It sets the orientation of Y-axis. A conventional plot has a vertical Y- axis.

Field summary

S.N.	Type	Field & Description
1	PlotOrientation	**HORIZONTAL** For a plot where the range axis(Y-axis) is horizontal.
2	PlotOrientation	**VERTICAL** For a plot where the range axis(Y-axis) is vertical. This is the default orientation.

Class Method

S.N.	Method and Description
1	**isHorizontal()** This method returns true if this orientation is HORIZONTAL, and false otherwise.
2	**isVertical()** This Method returns true if this orientation is VERTICAL, and false otherwise.

XYPlot Class

This is a general class available in **org.jfree.chart.plot** package and it is used for the plotting data in the form of (x,y) pairs. This plot can use data from any other class that implements the XYDataSet Interface. XYPlot makes use of a XYItemRenderer to draw each point on the plot.

Class Constructor

S.N.	Constructor and Description
1	**XYPlot()** This contractor creates a new XYPlot instance with no dataset, no axes and no renderer.
2	**XYPlot(XYDataset** dataset, **ValueAxis** domainAxis, **ValueAxis** rangeAxis, **XYItemRenderer** renderer) This constructor creates a new plot with the specified dataset, axis, and renderer.

Class Method

S.N.	Method and Description
1	**setRenderer(XYItemRenderer** renderer) This method sets the renderer for the primary dataset and sends a change event to all registered listeners.

NumberAxis Class

This class is available in **org.jfree.chart.axis** package and it can access the numerical data of any axis. When we set the range of any axis to default, it fits according to the range of the data. But using NumberAxis, class we can set the lower margin and upper margin of domain and range axes.

Class Constructor

S.N.	Constructor and Description
1	**NumberAxis( )** This is a default Constructor of NumberAxis.
2	**NumberAxis(java.lang.String label)** The constructor NumberAxis uses default values where necessary.

Class Method

S.N.	Method and Description
1	**setLowerMargin(double margin)** It sets the lower margin for the axis (as a percentage of the axis range) and sends an **AxisChangeEvent** to all registered listeners. This method is taken from parent class of the class ValueAxis.
2	**setUpperMargin(double margin)** It sets the upper margin for the axis (as a percentage of the axis range) and sends an **AxisChangeEvent** to all registered listeners. This method is also present in ValueAxis Class.

XYLineAndShapeRenderer Class

This is the class, available under **org.jfree.chart.renderer.xy** package, which takes care of connecting data points with lines and draws shapes at each data point. This renderer class is designed for use with the **XYPlot** class.

Class Constructor

S.N.	Constructor and Description
1	**XYLineAndShapeRenderer()** It creates a new renderer with both lines and shapes visible.
2	**XYLineAndShapeRenderer** (boolean lines, boolean shapes) It creates a new renderer with specific property.

Class Method

S.N.	Method and Description
1	**setSeriesPaint(int series, java.awt.Paint paint)** This method sets the paint used for a series and sends a **RendererChangeEvent** to all registered listeners. This method is taken from AbstratRenderer abstract class from renderer package in JFreeChart API.

2	**setSeriesStroke(int series, java.awt.Stroke stroke)** This method Sets the stroke used for a series and sends a **RendererChangeEvent** to all registered listeners. This method is taken from AbstratRenderer abstract class, which is super class of this package.

XYItemRenderer general datasets

This is an interface for rendering the format of a single (x, y) item on a XYPlot. The package is **org.Jfree.data.general**, which has classes and interfaces to define different types of datasets to construct charts.

PieDataset

This is an interface used as a general purpose dataset, where values are associated with keys. As the name suggests, you can use this dataset to supply data for pie charts. This interface extends KeyedValues and DataSet interfaces. All the methods used for this interface are taken from KeyedValues, Values, and Dataset interfaces.

DefaultPieDataset Class

This is a Default implementation class of a PieDataset interface.

Class Constructor

S.N.	Constructor and Description
1	**DefaultPieDataset()** This constructor creates a new dataset, initially empty.
2	**DefaultPieDataset(KeyedValues data)** It creates a new dataset by copying data from a **KeyedValues** instance.

Class Method

S.N.	Method and Description
1	**setValue(java.lang.Comparable key, double value)** It sets the data value for a key and sends a **DatasetChangeEvent** to all registered listeners.

2	**setValue(java.lang.Comparable key, java.lang.Number value)** It sets the data value for a key and sends a **DatasetChangeEvent** to all registered listeners.

SeriesException Class

This is an exception class. It raises an exception occurred in the time series of data in the dataset. Exceptions are raised on the occurrence of duplicate or invalid data. The time series must not be applied with duplicates and the format must be valid.

DefaultCategoryDataset

This is a default implementation class of CategoryDataset interface.

Class Constructor

S.N.	Constructor and Description
1	**DefaultCategoryDataset()** This constructor creates new empty dataset.

Class Method

S.N.	Method and Description
1	**addValue(double value, java.lang.Comparable rowKey, java.lang.Comparable columnKey)** This method adds a value to the table using comparable keys.
2	**addValue(java.lang.Number value, java.lang.Comparable rowKey, java.lang.Comparable columnKey)** This method adds a value to the table.
3	**setValue(double value, java.lang.Comparable rowKey, java.lang.Comparable columnKey)** This method adds or updates a value in the table and sends a **DatasetChangeEvent** to all registered listeners.

4	**setValue(java.lang.Number value, java.lang.Comparable rowKey, java.lang.Comparable columnKey)** This method adds or updates a value in the table and sends a **DatasetChangeEvent** to all registered listeners.

Refer JFreeChart API for more information on various other methods and fields.

Series Datasets

The series dataset is used by XY charts. The package is **org.Jfree.data.xy**, which contains classes and interfaces belonging to xy charts. The core interface is XYDataset.

XYDataset

This is an interface through which data in the form of (x,y) items can be accessed. As the name suggests, you can use this dataset to serve XY chart. Some of the methods in this interface are taken from SeriesDateset interface.

XYZDataset

This is an interface through which data in the form of (x,y,z) items can be accessed. As the name suggests, you can use this dataset to serve XYZ chart. Some of the methods in this interface are taken from SeriesDateset.

XYSeries

This is a class, which represents a sequence of zero or more data items in the form (x, y). By default, the items in the series are sorted into ascending order by x-value, and duplicate x-values are permitted. Both the sorting and duplicate defaults can be changed in the constructor. Y-values can be denoted as null to represent missing values.

Class Constructor

S.N.	Constructor and Description
1	**XYSeries(java.lang.Comparable key)** This constructor creates a new empty series.
2	**XYSeries(java.lang.Comparable key, boolean autoSort)** It constructs a new empty series, with the auto-sort flag set as requested, and duplicate values are allowed.

3	**XYSeries(java.lang.Comparable key, boolean autoSort, boolean allowDuplicateXValues)** It constructs a new xy-series that contains no data.

Class Method

S.N.	Method and Description
1	**add(double x, double y)** This method adds data item into the series.

The above method is used in the tutorial example. If you want to learn the remaining methods and fields, please refer JFreeChart API.

XYSeriesCollection

XYSeriesCollection class has parent classes like AbstractIntervelDataset, AbstractXYDatset, AbstractSeriesDataset and AbstractDataset. Some of the methods in this class belong to parent classes of this class.

Class Constructor

S.N.	Constructor and Description
1	**XYSeriesCollection()** It constructs an empty dataset.
2	**XYSeriesCollection(XYSeries xyseries)** It constructs a dataset and populates it with a single series.

Class Method

S.N.	Method and Description
1	**addSeries(XYSeries series)** This method adds a series to the collection and sends a **DatasetChangeEvent** to all registered listeners.

Refer JFreeChart API for the remaining methods and fields.

Default XYZDataset

DefaultXYZDataset class have parent classes like AbstractIntervelDataset, AbstractXYDatset, AbstractSeriesDataset, AbstractDataset and AbstractXYZDataset. Some of the methods in this class belong to parent classes of this class.

Class Constructor

S.N.	Constructor and Description
1	**DefaultXYZDataset()** It constructs an empty dataset.

Class Method

S.N.	Method and Description
1	**addSeries(java.lang.Comparable seriesKey, double[ ][ ] data)** This method adds a series to the collection and sends a **DatasetChangeEvent** to all registered listeners.

Please refer JFreeChart API for the remaining methods and fields.

Time Series in JFreeCharts

The package is **org.jfree.data.time**. This package contains classes and interfaces, which are used for the time related data.

TimeSeries

This class represents a sequence of data items in the form of period values, where period is some instance of RegularTimePeriod abstract class such as Time, Day, Hour, Minute, and Second classes.

Class Constructor

S.N.	Constructor and Description

1	**TimeSeries(java.lang.Comparable name)** It creates new empty series.
2	**TimeSeries(java.lang.Comarable name, java.lang.String domain, java.lang.Strin range)** It creates new time series that contains no data.

Class Method

S.N.	Method and Description
1	**add(RegularTimePeriod period,double value)** This method adds a new data item to the series.

Refer JFreeChart API for the remaining methods and fields.

TimeSeriesCollection

This is a class used as a collection of time series objects. This class implements the XYDataset interface, as well as it extends IntervelXYDataset interface. This makes it convenient to collect series data objects.

Class Constructor

S.N.	Constructor and Description
1	**TimeSeriesCollection()** It constructs an empty dataset, tied to the default time zone.
2	**TimeSeriesCollection(TimeSeries** series) It constructs a dataset containing a single series (more can be added), tied to the default time zone.
3	**TimeSeriesCollection(TimeSeries** series, java.util.TimeZone zone) It constructs a dataset containing a single series (more can be added), tied to a specific time zone.
4	**TimeSeriesCollection**(java.util.TimeZone zone) It constructs an empty dataset, tied to a specific time zone.

Class Method

S.N.	Method and Description
1	**addSeries(TimeSeries** series) This method adds a series to the collection and sends a **DatasetChangeEvent** to all registered listeners.

Please refer JFreeChart API for the remaining methods and fields.

Second

This class represents a second in a particular day. This class is immutable, which is a requirement for all RegularTimePeriod subclass.

Class Constructor

S.N.	Constructor and Description
1	**Second()** It constructs a new Second, based on the system date/time.
2	**Second(java.util.Date time)** It constructs a new instance from the specified date/time and the default time zone.
3	**Second(java.util.Date time, java.util.TimeZone zone, java.util.Locale locale)** It creates a new second based on the supplied time and time zone.
4	**Second(int second, int minute, int hour, int day, int month, int year** It creates a new second.
5	**Second(int second, Minute minute)** It constructs a new Second.

Class Method

S.N.	Method and Description

1	**getSecond()** It returns the second within the minute.
2	**next()** It returns the second following the present second.

Please refer JFreeChart API for the remaining methods and fields.

Frames in JFreeCharts

The package is **org.jfree.ui.** This is the package belongs to JCommons API of JFreeChart. It contains utility classes used for creating frames for pre-configured charts.

ApplicationFrame

This is a base class for creating the main frame for simple applications. The frame listens for window closing events, and responds by shutting down the JVM. This is fine for small demo applications. For enterprise applications, you need to use something more robust. The main core methods in this class are taken from Component, Container, Window, Frame, and Jframe classes.

Class Constructor

S.N.	Constructor and Description
1	**ApplicationFrame(java.lang.String title)** It creates an application frame with the string title.

This class helps to create AWT Frames. This is the reason for why we use this class as super class in this tutorial examples.

The methods, which are taken form the parent classes are used for opening a frame, closing a frame, changing the size, changing the background or foreground color, and listeners.

RefineryUtilities

This is a class collection of utility methods relating to user interface.

Class Method

S.N.	Method and Description
1	**centerFrameOnScreen(java.awt.Window frame)** It positions the specified frame in the middle of the screen.

The above method is used in the tutorial example. Refer JFreeChart API for remaining methods and fields.