Aula 24

JAR e Bibliotecas

Rafael Geraldeli Rossi

JAR

- Um arquivo no formato **JAR** (**J**ava **AR**chive) corresponde a um **arquivo compactado utilizado para distribuir uma conjunto de classes java** (geralmente um aplicativo ou uma biblioteca)
- Inclusive, é possível visualizar os arquivos que compõem um `.jar` por meio programas de (des)compactação padrão
- **No JAR estão disponíveis apenas classes compiladas e metadados associados que podem constituir um aplicativo ou uma biblioteca**

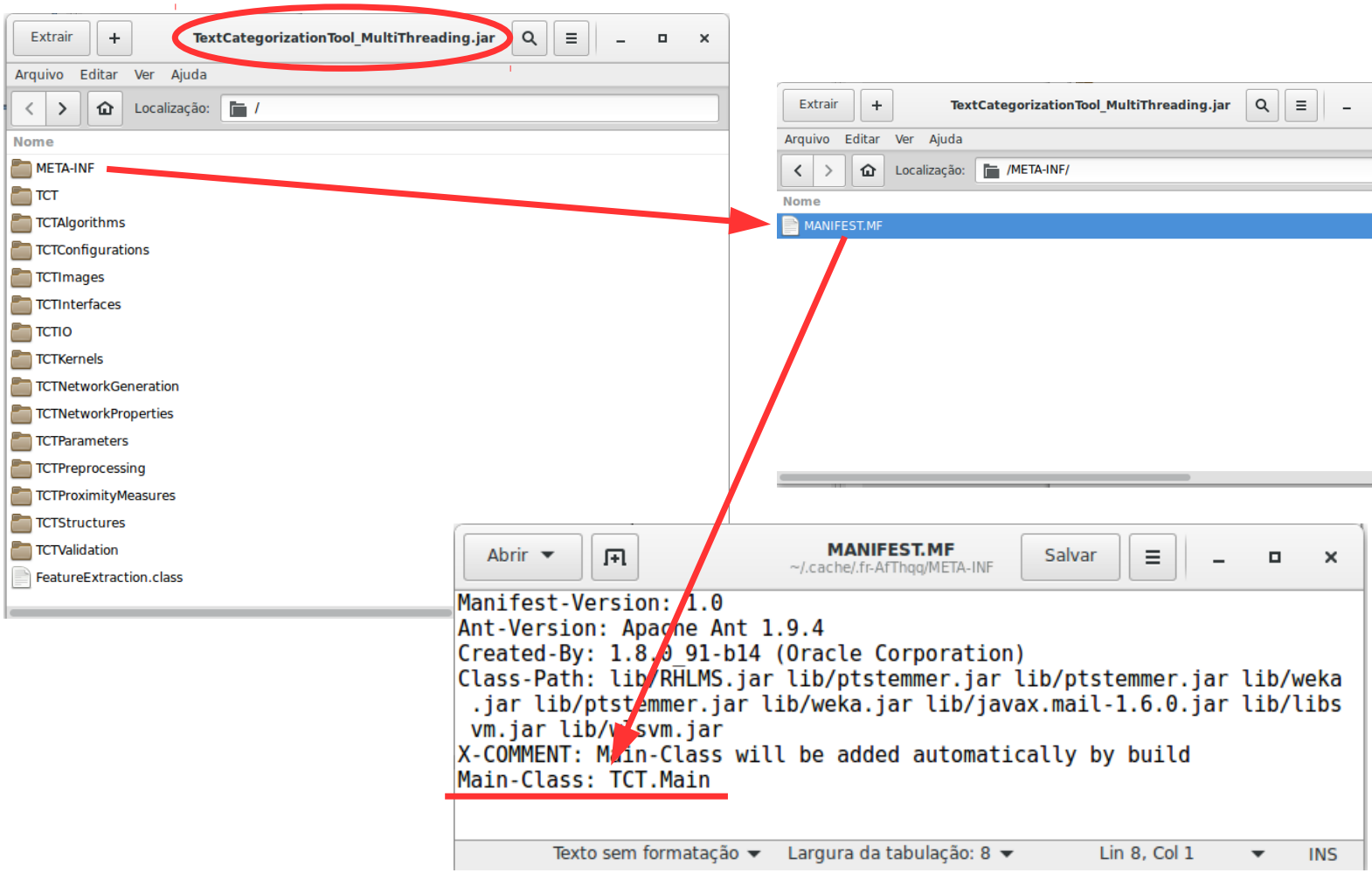
JAR

- Arquivos JAR podem ser criados e extraídos usando o utilitário "jar"do JDK
- As IDEs geralmente possuem utilitários para a criação do .jar
- Vale ressaltar que ferramentas de compressão (como o Winzip) também podem criar/extrair arquivos .jar

JAR

- Um arquivo JAR possui um arquivo manifesto localizado no caminho META-INF/MANIFEST.MF
- Arquivos JAR que têm a intenção de serem executáveis (como o *.exe do Windows) terão uma de suas classes especificadas como a classe "principal"
- O arquivo manifesto terá uma entrada como: `Main-Class:meusProgramas.MinhaClasse`

Exemplo de um JAR (criado no Netbeans)



Criando arquivos .jar via linha de comando

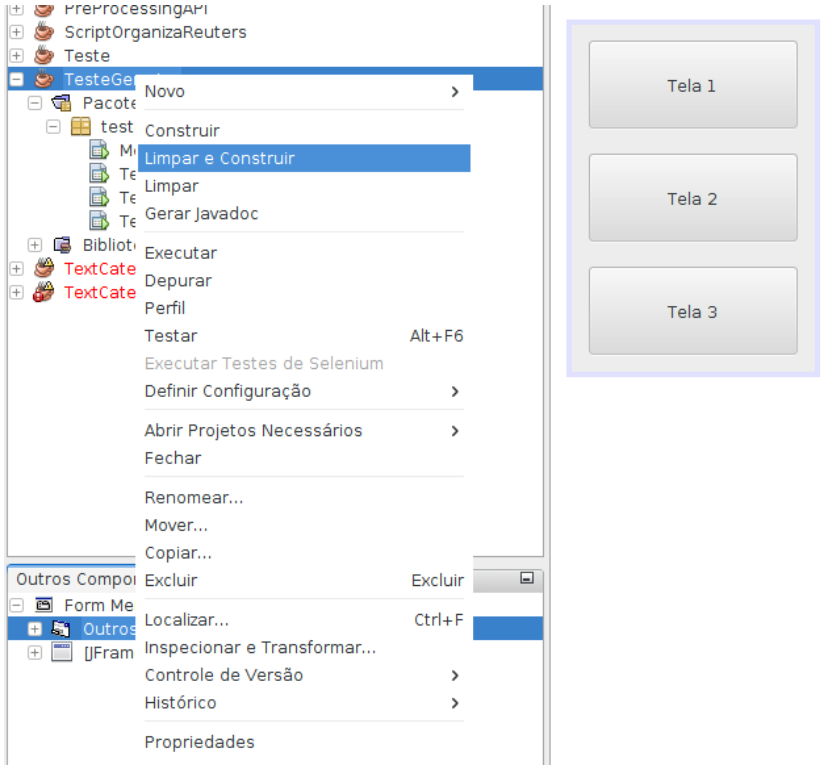
- O comando básico para se criar um .jar é
  `jar -cf [nome do jar] [arquivos de entrada]`
  Ex: `jar -cf programa.jar *.class`

- Com esse comando, um jar será gerado, porém, não foi definida uma classe principal que seria invocada ao executar o .jar

- Para especificar a classe principal deve-se executar o seguinte comando: `jar -cfe [nome do jar] [nome da classe principal] [arquivos de entrada]`
  Ex: `jar -cfe programa.jar classe1 *.class`

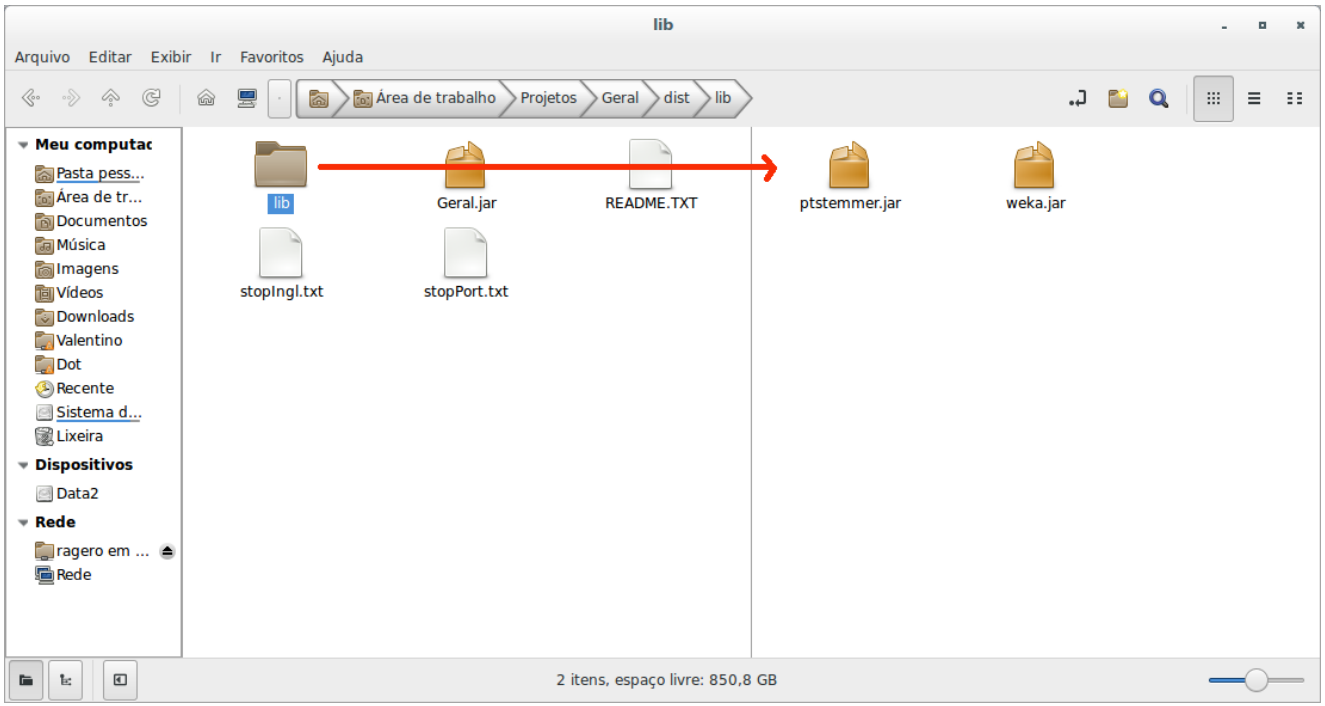
Criando arquivos .jar pelo NetBeans

- Para gerar um .jar no NetBeans, basta clicar com o botão direito em cima do projeto e depois em `Construir` ou `Limpar e Construir`



Criando arquivos .jar pelo NetBeans

- O Netbeans gera uma pasta nomeada com `dist` dentro da pasta do projeto
- Essa pasta conterá o `.jar` e caso seu projeto utilize bibliotecas externas, esta bibliotecas são copiadas na pasta `lib` que ficará dentro da pasta `dist`



Executando arquivos .jar via linha de comando

- Para executar um .jar via linha de comando deve-se utiliza a seguinte sequência:
  `java -jar [nome do arquivo jar].jar`
  Ex: `java -jar programa.jar`

- Caso um programa possua mais de uma classe com um método `main` e o usuário queira executar os *mains* de outras classes que não seja a classe principal, pode-se chamar tais métodos de tais classes com a seguinte sequência:
  `java -cp [nome do arquivo jar].jar [classe com método main a ser executada]`
  Ex: `java -cp programa.jar classe2`

Executando arquivos .jar via interface gráfica

- No Windows: duplo clique
- No Linux:
  - Se estiver com o Open JDK basta clicar com o botão direito e depois ‘‘OpenJDK Java X Runtime’’
  - Se estivem com o Oracle JDK instalado via o gerenciador gráfico de pacotes, basta clicar com o botão direito e depois ‘‘Java Runtime Environment (JRE)’’
  - Para qualquer situação, pode-se:
    1. Clicar com o botão direito do mouse no arquivo .jar
    2. ‘‘Open With’’ ou “Abrir Com”
    3. ‘‘Other Application’’ ou “Outras Aplicações”
    4. ‘‘Enter a custom command’’ ou “Digite um comando personalizado” e digite java -jar
    5. Defina esse comando como comando padrão

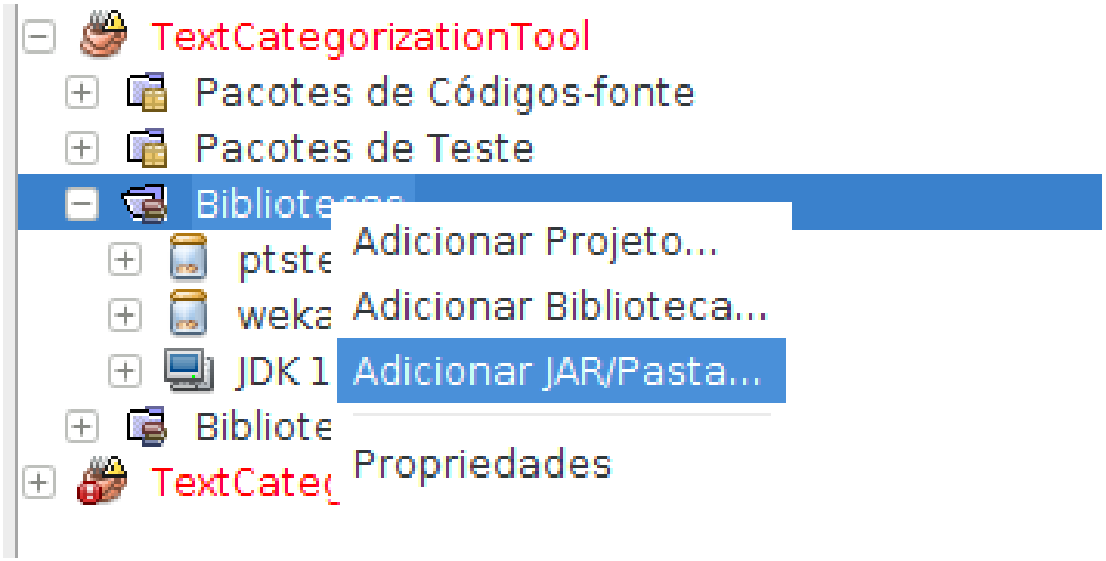
Utilizando Bibliotecas em Java

- Desde as mais diversas até as principais bibliotecas Java desenvolvidas por terceiros são distribuídas em arquivos JAR

- Alguns exemplos famosos:
  - `Weka`: algoritmos de aprendizado de máquina
  - `Gson`: serialização e deserialização de objetos no formato JSON
  - `JFreeChart`: criação de gráficos gráficos
  - `JCalendar`: seleção de datas por meio de componentes GUI
  - `iText`: manipulação de PDFs

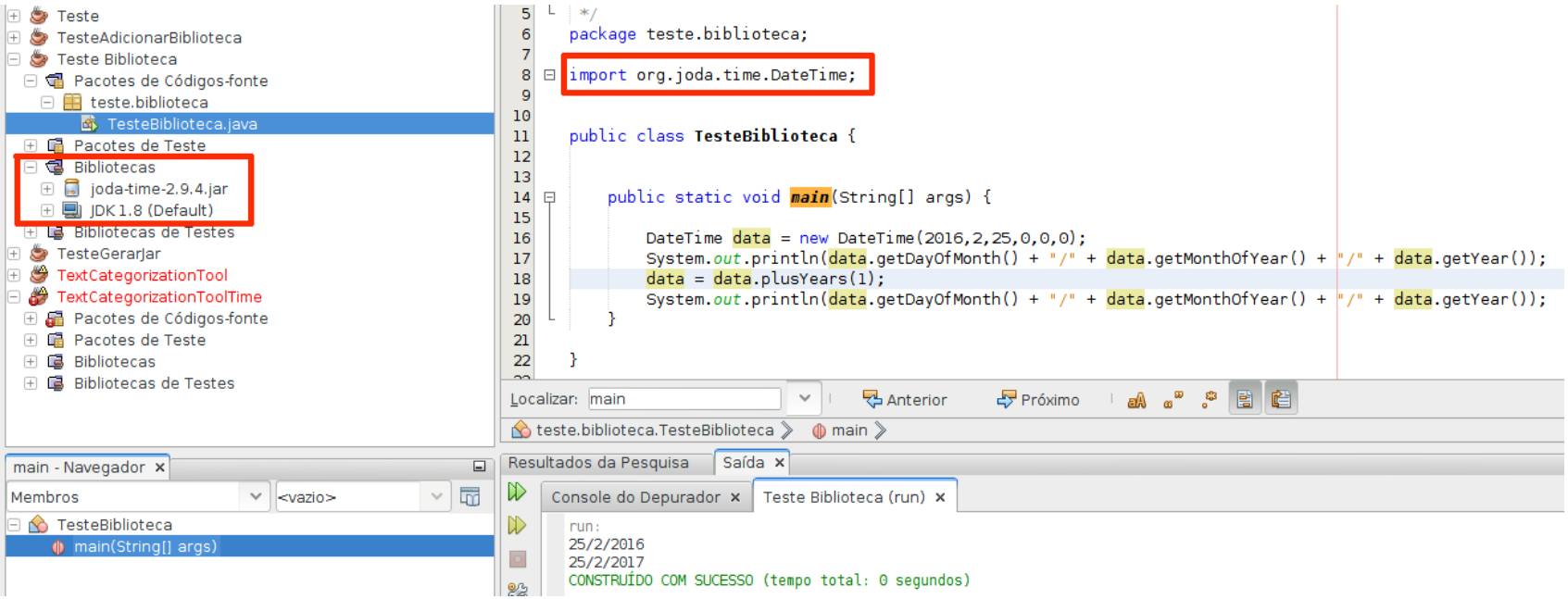
Inserindo uma Biblioteca no Netbeans

- Para adicionar uma biblioteca no Netbeans basta clicar com o botão direito em cima do projeto, depois em `Adicionar Jar/Pasta` e, por fim, definir o caminho da biblioteca desejada



- Depois de inserida a biblioteca, pode-se utilizá-la normalmente como se fosse a biblioteca padrão do Java

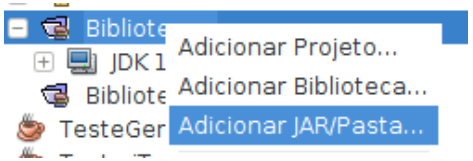
Inserindo uma Biblioteca no Netbeans

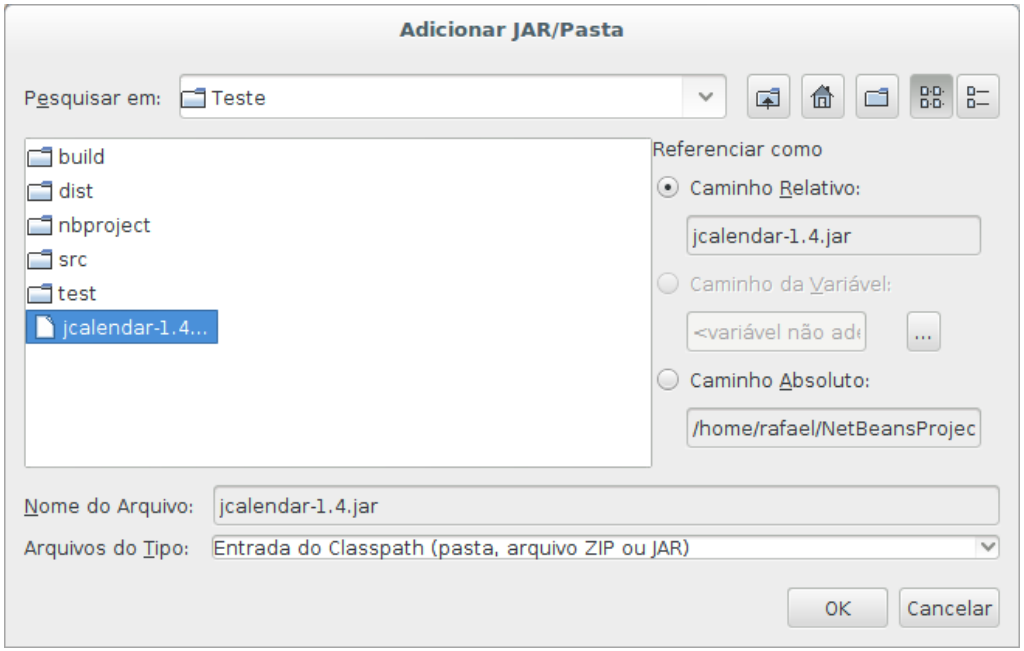


Exemplo de uso da biblioteca JCalendar

- O JCalendar permite o usuário definir uma data de maneira gráfica, isto é, interagindo com um componente gráfico por meio do mouse ao invés de digitar a data completa em um campo texto, por exemplo
- Para exemplificar o uso do JCalendar, baixe o .jar dessa biblioteca no Moodle
- Em seguida, vamos inserir esse .jar no nosso projeto

Exemplo de uso da biblioteca JCalendar



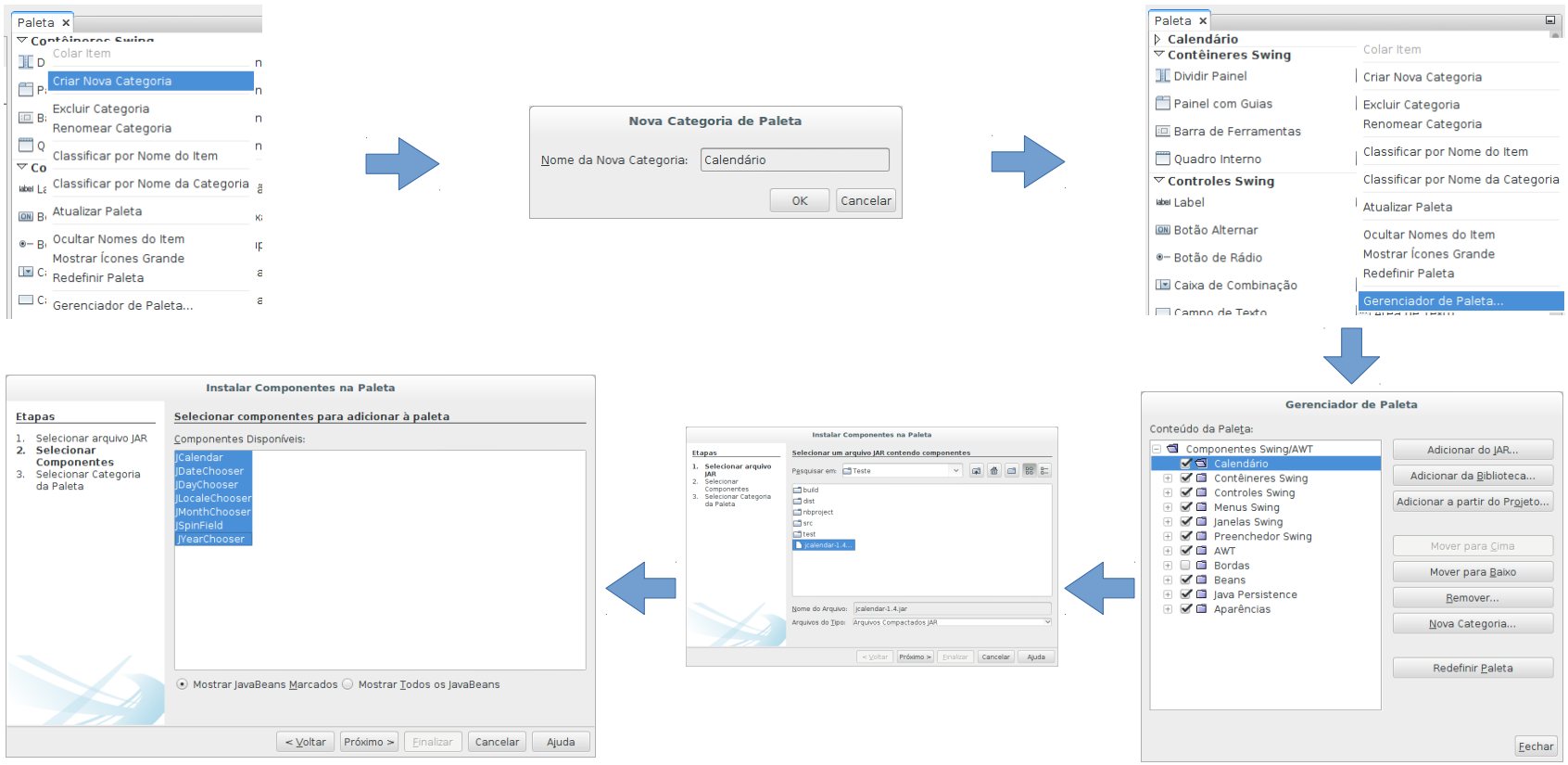


Exemplo de uso da biblioteca JCalendar

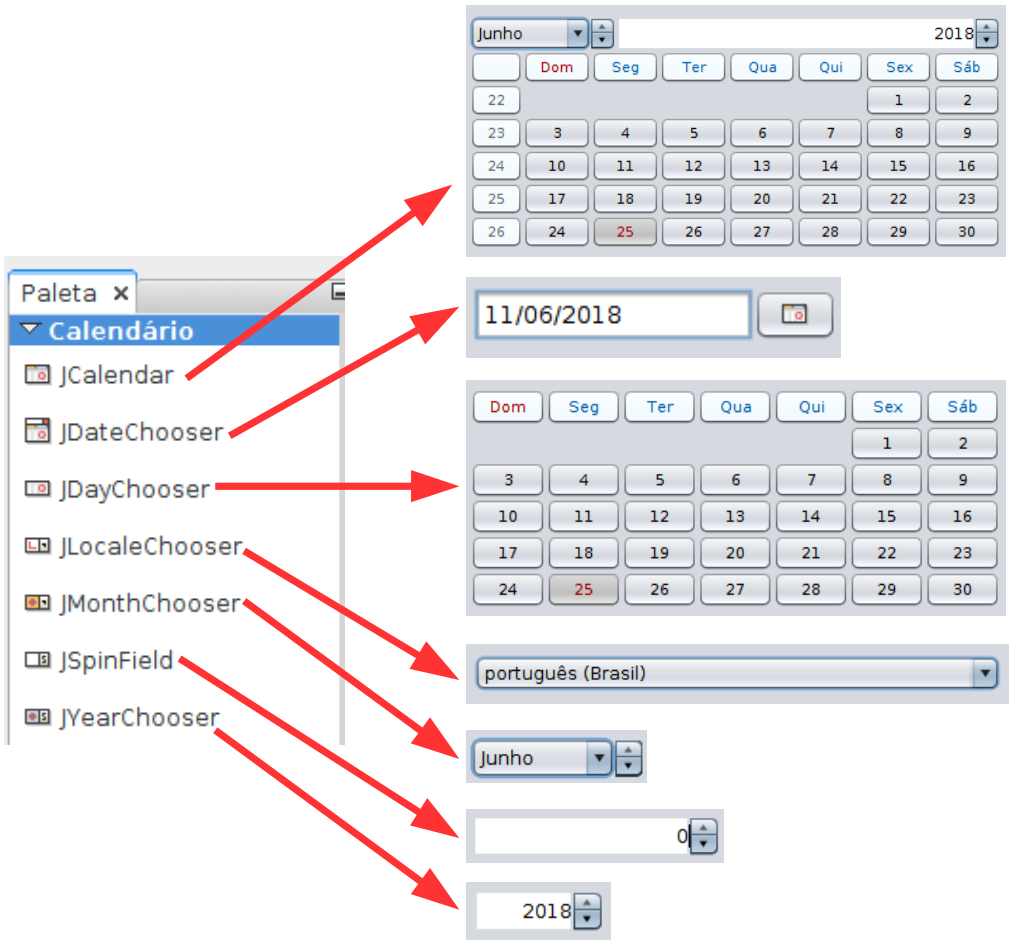
- Podemos criar os componentes dessa biblioteca e adicioná-los manualmente

- Ou podemos adicioná-los na paleta para que possamos "desenhá-los"

Exemplo de uso da biblioteca JCalendar

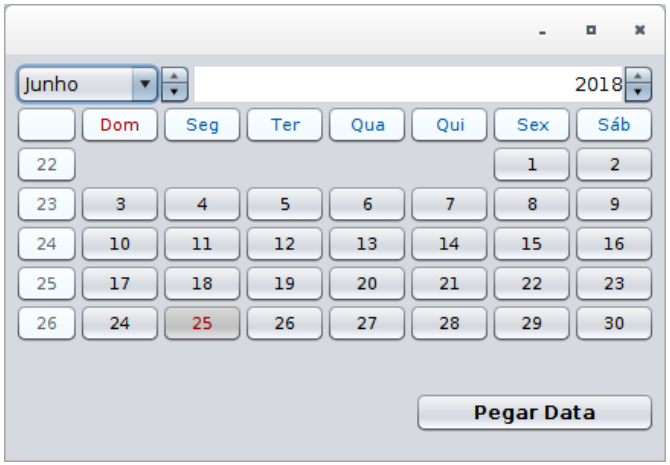


Exemplo de uso da biblioteca JCalendar



Exemplo de uso da biblioteca JCalendar

- Para cada um dos componentes da biblioteca JCalendar, há um método get para retornar um objeto de data ou um determinado valor dependendo do tipo do objeto




```
private void bPegarDataActionPerformed(java.awt.event.ActionEvent evt) {
    JOptionPane.showMessageDialog(null,cCalendario.getDate());
}
```

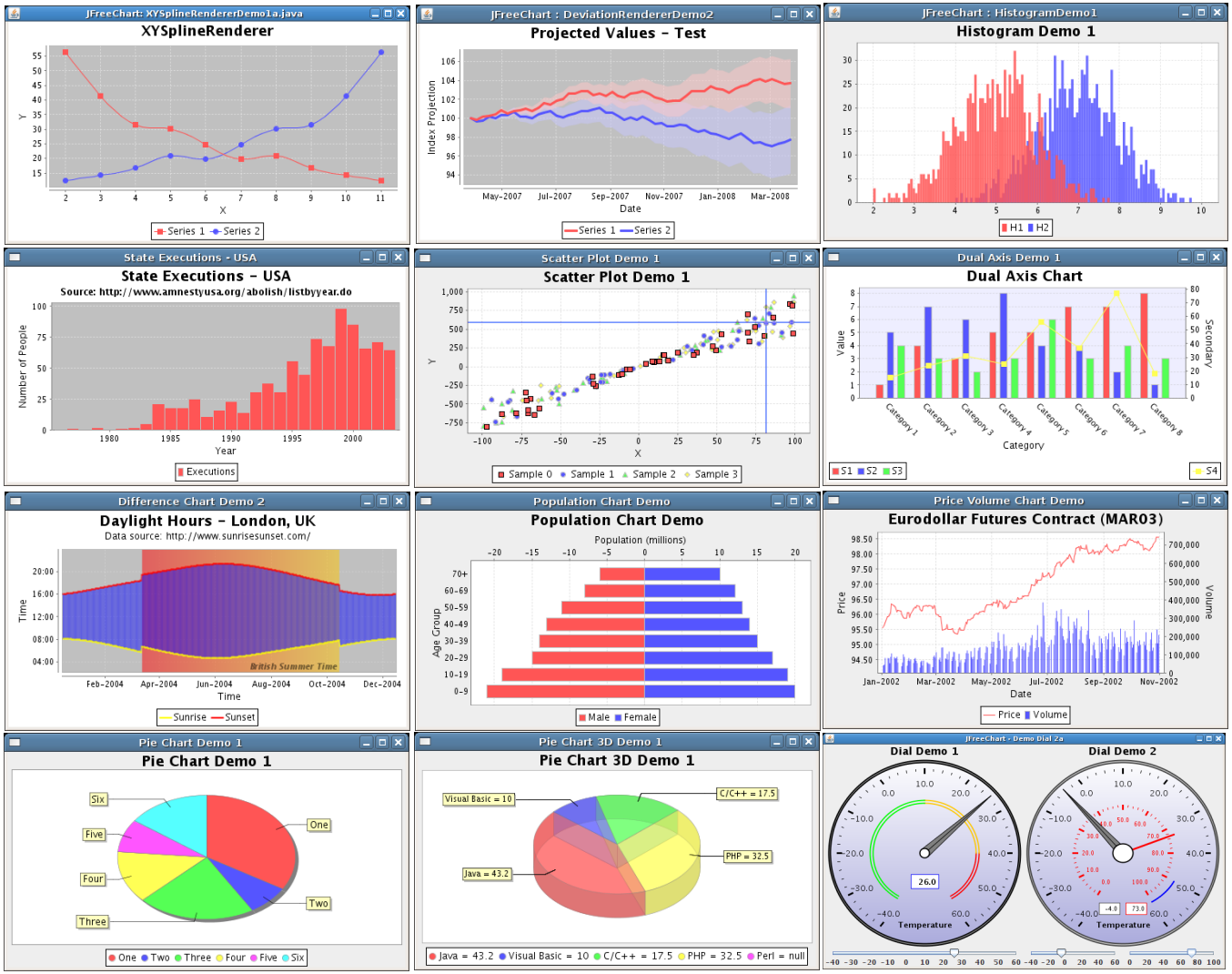
Exemplo de uso da biblioteca JFreeChar

- JFreeChart é uma biblioteca conhecida para gerar gráficos de diversos formatos
- Possui facilidades para customização dos gráficos quanto à cores, formatos dos itens dos gráficos, etc.
- Gera saídas na própria interface gráfica quanto em arquivos JPG, PNG, SVG e EPS
- Para maiores informações, exemplos e documentação: `http://www.jfree.org/jfreechart/`

Exemplo de uso da biblioteca `JFreeChar`

- A principal classe do `JFreeChart` é a classe `JFreeChart` $\rightarrow$ classe pai que pode assumir qualquer tipo de gráfico
- Outra classe importante é a classe CharctFactory, a qual possui métodos estáticos para gerar diferentes tipos de gráficos
- Também é necessário criar um dataset para inserir os dados que serão plotados no gráfico
  - Neste caso vamos usar um DefaultCategoryDataset
  - Os dataset possuem métodos para adicionar valores
- A classe ChartUtilities possui métodos utilitários para, por exemplos, gravar os dados em imagens

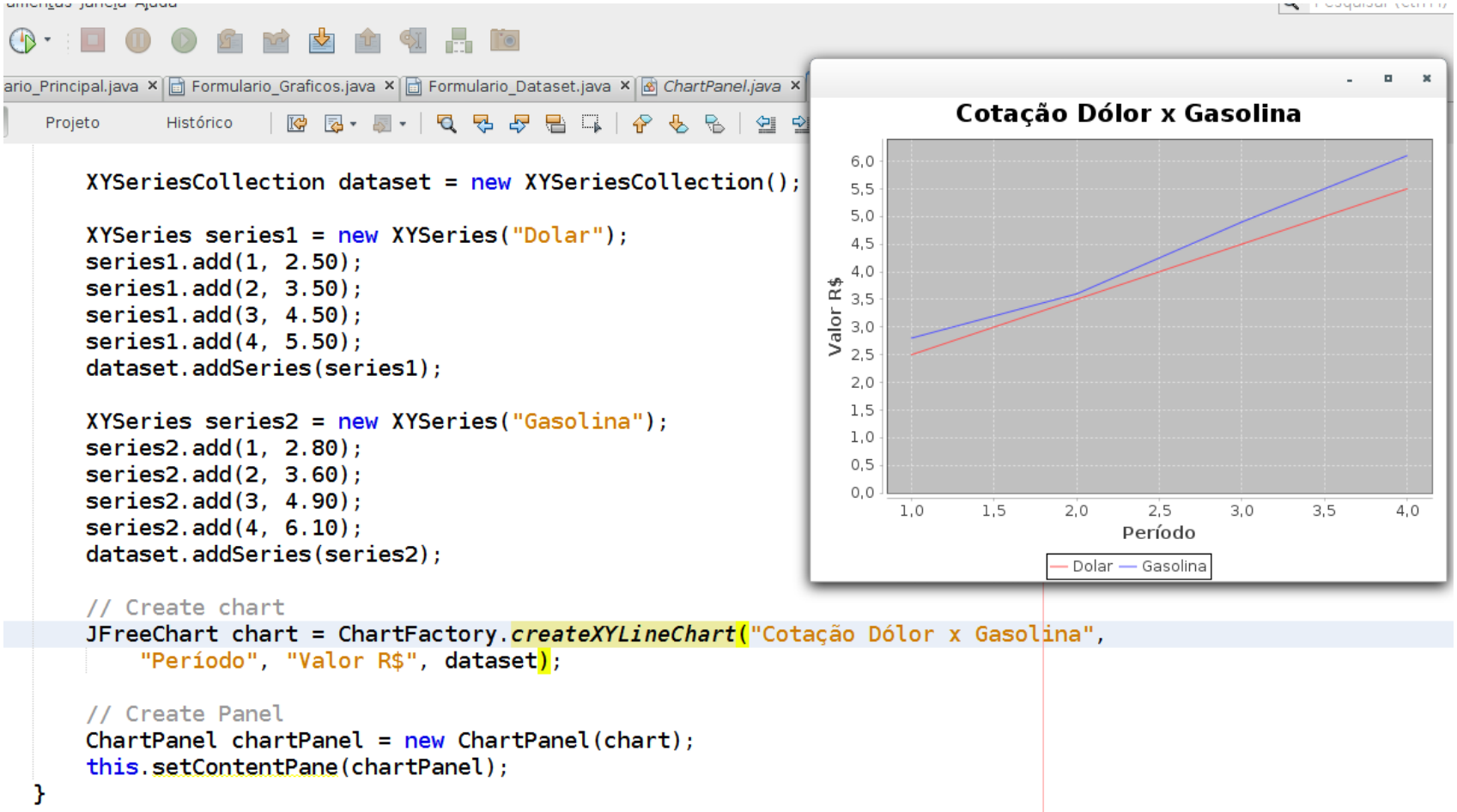
Exemplo de uso da biblioteca JFreeChar



Exemplo de uso da biblioteca JFreeChar

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

    //Criando a base de dados
    DefaultCategoryDataset ds = new DefaultCategoryDataset();
    ds.addValue(1000, "Vendedor A", "Janeiro");
    ds.addValue(1200, "Vendedor B", "Janeiro");
    ds.addValue(1050, "Vendedor A", "Fevereiro");
    ds.addValue(1250, "Vendedor B", "Fevereiro");
    ds.addValue(1150, "Vendedor A", "Março");
    ds.addValue(1350, "Vendedor B", "Março");

    //Criando o gráfico
    JFreeChart chart = ChartFactory.createBarChart("Teste Gráfico de Barras", "Meses", "Total de Vendas", ds);

    //Gravando o gráfico
    try {
        OutputStream arquivo = new FileOutputStream("grafico.png");
        ChartUtilities.writeChartAsPNG(arquivo, chart, 550, 400);
        arquivo.close();
    } catch (IOException ex) {
        Logger.getLogger(TesteJFreeChart1.class.getName()).log(Level.SEVERE, null, ex);
    }

}
```

Exemplo de uso da biblioteca JFreeChar



Exemplo de us da biblioteca iText - Texto Simples

```
22  public class ExemploUsoIText {
23
24
25      public static void main(String[] args) {
26
27          File file = new File("teste_pdf.pdf");
28
29          Document document = new Document(); // Criando um documento
30          document.setPageSize(PageSize.A4); // Definido o tamanho do documento
31
32
33
34          try{
35              PdfWriter.getInstance(document, new FileOutputStream(file)); // Atribuindo o documento criado à um arquivo
36              document.open(); // Abrindo o documento
37
38              document.add(new Paragraph("Hello World!\n\n")); // Adicionando um parágrafo no documento
39
40              document.close(); // Fechando o documento
41
42              Desktop.getDesktop().open(file); // Abrindo o PDF com o programa padrão do Desktop
43
44          }catch(Exception e){
45              e.printStackTrace();
46          }
47
48      }
49
50  }
```

Exemplo de us da biblioteca iText - Tabela

```
31          File file = new File("teste_pdf.pdf");
32
33          Document document = new Document(); // Criando um documento
34          document.setPageSize(PageSize.A4); // Definido o tamanho do documento
35
36
37
38          try{
39              PdfWriter.getInstance(document, new FileOutputStream(file)); // Atribuindo o documento criado à um arquivo
40              document.open(); // Abrindo o documento
41
42              float[] columnWidths = {200,200};
43              PdfPTable table = new PdfPTable(columnWidths);
44              table.setTotalWidth(columnWidths);
45              table.setLockedWidth(true);
46
47
48              PdfPCell cell = new PdfPCell();
49              cell.setPaddingTop(4.5f);
50              cell.setPaddingBottom(4.5f);
51              cell.setVerticalAlignment(Element.ALIGN_MIDDLE);
52              cell.setBackgroundColor(BaseColor.GRAY);
53              cell.setBorderColor(BaseColor.BLACK);
54              Paragraph par = new Paragraph("Célula1");
55              par.setAlignment(Element.ALIGN_RIGHT);
56              cell.addElement(par);
57              table.addCell(cell);
58
59              cell = new PdfPCell();
60              cell.setPaddingTop(4.5f);
61              cell.setPaddingBottom(4.5f);
62              cell.setVerticalAlignment(Element.ALIGN_MIDDLE);
63              cell.setBackgroundColor(BaseColor.BLUE);
64              cell.setBorderColor(BaseColor.BLACK);
65              par = new Paragraph("Célula2");
66              par.setAlignment(Element.ALIGN_LEFT);
67              cell.addElement(par);
68              table.addCell(cell);
69
70              document.add(table); // Adicionando um parágrafo no documento
71
72              document.close(); // Fechando o documento
73
74              Desktop.getDesktop().open(file); // Abrindo o PDF com o programa padrão do Desktop
75
76          }catch(Exception e){
77              e.printStackTrace();
78          }
```

Exemplo de us da biblioteca iText - Imagem

```
27  public class ExemploUsoIText {
28
29
30      public static void main(String[] args) {
31
32          File file = new File("teste_pdf.pdf");
33
34          Document document = new Document(); // Criando um documento
35          document.setPageSize(PageSize.A4); // Definido o tamanho do documento
36
37
38
39          try{
40              PdfWriter.getInstance(document, new FileOutputStream(file)); // Atribuindo o documento criado à um arquivo
41              document.open(); // Abrindo o documento
42
43              Image image = Image.getInstance("ufms_logo.png"); // Obtendo uma instancia de imagem
44              image.scaleToFit(200, 200); // Reescalando a imagem
45
46              document.add(image); // Adicionando um parágrafo no documento
47
48              document.close(); // Fechando o documento
49
50              Desktop.getDesktop().open(file); // Abrindo o PDF com o programa padrão do Desktop
51
            }catch(Exception e){
                e.printStackTrace();
54          }
55
56      }
57
58  }
```

Extra: Alocando Mais Memória para a Execução do seu JAR

- Muito do consumo de memória dos programas se dá na área de HEAP (área destinada à alocação de memória dinâmica)
- Se objetos são alocados dinamicamente $\rightarrow$ objetos são alocados na área de HEAP.
- Geralmente o tamanho máximo padrão da área de HEAP de uma JVM é 1/6 da memória física

Extra: Alocando Mais Memória para a Execução do seu JAR

- Entretanto, pode-se alterar o tamanho do heap de um programa Java caso este necessite processar um grande volume de dados e criar muitos objetos

- Pode-se utilizar os seguinte parâmetros em conjunto com a quantidade de memória
  - -Xms: definir a quantidade de memória inicial do *heap*
  - -Xmx: definir a quantidade máxima de memória do *heap*

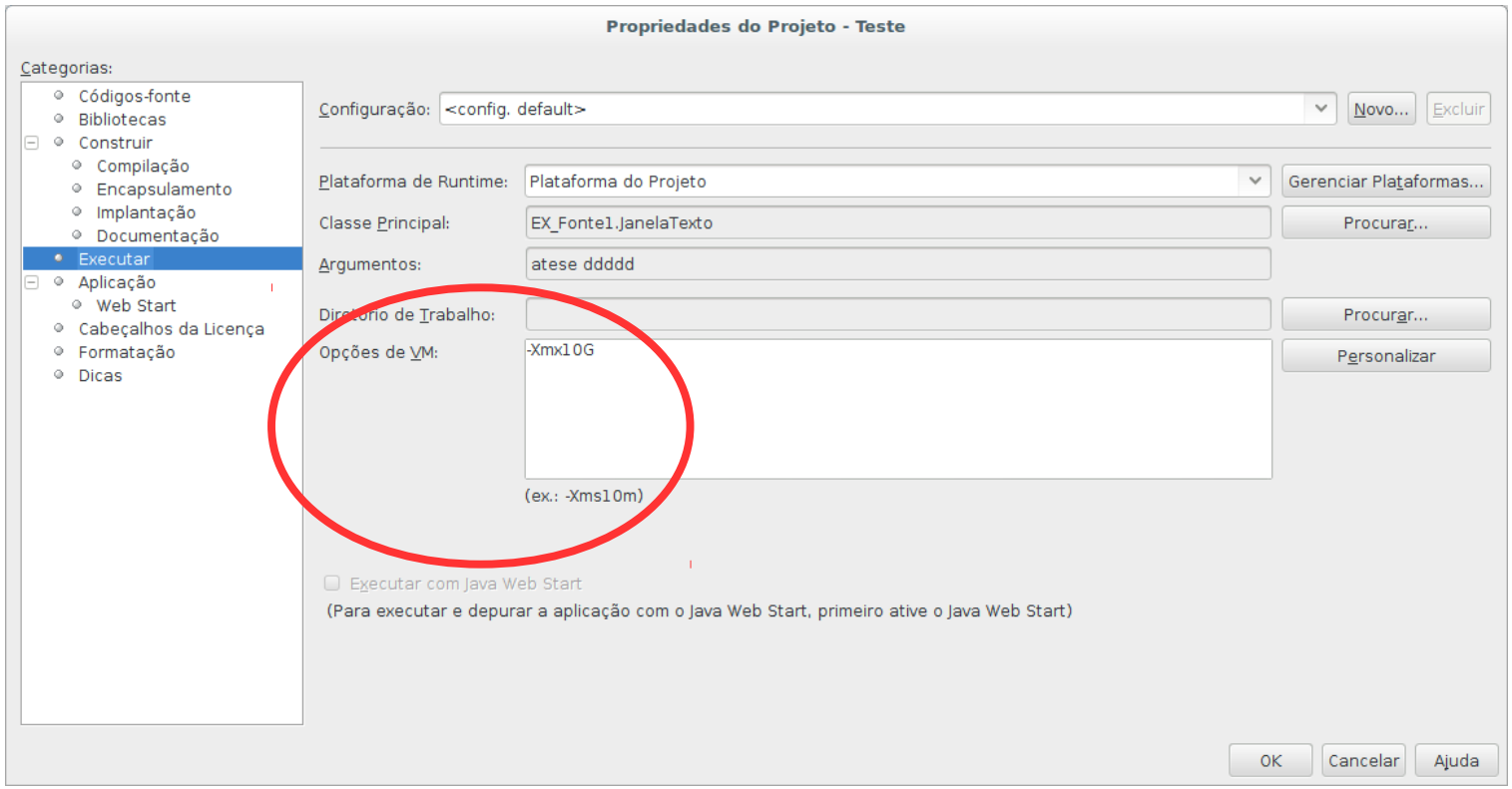
Extra: Alocando Mais Memória para a Execução do seu JAR

- Exemplo em linha de comando:

```
java -Xmx50000M -jar TextCategorizationTool_MultiThreading.jar
```

Extra: Alocando Mais Memória para a Execução do seu JAR

- No Netbeans: `Executar > Definir Configurações do Projeto > Personalizar`



Material Complementar

- JAR (formato de arquivo)

  `https://pt.wikipedia.org/wiki/JAR_(formato_de_arquivo)`

- Lesson: Packaging Programs in JAR Files

  `https://docs.oracle.com/javase/tutorial/deployment/jar/`

- Utilizando Bibliotecas JAR: Primeiros Passos

  `http://www.devmedia.com.br/`

  `utilizando-bibliotecas-jar-primeiros-passos/25339`

Material Complementar

- Começando com parâmetros e configurações da JVM

  `http://blog.caelum.com.br/`

  `comecando-com-parametros-e-configuracoes-da-jvm/`

- Xms, Xmx, XX:MaxPermSize, XX:PermSize - Qual a diferença?

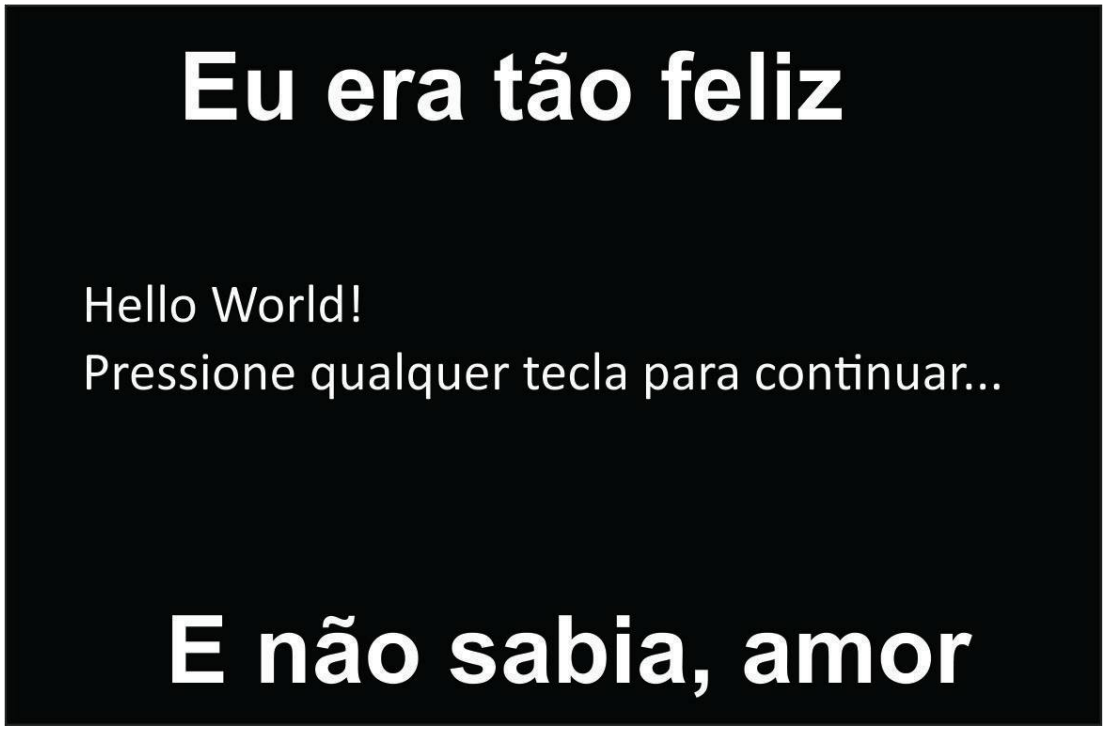
  `https://pt.stackoverflow.com/questions/37872/`

  `xms-xmx-xxmaxpermsize-xxpermsize-qual-a-diferen%C3%A7a`

Material Complementar

- Gráficos com JFreeChart
  `http://www.caelum.com.br/apostila-java-testes-xml-design-patterns/graficos-com-jfreechart/`
- JFreeChat
  `http://www.jfree.org/jfreechart/`
- JCalendar
  `https://toedter.com/jcalendar/`
- Gerando PDF: iText
  `https://www.devmedia.com.br/gerando-pdf-itext/18843`

Material Complementar



Programação Orientada a Objetos
http://lives.ufms.br/moodle/

Rafael Geraldeli Rossi
rafael.g.rossi@ufms.br

Slides baseados em [Deitel and Deitel, 2010]

Referências Bibliográficas I


Deitel, P. and Deitel, H. (2010).
Java: How to Program.
How to program series. Pearson Prentice Hall, 8th edition.