

JFreeChart



JfreeChart

- É comum nos depararmos com necessidades de geração de relatórios que, frequentemente traz informações na forma de gráficos, entre outras.
- E, como qualquer outra necessidade comum a desenvolvedores Java, já existe uma biblioteca capaz de lidar com dados de uma forma muito mais simples do que utilizando diretamente a API do Java.
- O JFreeChart é hoje a biblioteca mais famosa para desenho de gráficos.
- É um projeto de software livre iniciado em 2000 e que tem ampla aceitação pelo mercado, funcionando até em ambientes Java 1.3.

JfreeChart

- Além do fato de ser livre, possui a vantagem de ser bastante robusta e flexível.
- É possível usá-la para desenhar gráficos de pontos, de barra, de torta, de linha, gráficos financeiros, gantt charts, em 2D ou 3D e muitos outros.
- Consegue dar saída em JPG, PNG, SVG, EPS e até mesmo exibir em componentes Swing.
- Sua licença é LGPL o que permite ser usada em projetos de código fechado. O site oficial possui links para download, demos e documentação:

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



JfreeChart

- A classe principal é a `org.jfree.chart.JFreeChart`, ela representa um gráfico e pode assumir vários formatos (torta, barra, pontos, linhas etc).
- Trabalhar diretamente com essa classe é um pouco trabalhoso e a biblioteca optou por criar uma classe com uma série de métodos estáticos para facilitar o trabalho de criação dos gráficos: são métodos que se comportam como uma fábrica!
- Esses, também são um design pattern, chamado Factory method. A classe que tem tais métodos no JFreeChart é a `org.jfree.chart.ChartFactory`.



JfreeChart

- Queremos criar um gráfico de linha.
- Consultando o Javadoc da ChartFactory, descobrimos o método `createLineChart` que devolve um objeto do tipo `JFreeChart` prontinho.



JfreeChart

- Este método recebe alguns argumentos:
 - um título (String);
 - duas Strings com os labels dos eixos X e Y;
 - os dados a serem exibidos (do tipo CategoryDataset);
 - orientação da linha (PlotOrientation.VERTICAL ou HORIZONTAL);
 - um boolean indicando se queremos legenda ou não;
 - um boolean dizendo se queremos tooltips;
 - um boolean para exibir URLs no gráfico ou não.



JfreeChart

- O principal aqui é o `CategoryDataset`, justamente o conjunto de dados que queremos exibir no gráfico.
- `CategoryDataset` é uma interface e há várias implementações:

por exemplo, uma simples onde adicionamos os elementos manualmente, outra para trabalhar direto com banco de dados (JDBC) e outras.

- Vamos usar a `DefaultCategoryDataset`, a implementação padrão e mais fácil de usar:



JfreeChart

```
// cria o conjunto de dados
DefaultCategoryDataset ds = new DefaultCategoryDataset();
ds.addValue(40.5, "maximo", "dia 1");
ds.addValue(38.2, "maximo", "dia 2");
ds.addValue(37.3, "maximo", "dia 3");
ds.addValue(31.5, "maximo", "dia 4");
ds.addValue(35.7, "maximo", "dia 5");
ds.addValue(42.5, "maximo", "dia 6");
```

```
// cria o gráfico
JFreeChart grafico = ChartFactory.createLineChart("Meu Grafico",
"Dia", "Valor", ds, PlotOrientation.VERTICAL, true, true, false);
```

