

Java 2 Standard Edition

Configuração do ambiente JEdit + Ant

Java "IDE" configurável

- *Este módulo mostra como montar um ótimo ambiente de desenvolvimento para aplicações Java e XML usando apenas ferramentas open-source*
 - **JEdit** (editor de textos, código e ambiente integrado)
 - **Ant** (ferramenta de gerenciamento de projetos)
 - **JSDK 1.4.0** (kit de desenvolvimento da Sun)
- *Onde conseguir o software*
 - JEdit: www.jedit.org
 - Ant: jakarta.apache.org/ant
 - JSDK: java.sun.com



■ 1. SDK

- a) Instale o JSDK seguindo as instruções mostradas na tela de instalação
- b) Configure uma variável de ambiente `JAVA_HOME` que aponte para o endereço onde o JSDK foi instalado
- c) Acrescente o caminho `$JAVA_HOME/bin` no seu `PATH`

■ 2. Ant

- a) Abra o ZIP da última versão binary do Ant em um diretório (ex: `/usr/local/ant` ou `c:\ant`)
- b) Configure uma variável de ambiente `ANT_HOME` que aponte para o endereço onde o ANT foi instalado
- c) Acrescente o caminho `$ANT_HOME/bin` no seu `PATH`


■ 3. JEdit

- Instale o JEdit clicando duas vezes no JAR de distribuição ou executando `java -jar jEdit4Install.jar`

Teste da instalação

- **SDK e JRE**
 - Digite *java -version*
 - O resultado deve ser o número da versão instalada
 - Digite *javac*
 - O resultado deve ser uma mensagem de erro com a lista de opções válidas para o *javac*.
- **Ant**
 - Digite *ant*
 - O resultado deve ser a mensagem de erro *Buildfile: build.xml does not exist*
- **JEdit**
 - *Windows: clique duas vezes no ícone criado*
 - *Unix: rode o executável no diretório onde foi instalado*

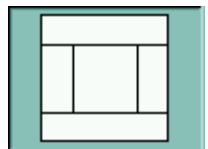
JEdit: plug-ins

- O JEdit, sem plug-ins, é um simples editor de textos
- Com plug-ins pode ter mais recursos e melhor integração que muitos IDEs comerciais
- Para instalar plug-ins (precisa estar conectado à Internet)
 - a) Inicie o JEdit
 - b) Selecione o menu "Plugins" / "Plugin Manager" (ou ícone )
 - c) Aperte o botão "Install Plugins". Após a conexão será mostrada uma lista dos plug-ins disponíveis. Clique em cada um para ver sua descrição.
 - d) Selecione o radio-button "Install in system plug-in directory"
 - e) Marque os plug-ins desejados e aperte "Install"
- Instale pelo menos...
 - AntFarm, AStyle, Buffer Tabs, Code2HTML, Console, Drag & Drop, Editor Scheme, Error List, Java Style, JBrowse, JCompiler, Project Viewer, SpeedJava, Templates, XML, XSLT

Personalização do JEdit (I)

- Reinicie o JEdit após a instalação dos plug-ins
 - Se algum plug-in instalado apresentar erro, uma mensagem irá ser exibida na abertura do JEdit. Você pode
 - corrigir o erro (a mensagem explica como ou a causa)
 - desinstalar o plug-in (no menu Plugins)
- Docking
 - Coloca plug-ins frequentemente usados nas laterais da área de trabalho para fácil utilização
 - Selecione o menu "Utilities" / "Global Options"
 - Na opção jEdit Options / Docking, selecione a segunda configuração de tela: —————→
 - Mude a posição default dos seguintes plug-ins:

■ AntFarm: left	■ JBrowse: left
■ Console: bottom	■ Project: left
■ ErrorList: bottom	■ XMLTree: left



Personalização do JEdit (2)

- *Tabs para seleção de janelas de texto*
 - *Menu Global Options / jEdit Options / General*
 - *Desmarque "Show buffer switcher" e "Show search bar"*
 - *Menu Global Options / Plugin Options / Buffer Tabs*
 - *Marque "Enable buffer tabs by default"*
 - *Selecione "Location of buffer tabs:" para "top"*
- *AntFarm*
 - *Menu Global Options / Plugin Options / AntFarm*
 - *Selecione Build Options*
 - *Na seção "Build Execution Method", na segunda caixa de texto, informe o caminho até o executável do Ant na sua máquina (ex: c:\ant\bin\ant.bat)*
 - *Na seção "General Build Options" marque as opções "Load build files..." e "Save all buffers..."*

Personalização do JEdit (3)

- *ErrorList*
 - *Menu Global Options / Plugin Options / ErrorList*
 - Marque "automatically display on error"
- *JBrowse*
 - *Menu Global Options / Plugin Options / JBrowse*
 - Marque "Display Status Bar", "Automatic parse" e "Sort"
- *Numeração de linhas*
 - *Menu Global Options / jEdit Options / Gutter*
 - Marque "Line Numbering"
- *Tabulação*
 - *Menu Global Options / jEdit Options / Editing*
 - Em "Tab Width" e "Indent Width" coloque "4"

Resultado da personalização

The screenshot shows the jEdit IDE interface. The main editor displays the source code of a Java class named `Livro`, which extends `Publicacao`. The code includes package declarations, imports, and several methods. The IDE's status bar at the bottom indicates 14 errors and 0 warnings.

Source Code (Livro.java):

```
1 package biblioteca;
2
3 import java.util.Date;
4
5 /**
6  * Representa um livro (publicação com um ou mais
7  * autores que possui um assunto)..
8  */
9 public class Livro extends Publicacao {
10
11     private Assunto assunto;
12     private Autor[] autores; // pode ser vazio no caso de livros sem autor.
13
14     public Livro(int codigo, String titulo, Editor editor, Date data, Assunto assunto, Autor[] autores) {
15         super(codigo, titulo, editor, data);
16         this.assunto = assunto;
17         this.autores = autores;
18     }
19
20     public Assunto getAssunto() {
21         return assunto;
22     }
23
24     public void setAssunto(Assunto assunto) {
25         this.assunto = assunto;
26     }
27
28     public Autor[] getAutores() {
29         return autores;
30     }
31 }
```

Compilation Errors:

- 9: cannot resolve symbol
- 11: cannot resolve symbol
symbol : class Assunto
location: class biblioteca.Livro
- 12: cannot resolve symbol
symbol : class Autor
location: class biblioteca.Livro
- 14: cannot resolve symbol

The IDE interface includes a menu bar (File, Edit, Search, Markers, Folding, View, Utilities, Macros, Plugins 1, 2, 3, Help), a toolbar, a project tree on the left showing the file structure, and a bottom status bar with file encoding (Cp1252) and memory usage (14Mb/26Mb).

Como usar o Console / ErrorList

- O **Console** do JEdit serve para
 - *mostrar mensagens de erro*
 - *rodar o compilador*
 - *rodar outras aplicações do sistema*
- *Erros ocorridos durante a compilação, execução ou outro processo que produza erros são coletados no **ErrorList**, que*
 - *permite acesso rápido à fonte do erro através de double-click*
 - *oferece detalhamento dos erros*
- *Ajuste o Console e ErrorList na parte inferior do seu JEdit*

Criando um projeto

- *Para criar um novo projeto:*
 - *primeiro escolha um local no seu disco que será a raiz de seus projetos*
 - *depois crie um diretório para seu projeto*
 - *finalmente, usando o JProject, clique no ícone "New Project", informe um nome e o diretório*
- *Como adicionar arquivos ao projeto*
 - *Sempre que o JProject estiver aberto, ele perguntará se o arquivo salvo deve ser incluído no projeto*
 - *Arquivos também podem ser adicionados usando o ícone correspondente*

Como criar um template

- *Templates são úteis para se ganhar tempo com textos ou estruturas repetitivas, por exemplo:*
 - *Estrutura default de uma página HTML*
 - *Estrutura default de uma classe Java*
- *Para criar um novo template,*
 - *Edite-o no JEdit e, na hora de salvar, selecione o menu Plugins/Templates/Save Template*
 - *Selecione Refresh Templates para que apareça na lista*
 - *Crie um arquivo novo e selecione o template da lista para que o seu conteúdo seja copiado*
- *Crie templates básicos*
 - *Classe Java*
 - *Buildfile do Ant*

- *SpeedJava ("code insight")*
 - *Acionado quando você digita um ponto "."*
 - *Lista métodos e variáveis da classe do objeto selecionado*
 - *Limitação (versão 0.2):*
 - *classe tem que ter sido importada com import pacote.* (não funciona se classe foi importada nominalmente)*
- *Reformat Buffer e JavaStyle / Reformat Buffer*
 - *Rearruma código Java*
 - *Configure para refletir seu estilo de codificação*
 - *JavaStyle oferece mais opções de configuração*
- *XML e XSLT*
 - *Oferecem suporte para XML, XSLT e XPath*
 - *Validam XML com DTD, oferecem ajuda de contexto (quando há um DTD vinculado), montam árvore (plug-in XML Tree)*

- Os plug-ins do JEdit são desenvolvidos por programadores independentes e podem não ter a mesma qualidade ou utilidade dos recursos nativos do JEdit
 - Alguns contêm bugs (falham ocasionalmente)
 - Alguns poderiam ser melhores
 - A integração entre os plug-ins é inferior à desejável
 - Ainda não há plugins para desenho de GUI, construção de EJBs, etc.
- Soluções
 - 1) Esperar versões mais novas dos plug-ins (a atualização requer apenas apertar um botão (Update Plugins) no Plugin Manager)
 - 2) Enviar sugestões aos autores dos plug-ins
 - 3) Escrever macros, scripts do Ant (EJB, arquivos WAR, etc.)
 - 4) Participar do projeto open-source: baixar o código-fonte Java, fazer as alterações desejadas e enviar patches aos autores
 - 5) Escrever novos plug-ins (integrando ou não com os existentes)
 - 6) Usar outro IDE

Alguns IDEs alternativos

- **Open source**
 - *Projeto Eclipse*
 - *NetBeans*
 - *Jext (Java Text Editor - similar ao JEdit) - não edita GUI*
- **Free**
 - *Borland JBuilder Personal Edition*
 - *Sun Forté for Java Community Edition (baseado no NetBeans)*
- **Leves**
 - *JCreator*
 - *Kawa*
 - *IDEA*
- **Comerciais**
 - *IBM Visual Age / WebSphere Studio*
 - *Borland JBuilder*
 - *Sun Forté for Java*

Servem para aumentar a produtividade: evite usá-los para aprender Java (prefira um editor de textos como oJEdit ou Jext)

Teste cada um com as ferramentas e recursos que você costuma usar e veja o que melhor se adapta às suas necessidades.

I. Organize os exercícios que você fez nos módulos anteriores em projetos no JEdit

- *a) Crie um diretório para cada um (coloque as fontes no subdiretório src)*
- *b) Crie um projeto no Project Manager*
- *c) Faça toda a compilação através do Ant (defina o build.xml disponível no diretório cap03/ como template, salve-o na raiz do seu projeto e inclua-o clicando no "+" do AntFarm)*
- *d) Execute através do Console*

Dica: veja artigo da JavaMagazine (anexo) que mostra com detalhes como montar o ambiente deste módulo