# PROJETO DE SISTEMAS III

Prof. Maíra Gatti

**Tutorial Eclipse** 

# **Eclipse User Guide**

- 1. Instalação do JDK
- 2. Introdução
- 3. Perspectiva
- 4. Plugins
- 5. Workspace
  - 1. Adicionando views para uma perspectiva
  - 2. Arrumando o workbench
  - 3. Manipulando recursos com ferramentas externas
- 6. Instalação
- 7. Configuração Básica
  - 1. Estrutura das pastas nos projetos Java
  - 2. Use JDK ao invés de JRE (ou instale um novo JRE/JDK)
- 8. Configurando classpath variables
- 9. O primeiro projeto Java
  - 1. A primeira classe Java
- 10. Configurando o Launcher Executando um programa
- 11. Importando um projeto existente
- 12. Configurando o Java Buildpath
- 13. Rodando em modo Debug
  - 1. Debugger
    - 1. Breakpoints
    - 2. Remote Debugging
    - 3. Connection
  - 2. Debug View
  - 3. Realizando debugging nos programas
- 14. PLUGIN'S
  - 1. Lomboz
    - 1. Configuração
    - Ativação
- 15. Eclipse e Junit
  - 1. Configurando o projeto
  - 2. O primeiro teste
  - 3. Executando o primeiro teste:
- 16. Eclipse e Ant
  - 1. Executando Ant
  - 2. Clear Output

### Instalação do JDK

A primeira coisa a fazer para utilizar a linguagem Java é instalar o Java Development Kit, ou JDK.

Observe a estrutura de diretórios do drive C: se já houver um diretório JDK\* (JDK1.2, JDK1.2.2, JDK1.2.1, JDK1.3, JDK1.3.1, JDK1.4), o ambiente já está instalado. Se não houver, será necessário instalar uma das versões. Recomenda-se a instalação do JDK 1.4, disponível na página da Sun: selecione a versão adequada para seu sistema operacional (Windows, Linux, Solaris, etc), aceite a licença de uso, informe o diretório adequado (c:\jdk1.4) e confirme. Após a instalação, a seguinte estrutura estará disponível:

c∶∖jdk1.4

```
bin
demo
include
jre
bin
lib
```

 bin - contém todos os programas utilitários necessários para compilação e execução

 demo - contém diversos programas em Java para demonstrar alguns recursos da linguagem

 include - contém os headers para interface de programas Java e C (se você não entendeu para que serve isso, provavelmente não irá precisar)

 jre (Java Run-time Environment) - contém os programas e bibliotecas necessários para executar código Java.

 jre\lib - contém as bibliotecas necessária para a execução e compilação de programas Java

Observação: o instalador do JDK normalmente inclui o caminho c:\jdk1.4\bin no PATH. Para testar, abra uma janela DOS e digite:

```
c:\winnt> java -version
java version "1.4.0"
Java(TM) 2 Runtime Environment, Standard
Edition (build 1.4.0-b92)
```

# Java HotSpot(TM) Client VM (build 1.4.0-b92, mixed mode)

Se aparecer uma mensagem de erro informando que não foi possível achar o arquivo, ajuste corretamente o PATH. No Windows NT isso é feito pelo painel de controle, system properties. No Windows 95/98, deve ser alterado o arquivo AUTOEXEC.BAT e reinicializada a máquina.

# Introdução

O Eclipse é um *framework* para integrar diferentes tipos de aplicações. Uma de suas aplicações é a JDT(Java Development Tooling), a qual já vem com o Eclipse.

Essas aplicações são oferecidas em forma de plugins e automaticamente reconhecidas e integradas pela plataforma. Tendo seus próprios recursos para gerenciamento de mecanismo, que são gera lmente arquivos no seu Hard Disk. Eles residem no seu workspace, uma pasta especial localizada no seu sistema de arquivos. As aplicações instaladas comunicam-se entre si, com isso, se uma aplicação altera um recurso qualquer, todas as outras aplicações instaladas serão notificadas sobre essa mudança, garantindo uma consistência e integração em todo o seu ambiente de desenvolvimento.

Um usuário sempre trabalha no workbench, a parte "visível" da plataforma (GUI). A perspectiva escolhida determina a aparência atual do workbench. A perspectiva é uma coleção conhecida como "views e editores" que contribuem com ações especiais para o menu e a toolbar.

A maioria das views mostra informações especiais sobre os recursos. Dependendo da view somente partes ou relacionamentos internos dos recursos poderão ser mostrados. Como no caso de arquivos do tipo XML.

Um editor trabalha diretamente sobre um recurso (classes Java como exemplo). O Eclipse segue um ciclo de carrega-altera-salva que somente se um editor salvar suas alterações, a plataforma será capaz de notificar as outras aplicações.

Views especiais podem ser conectadas diretamente a um editor, adicionando recursos complementares ao manuseamento dos recursos. Por exemplo a Outline View da perspectiva Java é conectada diretamente ao Editor Java.

O que torna o Eclipse uma IDE especial é a extrema flexibilidade na qual podem ser combinadas views e editores. Dessa forma o workbench pode ser arrumado de uma forma livre e que melhor adapte o desenvolvedor. As views e editores podem ser adicionados em uma perspectiva aberta (mesmo se eles foram definidos em um plugin totalmente diferente). Portanto é possível ter a total liberdade para criar o ambiente de desenvolvimento que melhor agrade ao desenvolvedor, de uma forma agradável e customizada.

# Perspectiva

Para abrir uma nova perspectiva no seu worbench faça:

1. Escolha 'Window -> Open Perspective' no menu principal:



2. Escolha 'Other' para ver todas as perspectivas instaladas:



# **Plugins**

Para instalar novas aplicações, simplesmente copie os plugins para dentro da pasta \$ECLIPSE\_HOME/plugins como mostrado com o Tomcat plugin. Será preciso reiniciar o Eclipse para tornar a nova aplicação "ativa". Dependendo plugin uma nova perspectiva poderá ser escolhida, sendo encontradas novas opções no menu e toolbar.



# Workspace

Como default a pasta root do workspace é \$ECLIPSE\_HOME/workspace. Caso seja necessário atuar em vários projetos, ou mesmo compartilhar uma máquina para armazenamento dos projetos, faz mais sentido separar o workspace. Pode ser escolhido um diretório arbitrário para o wokspace, para ter uma pasta específica, como root, basta usar a opção -data na inicialização do Eclipse. Por exemplo "eclipse -data c:\MeuWorkspace".

### Adicionando views para uma perspectiva

1. Abra a perspectiva, ex. perspectiva Java.

2. Escolha 'Show View -> Navigator' no menu:



A navigator view será adicionada na perspectiva Java.



### Arrumando o workbench

1. Clique na barra de titulo da view ou do editor e arraste para outra posição dentro do workbench

Same Science Partners	50 <b>3</b>
Control Andrease and a control of the second s	
12 Mar	All Small

2. A aparência do workbench será ajustada automaticamente:



Para adicionar uma view ou um editor como uma tab faça:

3. Clique na barra de título da view ou do editor e arraste até o cursor mudar dentro do símbolo da tab.



4. A aparência do workbench será ajustada automaticamente:

A conclution further and the set that the set of the se
the life of the second se
Control         Control <t< th=""></t<>
Participa restantive facilitation and a second seco

# Manipulando recursos com ferramentas externas

Se um recurso existente no workspace for alterado por uma ferramenta externa, a notificação não chegará ao Eclipse automaticamente. Mas se o recurso alterado externamente estiver aberto em um editor no Eclipse, será recebida uma notificação perguntando se é desejado carregar as alterações ocorridas.

🖶 He Changed		×
? The file has been changed on the file	e system. Do you want to k	ad the
A cranter,		
	Yes	5jo

Caso sejam criados arquivos ou pastas para o seu projeto externamente, será preciso notificar ao Eclipse:

- 1. Selecione a pasta na qual foram criados novos arquivos / pastas na view (ex. Navigator)
- 2. Escolha 'File > Refresh'



# Instalação

- Sistema operacional:
  - o Windows
  - o Linux
  - o Solaris
  - o QNX
  - Mac OS/X
- Recomendável 256 MB RAM.
- Java 2 runtime environment (JRE) ou Java 2 Software Development Kit (J2SDK).
- Eclipse necessita da versão 1.3 ou superior.
- Eclipse 2.0.2-archive ou superior.
  - 1. Instale o respectivo JRE.
  - Descompacte o Eclipse-archive para uma pasta arbitrária. No Windows o Eclipse acha a instalação do JRE/SQK automaticamente, através do registry.
  - Inicialize o Eclipse executando eclipse.exe (Windows) ou eclipse.sh (Linux) localizado no diretório de instalação. Após isso, o workbench aparecerá:

a dift turkper Seach	Popul Tonial Auto Welder Help	
·	8 V 9 *	
Nangetar III.	a Desire X	
	Eclipse Platform	
	The page with the familiaries you with the balance intertained. To get started, read the sectors below and dot on the relate the installed features	(84).
E Guilre An outhreis not available.	The Autom doc the final is indiced and work     all disc that is indiced and work     all disc that is an an and address     a make contrast one or more perpendients, is perpendie     all disc contrast one or more perpendients, is perpendie     all disc contrast one or more perpendients, is perpendie     all disc contrast one or more perpendients, is perpendie	e chatta de balbade y e chatta quese (el
) Outre An outre a not a slidek	the dust number doubter induces induction such and and induces the future induces induces.     Preparation, does and others     such as the output of the induces.     The dust is a such as the induces.	e constituti d' esset (s.) North, c'onstituti de la s
🖡 Guller An suffer Einst an Albin	the first source occurs framework in the second source occurs and source the second source occurs and source of the second source occurs and source occ	e constituir d'avec (s) - gan nes pergectes V = 3 + 1 (soft
E Guitre An outfine to constant	the out one doubt the future radiation and some end and the future radiation and some end and the future radiation and some and some future radiation and some and addiate the soliding addiates and the future radiation and the solidition and the the future radiation and the solidition and the the future radiation and the solid time and the the solid time radiation and the solid time and the the solid time radiation and the solid time and the the solid time radiation and the solid time and the solid time radiation and the solid time and the solid time and the time radiation and the solid time and the solid time and the time radiation and the solid time and the solid time and the time radiation and the solid time and the solid time and the solid time and the time radiation and the solid time and time and the solid time and time and the time radiation and the solid time and time a	<ul> <li>Constitution of street (section)</li> <li>Constitution of street (section)</li> <li>V = 2 + 3</li> <li>Constitution</li> </ul>
E Galler An address of a skille	The States of the States and	e constit d'avec (c) - gan rave pergection V = \$ + 1 - local

# Configuração Básica

Após a inicialização do Eclipse, serão necessárias algumas configurações da ferramenta.

# Estrutura das pastas nos projetos Java

Por padrão os arquivos fonte e compilados serão salvos diretamente na pasta do projeto. É melhor separar o código fonte do código compilado (arquivos .class). Com isso devemos ter uma pasta especial (source) que contem somente o código fonte.

Um projeto pode crescer consideravelmente através da adição de novas pastas de código fonte.

1. Escolha 'Window -> Preferences' no menu.

- 2. Expanda o nó 'Java' e escolha 'New Project'.
- 3. Escolha o radio-button 'Folders'.

-1985	Area Project		
Initial/Update     Apparative     Apparative     Code Formatter     Code Formatter     Code Generation     Compter     H Debug     Seture	Specify the daspath entry creation repart. As source and output local C growth Bollett Space Rider name: Codest location name:	s used as default by the New Ion use: Ion	3eaircy
<ul> <li>Protect Artis</li> <li>Protect Artis</li> <li>Protect</li> <li>Protect</li> <li>Properties</li> <li>Properties</li> <li>Properties</li> <li>Properties</li> <li>Properties</li> <li>Properties</li> </ul>	AL DE ROMY LAN	7638 value	
K TOPOR		Restore Onfailes	APERY.

Prescione 'Apply'.
 Prescione 'OK'.

### Selecionar compilador e editor para o JDK

O compilador Java e editor do Eclipse podem ser configurados para trabalhar com diferentes versões de JDK.

- 1. Escolha 'Window -> Preferences' no menu.
- 2. Expanda o nó 'Java' e escolha 'Compiler'.
- 3. Escolha 'JDK Compliance'.
- 4. Escolha '1.4' na lista de seleção 'Compile compliance level'.

	A Complex
E-Deny      E	Gran Fr P. Sex service: The first of a second is exact to salar darges which is: The main of the second is not a second in the second is a "Second in the conductive second is a "Second in the conductive second is a processing second in the second is a processing second is a second in the second is a processing second in the second is a processing second is a second in the second is a processing second in the second is a processing second is a second in the second is a processing second is a second in the second is a processing second is a second in the second is a processing second is a second in the second in the second is a second in the second in the second in the second is a second in the second in t
ii Tean	
a Tomat	
	Status Table 1
	PARTY DATA ALL ADDA

- 5. Pressione 'Apply'.
- 6. Pressione 'OK'.

Use JDK ao invés de JRE (ou instale um novo JRE/JDK)

Para ver o código fonte das classes padrão do Java, o Eclipse precisa ser "avisado" para usar o JDK correspondente. Por padrão o Eclipse utiliza o JRE, o qual não possui código fonte (o arquivo src.zip).

Ao usar o JDK, ajudas adicionais serão oferecidas: Code Assist pode mostrar nomes e retornos dos métodos existentes no JavaDoc e comentários através de um hover help.

Pré-requisito: O JDK precisa estar instalado no Sistema Operacional

- 1. Escolha 'Window -> Preferences' no menu.
- 2. Expanda o nó 'Java' e escolha 'Installed JREs'.
- 3. Escolha a primeira linha. (Standard VM).



4. Pressione o botão 'Edit' (ou 'Add'). O diálogo 'Edit JRE' (ou 'Add JRE') aparecerá:

🚳 Edit JRE		×
Rill type: Standard VM		×
FE game: Englished States		_
JFE home directory: C1(25dk1.4.0		growse
Jøgadoc URL:		Brogse
34E system Braies:		
C:(2:dt140)tn/b/t3#		UR:
		Quen.
		Benove
		Egterni 34ks
		mach Source
🕫 gan default system Brates	OK.	Canod

- 5. Pressione o botão 'Browse' na linha 'JRE home directory'.
- 6. Escolha a pasta no diretório de instalação no diálogo de seleção. No Windows deve ser C: \j2sdk1.4.1.
- 7. Prescione 'OK' para a pasta selecionada.
- Prescione 'OK' para o diálogo 'Edit JRE'.
   Prescione 'OK' para o diálogo 'Preferences'.

# **Configurando classpath variables**

Classpath variables são definidas a nível do workbench. Pastas e arquivos .jar ou .zip podem receber nomes, fazendo com que seja fácil incluí-los no buildpath.

1. Escolha 'Window -> Preferences'.

2. Expanda o nó 'Java' e escolha 'Classpath Variables'. O seguinte diálogo aparecerá:

Preferences		
Build Onder	*	Gangath Variables
Dellug Ecternal Took Help Svitabl.p.date 3era	Ì	A despath variable on be added to a project's dass path. It can be used to define the location of a bid the tract and part of the workspace. The reserved dass port vanishes IEL, IBL REL_SIG, IRE_SIGNOOT are set intensity depending on the XHS setting. Defined paragraph variables:
Appearant Encode Servator Code Servator Code Servator Code Servator Code Servator Code Servator Code Servator	:	+10.78,10         Hg.,           >91,2,3         Str.2,3           >91,2,3         Str.3,3           >91,2,3         Str.3,3           >91,2,3         Str.3,3           >91,2,3         Str.3,3           >91,2         Str.3,3           >91,2         Str.3,3           >91,2         Str.3,3           >91,2         Str.3,3           >91,3         Str.3,4           >91,4         Str.3,4           >91,4         Str.3,4
c x	ľ	Antoripriado apriv

3. Pressione o botão 'New'. O diá logo 'New Variable Entry' aparecerá:

New Variable Entry	X
Same:	
Eagl: 1	Be
	Cancel

- 4. Especifique o nome da variável no textfield 'Name'.
  - 5.
  - a) Se a variável apontar para um arquivo (.zip ou .jar), pressione o botão 'File'.
  - o b) Se a variável apontar para uma pasta, pressione o botão 'Folder'.

Variáveis apontam para uma pasta que deve complementar o path para dentro do projeto que irá utilizá-la. Isto significa que tanto pastas .zip ou .jar serão adicionados ao buildpath do projeto.

- 6. Pressione 'OK' no diálogo 'New Variable Entry'.
- 7. Pressione 'OK' no diálogo 'Preferences' Dialog.

# O primeiro projeto Java

Como criar um projeto Java simples, onde o código fonte esta separado na sua pasta de "saída'.

### Criação do projeto Java:

Não importa se será criado um novo projeto ou será importado um projeto existente, é necessário criar um projeto Java no Eclipse antes.

- 1. Abrir a perspectiva Java
- 2. Abrir o assistente 'New Java Project'.

Existem várias maneiras de iniciar esse assistente. A mais fácil é pressionando o botão na toolbar.

Também pode ser escolhido 'File -> New -> Project' para obter um diálogo que deixa escolher 'Java project'.

O diálogo 'New Java Project':

Lew	
lawa Project Orote a new Java project.	ij
Engectrianie MeriPropert	
Project contents:	
wards Pressent and the set	<u>n</u>

- 3. Escolha um nome para o seu projeto, ex. "MeuProjeto". O padrão para a pasta do projeto é workspace\_home/MeuProjeto .
- 4. Pressine 'Next'. O diálogo 'New Java Settings' aparecerá:

Com esse diálogo será editado o arquivo .classpath do projeto.

New	
Java Settings Define the Java build settings.	j
Source Source & grogects & Loranes 11 Order and Use the project as source folder C Use source folders contained in the project	b.port
	Greate New Folder
	Add Existing Folders
	Edt
	Semone
Build output folder:	
MeniFroyekt	Browse
< Back Sport >	Brith Canod

A tab 'Source' serve para especificar pastas que contenham recursos Java (arquivos .java).

- 5. Selecione o radio button 'Use source folders contained in the project'.
- 6. Pressione o botão 'Create New Folder'. O diálogo 'New Source Folder' aparecerá:

grear a path readive to "Memoropect"	
	OK Canot

- 7. Digite "src" no textfield.
- 8. Pressine 'OK'. O diálogo 'Source Folder Added' aparecerá:

Source Folder Added	<b>X</b>
OK to update build output folder to MenProjekt, bri?	
Yes	NP

9. Pressione 'Yes'. A pasta de saída para as classes compiladas será especificada como workspace\_home/MeuProjeto/bin

O diálogo 'New Java Settings' aparecerá:

New	
Jawa Settings Define the Jawa build settings.	J
Source Segregation (Linear Segregation) Segregation (Linear Segre	oport
@MeinProjekt/acc (new)	Greate New Folder
	Add Existing Folders
	Edt
	Bemove
Bugo output roler:	Recent
Lunn Lohndon I	
< Back. (jmd >	mah Canoel

10. Pressione 'Finish'.

# A primeira classe Java

Criação da classe Java:

- 1. Abra a perspectiva Java
- 2. Abra o assistente 'New Java Class'.

Existem várias formas de iniciar o assistente, a mais fácil é pressionando o

botão na toolbar.

Também pode ser escolhido 'File -> New -> Class' e será mostrado o dălogo para escolher 'Java project'.

O diálogo 'New Java Class' aparecerá:

Contraction of the local division of the loc	the distance of the second	Conservation of the server of
<ul> <li>Construction</li> <li>Const</li></ul>		iner a data
	W Table Peter Independent of Lines	1.2 .

O primeiro botão chamado 'Browse' serve para selecionar a pasta para os códigos fontes, no qual as classes serão criadas, nesse caso 'MeuProjeto/src'.

- 3. Digite "test" como o nome do package Java. (Package)
- 4. Digite "Test" como o nome da classe. (Name)
- Marque o checkbox 'public static void main(String[] args)' Podem ser especificadas superclasses e interfaces para implementação. Para o primeiro teste será criado um simples método main.

O diálogo será:

New		
Java Class Create a new Java	dam.	C
Source Folger: Package: If Enclosing type:	Menihopit (Jac	Broger. Broger.
Nage: Hodfes: Supercles:	Text ¥ipskic Coldagt Concor Concord Fakigast Final Finan; [peaking-biget	Browsg
provfaces:		835
which method stu	s wold you like to center? © public table word man(Cenrog[] ango) IF Constructors from supervises © Infgented abstract methods	
	Brith	Canod

6. Pressione 'Finish'

A nova classe será aberta no editor Java:

		i I I I I
5 - 13.0 Kdd	anges Seen Anget Tonie fun Weder Hep D + # + # ####### / 4 + 15	13
S Stateptown + + + + + + + + + + + + + + + + + + +	Autoritation and a second seco	B convert P 0 v = 0 0
	X	A
	Taka g-beng	
	101. Desident	Lower.
Padage Digter Herady		

7. Digite 'System.out.println("Hello world!");' no corpo do método main.

8. Salve as mudanças para a nova classe. (Ctrl+S ou no menu)

Assim que ocorrer o salvamento, o código fonte será compilado. (Essa ação pode ser alterada em preferences) Para executar a classe, será necessário configurar o launcher.

# Configurando o Launcher - Executando um programa

Para executar o método main, será preciso configurar o launcher:

Como atalho pode ser escolhido 'Run As - > Java Application' e no menu dropdown 'Run'. Isso é totalmente suficiente para executar um programa simples (sem parâmetros etc.)

1. Pressione o botão 'Run' na toolbar:



O diálogo 'Launch Configurations' aparecerá:



2. Selecione 'Java Application' e pressione 'New'. A segunda parte do diálogo 'Launch Configurations' aparecerá:

E Launch Camigurations		×
Create, manage, and run laun	In configurations	*
Lanch Configurations	Same Test.	
<ul> <li>In Jan NgRuden</li> <li>In Jan NgRuden</li> <li>In Jan</li> <li>In Jan</li> <li>In Jan</li> <li>In Jan</li> <li>In Jan</li> <li>In Jan</li> </ul>	Principer Arguments   B. 200   No. Geoports   District   Distrec   District   District   Distrect   District   District   Distr	green
Deg Deg	40% 8gn	Dow.

Várias configurações podem ser informadas (parâmetros para a VM, classpath, ações do workbench quando inicializado etc.).

3. Pressione 'Run'. A Console view da perspectiva Java aparecerá contendo a saída padrão:



Pressionando o botão 'Run', a última ação será executada novamente, nesse caso a classe 'Test'.

Será necessário criar uma nova configuração:

1. Escolha 'Run' no menu dropdown:



O seguinte diálogo 'Launch Configurations' aparecerá para completar a configuração:



Selecione 'Java Application':



Pressione 'New'. A nova classe aparecerá no diálogo:

🔤 Launch Canfiguration		8
Oreate, manage, and run laur	ch cariligerations	*
Lanch Confgantion: ■ Dava Agatetic = A Test = A Test = A Arts = A Arts = B Lancher wontench	Specie (1962) III Man (Her arguments (III 396) NL Charpeth (III Source) II Brownis (Herringets Species) Sectored Sectored Translate optimulties when sectored for a num class	green. green.
No. Dep	401	inger
	Non by	0.04

Pressione 'Run'.

# Importando um projeto existente

Para importar um projeto existente, será necessário:

- 1. Criar um projeto Java.
- 2. Escolha 'File -> Import'. O diálogo 'Import Select' aparecerá.
- 3. Selecione 'File System'.

Import	
Import resources from the local file system	Ľ
Select an import source:	
Existing Project into Workspace     External Plupine and Prognents     Recenters	
P, Zo file	
<56 (ML)	Cancel

- 4. Pressione 'Next'. O diálogo 'Import File System' aparecerá
- 5. Especifique o diretório que conterá o projeto importado com o primeiro botão 'Browse'. Esse diretório será o da raiz para o seguinte diálogo de seleção.
- 6. Selecione as pastas e arquivos que deverão ser importados para dentro do projeto.
- 7. Especifique a pasta que na qual serão armazenados os arquivos importados, com o segundo botão 'Browse'.

Poderão ser escolhidas pastas de projetos existentes e seus subdiretórios. A opção 'Create complete folder structure' também cria pastas hierarquicamente no projeto.

Import	
File system	
Inport resources from the local file system.	
Deathers ColorcectDeats	T Broom
Contraction of the second	
	(2) Test.an-a
Piter Spec. Select All Secolect All	
Fuller Net Provid	Broom_
- Antonio -	
Converte entire resources who average	
C Queste complete fuider structure	
(# Oreste agented hidlers only	
qat	SHCP [

8. Pressione 'Finish'. As pastas e arquivos selecionados serão adicionados ao projeto.

# Configurando o Java Buildpath

O buildpath avisa ao compilador onde procurar por vários arquivos e recursos para realizar o seu trabalho. Isso significa que devem ser compilados o código fonte e as classes definidas através das implementações. Essas informações são armazenadas no arquivo .classpath do projeto Java. Não é aconselhável editar esse arquivo diretamente, mas sim usar um assistente especial.

- 1. Selecione o projeto no Package Explorer ou Navigator view.
- 2. Abra o menu para esse projeto (right-click):
- 3. Escolha 'Properties' no menu:



O diálogo 'Properties for ProjektName' aparecerá.

4. Selecione 'Java Build Path'. O diálogo aparecerá:



5. Configure o buildpath com a seguinte descrição.

### Significado das várias tabs:

Source:



Aqui serão especificadas as pastas nas quais os códigos fonte serão armazenados. Somente nessas pastas novas classes e packages poderão ser criados. Por padrão novas classes serão criadas diretamente nessas pastas dos projetos. Para separar o código fonte dos arquivos compilados, precisa ser especificado aqui.

Projects:

245	Jevo Bulki Puth	
E-ternal Tuck buckets Deve build Path Deve build Path	Source Self-rights ( Journey ) to grant Encourse ( projects on the bull parts	andE-port
Tomost	Carlorg aparte combact	Select all
	Barry advoc contraction Barry advoc net white Barry advoc net white Barry Barron	Quelet #
	hat support hitler:	_
	meningeisten	trops.
	Bud subut tear: [wenitcpitton	tog

Aqui projetos existentes podem ser adicionados ao buildpath. O compilador procura por classes definidas nos projetos especificados.

### Libraries:



Referências para arquivos .jar e .zip dentro do workspace ou em qualquer lugar do seu HardDisk pode ser especificado aqui.

Detalhes:

- Add JARs: Arquivos que existem dentro de projetos no workspace serão adicionados aqui.
- Add External JARs: Arquivos arbitrários no seu HardDisk serão adicionados aqui.
- Add Variable: Adiciona uma classpath variable.
- Advanced: O diálogo auxilia a criação de pastas para classes. Essas pastas contem definições de classes descompactadas (arquivos .class, não .jar ou .zip).
- Edit: Muda entradas existentes.
- Attach Source: Adiciona o código fonte para uma referência existente. O código fonte será mostrado no editor Java, mas não poderá ser alterado.
- Remove: Seleciona uma entrada e remove do buildpath.
- Change the order of the entries. O compilador sempre utiliza a definição de classes que encontrar primeiro.
- Textfield 'Build output folder': A pasta que recebe as classes compiladas. Essa pasta não deve conter código fonte.

# Rodando em modo Debug

O Eclipse permite rodar programas em modo debug, ou seja, habilitar ferramentas como breakpoints e variable tracking.

# Debugger

O debugger permite detectar e diagnosticar erros em programas sendo executados localmente ou remotamente. Torna-se fácil controlar a execução dos programas, através da adição de breakpoints, os quais suspendem a inicialização, possibilitando analisar o código por "dentro" e examinar o conteúdo das variáveis.

A utiilzação dessa perspectiva é através de um design cliente/servidor, sendo assim os programas podem rodar remotamente em outros sistemas operacionais na rede, como rodariam localmente na estação de trabalho do desenvolvedor. O servidor para debugger roda na mesmo sistema operacional que o programa a ser analisado, o qual pode estar na maquina do desenvolvedor (local debugging) ou um sistema operacional que esteja acessível através da rede (remote debugging).

### BREAKPOINTS

Breakpoints são marcos colocados no programa, que avisam ao debugger onde parar. Quando o workbench esta rodando um programa e encontra um breakpoint, a execução e suspensa. O thread correspondente é suspenso (temporariamente para de rodar) permitindo que seja vista toda a stack para o thread em questão.

A execução é suspensa antes do statement que contém o breakpoint. Os conteúdos das variáveis e a stack podem ser checados nesse momento, além de realizar step over, step into em métodos ou classes, a execução rodará até encontrar o próximos breakpoint ou o fim do programa.

Os brakpoints podem ser desabilitados sem suspender a execução, e habilitados em outro momento.

#### REMOTE DEBUGGING

Remote debugging permite rodar aplicações em um sistema operacional e realizar o debug em outro sistema operacional. O sistema local roda o debugger, e o sistema remoto roda o debugging engine e o programa.

#### CONNECTION

O principal requerimento para remote debugging é acessar a maquina remota, na qual os arquivos onde ocorrerá o debug devem residir. Ao realizar o remote debugging em um programa, o debug engine daemon inicia escutando a conexão, após realizada pode ser feito o debug do programa.

### **Debug View**

Esta view permite o gerenciamento para realizar o debugging de um programa no workbench. Ela mostra uma janela com a stack de cada thread suspenso para cada tarefa que esteja realizando o debugging. Cada thread aparece como um nó da árvore.

Comandos da Debug View				
Botões da Toolbar	Comando	Descrição		
	Resume	Este comando executa o thread suspenso.		
00	Suspend	Este comando suspende o thread da tarefa, podendo navegar ou modificar o código, inspecionar os dados, etapas e assim por diante.		
	Terminate	Este comando termina a tarefa do debug selecionado.		
Somente no menu	Terminate & Remove	Este comando termina todas as tarefas de debug selecionadas e remove-as da janela.		
Somente no menu	Terminate All	Este comando termina todas as atividades iniciadas na janela.		
*	Disconnect	Este comando fecha todas as conexões das tarefas de debug		
selecionad	las quando exe	cutado remote debugging.		
-	Remove All Terminated Launches	Este comando limpa da janela todas as tarefas de debug terminadas.		
₹.	Step Into	Este comando executa um steps into no statement selecionado.		
<b>⊙</b>	Step Over	Este comando executa um step over no statement selecionado. A execução continuará até a próxima linha no caso do mesmo método, ou voltará para o método de origem, caso o método corrente tenha sido chamado por outro método. O cursor é posicionado no método e seleciona a linha.		
_12	Run to Return	Este comando sai do método corrente. Esta opção para a execução após sair do método corrente.		
X2	Show/Hide Qualified Names	Esta opção pode ser selecionada para mostrar ou ocultar qualified names.		
Somente no menu	Relaunch	Este comando reinicia a tarefa de debug selecionada.		
Somente no menu	Properties	Este comando mostra as propriedades do processo selecionado, também permitindo visualizar as linhas de comando do processo selecionado.		

# Realizando debugging nos programas

1. Abra alguma classe na perspectiva Java.

 Coloque o cursor na barra do lado esquerdo da janela na qual esta aberta a classe e de um duplo clique ao lado do statement desejado, adicionando um breakpoint.



Adição de um Breakpoint.

O breakpoint está em azul porque ainda não foi verificado, significando que a classe que o contém ainda não foi carregada pela Java VM.

3. Clique no botão Debug na toolbar.

4. Selecione a classe na caixa de diálogo e pressione Debug.

Launch Configurations Create, manage, and run laun	h configurations	×
Launch Configurations:	Main       Main	ce Search
Ne <u>w</u> Dele <u>t</u> e	Арр	<b>y</b> Reyert
	Deb	2g Close

5. Tão logo tenha adicionado um breakpoint, a perspectiva Debug abre automaticamente, e a execução é suspensa. Perceba que o processo ainda está ativo (não foi terminado) na Processes view. Outros threads ainda devem estar rodando.

<u>Eile E</u> dit <u>P</u> erspective P <u>r</u> oject <u>D</u> ebug <u>W</u> indow <u>H</u> elp	
🖺 🎋 Debug 🗙 🔪 Breakpoints 🗆 🛥 📲	⊧ J⁰ ×
🗄 🕨 🗉 📽 🖘 🔊 🖄 🗱 🔍 💿 junit.samples.VectorTest [line: 24] - setUp()	
Thread [main] (Suspended (breakpoint 🔺	
VectorTest(TestCase).runBare() line	
TestResult\$1.protect() line: 106	
TestResult.runProtected(Test, Prote	
Naviaster Processes Debug	
🔄 🔄 TestImplementorTest.java 🚺 VectorTest.java 🗙	
protected void setUp() {     (Frantus new Vector());	<b>_</b>
fFull= new Vector();	
<pre>o fFull.addElement(new Integer(1));</pre>	
fFull.addElement(new Integer(3));	-
	Þ
🖳 Console 🥖 🗙 🐼 Variables 🏭 🎘 🖉 Ox	s F x
n Full= Vector (id=29)	
capacityIncrement= 0	
elementLount= U	-
Tasks Console Search Variables	

O breakpoint agora esta verde, pois foi verificado pela Java VM.6. No editor na perspectiva Debug, selecione a linha onde esta o breakpoint e no menu selecione Inspect.

VectorText isua	
protected woid setUp() {	Inspect
fEmpty= new Vector();	Display
fFull= new Vector();	Run to Line
o fFull.addElement(new Inte	eger(r),
fFull.addElement(new Inte	eger(2));

A Variables view mostra os valores das variaveis no stack frame selecionado.



7. Pressione Step Over para executar a linha de código selecionada.



Se o programa não terminou totalmente após o termino do debugging, selecione Terminate no menu para o programa iniciado, na Processes view ou Debug view.

# **PLUGIN'S**

Além dos plug-ins como o JDT para editar, compilar e debugar aplicações, outros plug-ins estão disponíveis para suportar completamente o processo de desenvolvimento através de modelagem, automação de deployment, teste unitário, teste de performance, controlador de versão e gerencia de configuração.

Dica: Adicionando novas aplicações ao Eclipse

Para instalar novas aplicações, simplesmente copie os plugins para dentro da pasta \$ECLIPSE\_HOME/plugins. Será preciso reiniciar o Eclipse para tornar a nova aplicação "ativa". Dependendo do plugin uma nova perspectiva poderá ser escolhida, sendo encontradas novas opções no menu e toolbar.

### Lomboz

Esse plugin auxilia aos programadores Java realizarem build, teste e deploy das aplicações Java nos Servidores de aplicação J2EE.

Ele oferece uma série de funcionalidades, tais como:

- Editor de JSP com sintaxe highlighting e code assist
- Verificador de sintaxe JSP
- Wizards para geração de modulos Web e Ejb
- Wizards para geração de Ejb test clients
- Suporte a deployment de WAR e Ejb(Jar)
- Produtividade usando wizards e geradores de código
- Desenvolvimento integrado de EJB 1.1 e 2.0 com Xdoclet (Session, Entity and Message Driven Beans)

### Instalação

• Execute o unzip do arquivo lomboz.zip para o diretório <Eclipse\_Home>



- Será criado um novo diretório chamado com.objectlearn.jdt.j2ee no diretório plugin
- Agora inicialize o Eclipse

# Configuração

• Caixa de diálogo no menu - Preferences e expanda o item Java> New Project.

• As configurações devem estar igual à figura.

	New Project	
Easy Struts     Help     Install/Update     Java     ① Appearance     Classpath Variables     Code Formatter     Code Generation     Compiler     Debug     ① Editor     Installed JREs     Javadoc     JDepend     JUnit     New Project     Organize Imports     Refactoring     Task Tags     Lomboz	Specify the classpath entries used as default Source and output folder Project Eolders Source folder name: src Output folder name: bin As JRE library use: JRE container	t by the New Java Project creation wizard:

 A maioria dos Applications Servers utilizam o compilador Java padrão (javac), o mesmo é localizado em tools.jar que vem com a distribuição padrão do Java

Ant AntView Set Lomboz Preferences Build Order Checkclipse Preferences Checkstyle Enable jsp syntax checking ✓ DbEdit Restart server after deploy (Tomcat) Debug Lonboz Help Use application server classpath ✓ Install/Update Java Comboz JSP Editor Server Definitions Plug-In Development Team
Bestore Defaults   Applu

- Observer des part reinididades
  Update server classpath and deployment parameters O Lomboz verifica o classpath do projeto e as propriedades de deployment.
  Use application server classpath Utiliza o classpath definido na
- Use application server classpath Utiliza o classpath definido na configuração do server para inicia-lo e não o classpath do projeto.
   Evita problemas de class loaders

• Definindo o Application Server

Ð Workbench	Server Definitions	
<ul> <li>Ant</li> <li>AntView</li> <li>Build Order</li> <li>Checkclipse Preferences</li> <li>Checkstyle</li> <li>DbEdit</li> <li>Debug</li> <li>Easy Struts</li> <li>Help</li> <li>Install/Update</li> <li>Java</li> <li>Jorden JSP Editor</li> <li>Server Definitions</li> <li>Plug-In Development</li> <li>Team</li> <li>XML</li> </ul>	Server types:       JBOSS 3.0 ALL         Properties       Classpath         JBOSS 3.0 ALL       Application Server Directory:         C:/Programas/jboss-3.0.6       .         Address:       127.0.0.1         Port:       8080         Classpath Variable Name:       JBOSS 300         Classpath Variable:       C:/Programas/jboss-3.0.6         Restore Defaults	
Import Export	ОК	Cancel

- Classpath variable O caminho que será usado para armazenar a variável. As java libraries serão um caminho relativo
- Server Home Directory O caminho para o diretório de instalação do Application Server

o Classpath Page



- Java libraries que serão utilizadas para iniciar o Application Server
- Server Classpath é utilizado para iniciar o Application Server
- Client path utilizado por aplicações cliente (ex.: Ejb Test Clients )

# Ativação

Adicionar ações, wizards e views para o Eclipse.

Customize Perspective		×
Select the items to be displayed in the current perspect Available [tems:	Ne (Java). Details:	
		OK Cancel

- Selecione no menu 'Window>Customize Perspective...', expanda 'File>New' e marque todos os itens relacionados ao Lomboz.
- Selectione no menu Window>Customize Perspective...', expanda 'Other' e marque 'Lomboz Actions'
- Selectione no menu Window>Customize Perspective...', expanda 'Window>Show View' e marque 'Lomboz J2EE View'
- Após essas seleções, será adicionado um novo botão a toolbar

(- )	ava -	Eclipse	Proje	ct SDK		
File	Edit	Source	Ref	actor 1	Vavigate	e Proj
	•	日風	₽(			10
6	-	Packages	7	* ×		
	4	中 分	1			
ゆ 同	0	° o				

 Novos wizards estarão disponíveis no diálogo New Project e no item de menu New



# Eclipse e Junit

Através do uso do Eclipse com o Junit, torna-se fácil desenvolver códigos de boa qualidade.

- 1. Download JUnit em www.junit.org. Descompacte src.jar a partir do arquivo .zip.
- 2. Crie um novo projeto Java Junit.
- 3. Escolha 'File->Import-> Zip file-> Browse-> src.jar .



4. Selecione a pasta src (Pressione o botão 'Browse'):



- 5. Pressione Finish.
- 6. Suporte opcional ao JavaDoc: Selecione o caminho para o JavaDoc Path em 'Preferences':

a similardy	Jevador
Dad one cohig Cohing Cohing International International International Constant Const	lands yn awstrin of the solaid someni tothe and in the bosin cano some i soleit own i ander and an i genetic cannon fi {*2mm. genetic cannon fi {*2mm.

e selecione a documentação em junit3.8.1.zip :

tale Descent have Rubbert Second Park Second Park	Investigation interactions baseds for examine 2012 of the property interaction decomposition because acquire transmission of the two baseds based baseds in the composition of the two baseds based on the based of the composition of the two based on the two baseds in the composition of the two based on the two baseds of the composition of the two based on the two baseds of the composition of the two based on the two based on the two based on the two based on the two based on the two based on the two of the two based on t	The follows is used for a swelling attain, the
Propert Californian	Analy Later ( No. 10, 00, (and 12, 1), and	(Braget)
		1000

O local no qual o arquivo .zip foi extraído.

# Configurando o projeto

1. Adicione o projeto Junit ao 'Java build path' na 'Properties' do projeto:

	O X O X	
junit	New	•
~~~	Keepbee With	
	Rectore From Local Htich	D19
	Francisco	
	Properties 6	
	Properties 🔓	
	Properties R	
ackage Explo	Freportes 22	
ackage Explo	Freportes R	
ackage Explo	Freportics Ref Hierarchy	
Package Explo	Freparties	( 11 gete entryen)

# O primeiro teste

O primeiro teste é para uma instancia da classe AmpelModell que deve estar no estado vermelho. O estado dessa instancia é acessado por três métodos getters, getRed(), getYellow() e getGreen() cada um retornando valores do tipo boolean.

1. Criar uma nova classe AmpelModelITest:

	-	9 9	
	Cr	eate a ]	Java Class
101			<b>13</b>
sa Class	222		2
ALC: NO. 1 YO	000	(	0
week Publics	engethodelips;	. Rev	- 1
ethager	angemaded		
Declaring type:	1		- 1 C
and i	Falls Callet From	C. comment	
	Cabrez Citral Com		
owners .	prov. lang. Object	(bere	
rantalasi		- 441	ter al la constante de la const
			1.1
Ind rested state	e multi you Re to creater	100	120
	addic static void name(comp@wpi)		
	Constructions from supervision		
	and the second sec		

2. Selecione junit.framework.TestCase as superclass ('Browse'):

Courses (Genete)	138
Choose a type:	
Text	
Hatching types:	
TestSecuptor     TestSecuptor     TestSecuptor     TestSetM     TestSetM     TestSetAre     TestSetAre     TestSetAp     TestSetAp     TestSetAp     TestSetAp     TestSetAp     TestSetAp	
Qualifierr	
Bant, hanevork - Juntáre,	
OK	Canad

3. Pressione Finish.

Eclipse cria uma nova classe:

instantial and X	
package ampeimodel,	
1	
Element side Raisenauk, Teattank	
4	
Spatialization Ampaintonial/Tagt calcosts TextCale (	
7)	
x.	

Métodos que contenham os testes serão adicionados. JUnit procura automaticamente por métodos que iniciam com 'test':

```
public class AmpelModellTest extends TestCase {
public void testNewAmpel() {
 AmpelModell a = new AmpelModell();
 assertTrue(a.getRed());
 assertFalse(a.getYellow());
 assertFalse(a.getGreen());
 }
}
public class AmpelModell {
public boolean getRed() {
 return false;
 }
public boolean getYellow() {
 return false;
 }
public boolean getGreen() {
```

```
return false;
}
```

Executando o primeiro teste:

1. Escolha 'Run As->Unit Test' no menu:



2. A view Junit aparecerá. A barra vermelha indica que algo deu errado:



É mostrado o total dos testes executados, as exceções lançadas e o testes que falharam. Tão logo algum teste falhe, a barra de status aparecerá vermelha. A área do meio da view Junit tem duas abas.

- o 'Failures'
  - Lista os testes que falharam.
- o 'Hierarchy'
  - Mostra a visão geral de todos os testes que foram executados e é extremamente usual quando executando um conjunto de testes(TestSuite).

A view abaixo mostra o 'Failure Trace'. Pode ser visto a call stack da falha do teste. Um clique duplo localiza o código fonte onde foi ocorrido a falha.

