



O Ambiente de Desenvolvimento Eclipse

João Magalhães
magalha@mindsatwork.com.br



Agenda de Hoje

- Introdução
 - O que é o Eclipse?
 - Instalação do Eclipse
 - A idéia dos plugins
 - Licença de uso
- Conceitos importantes
 - Visões, perspectivas
 - Workspace
 - Projetos, recursos
 - Armazenamento dos projetos
- O code assistant



Agenda de Hoje

- Importação de recursos
- Execução de aplicações
- Configurações de projeto
 - O classpath
 - Níveis de compilação
- Debug no Eclipse
- Deployment de aplicações
- Reengenharia de Código

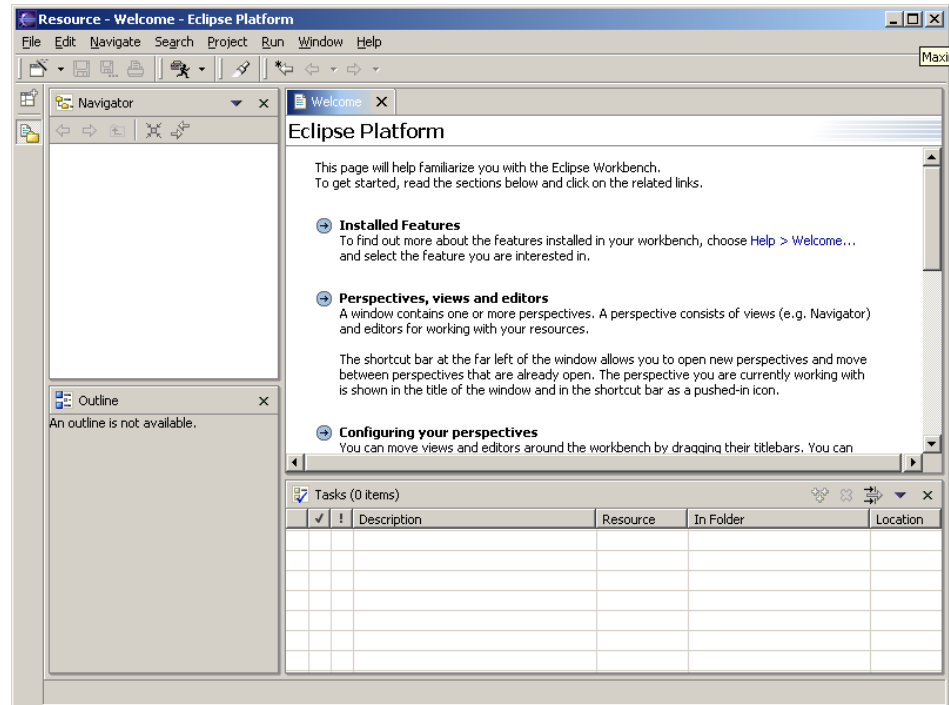


Agenda de Hoje

- Configurações gerais do ambiente
- Desenvolvimento em equipe
 - Suporte ao CVS

Conceitos Importantes

- Eclipse é um framework para integração de diferentes tipos de aplicação.
 - É um “meta-ambiente de desenvolvimento de software”, um “ambiente para construção de ambientes”.





Conceitos Importantes

- Uma dessas aplicações é o Java IDE-JDT (Java Development Tooling) que já vem com a plataforma Eclipse.
 - Isso significa que Java já é suportado pelo meta-ambiente



Instalando o Eclipse

- O eclipse pode ser obtido no site www.eclipse.org. Para executá-lo, é preciso uma máquina virtual instalada
 - Basta obter na Sun ou IBM
 - A versão mais recente é a 1.4.2



Instalando o Eclipse

- Existem versões de eclipse para diversas plataformas
 - Linux
 - Solaris
 - QNX
 - Windows
 - MAC/OS



Instalando o Eclipse

- Basta descompactar em um diretório e executar o arquivo “eclipse”.



Conceitos Importantes

- As aplicações são desenvolvidas e instaladas no Eclipse sob a forma de **plugins**. Plugins são automaticamente reconhecidos e integrados à plataforma.
 - São os plugins os reais responsáveis pelas funcionalidades do ambiente



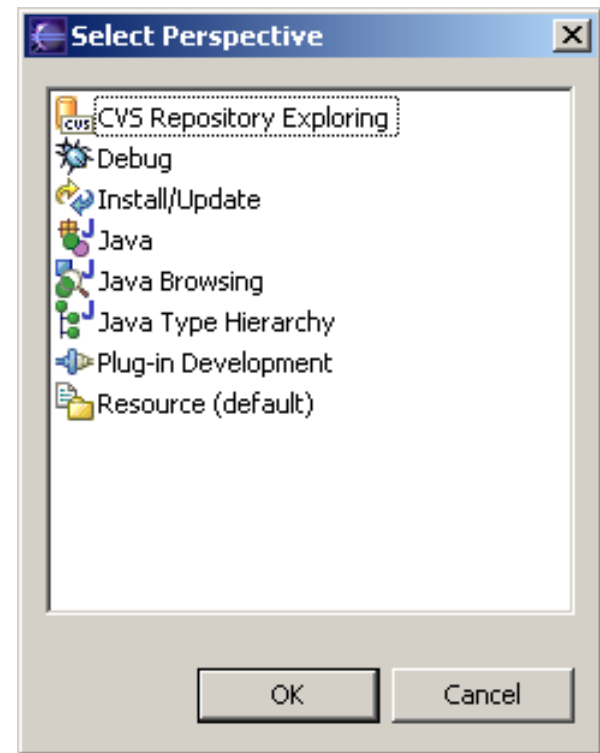
Conceitos Importantes

- Copy-and-use

- Para instalar novas aplicações, basta copiar os plugins para a pasta `$ECLIPSE_HOME/plugins` e reiniciar o Eclipse.
- Dependendo do plugin, novas **perspectivas** ou opções no menu estarão disponíveis. Em teoria, a documentação do seu plugin explicará tudo!

Conceitos Importantes

- Uma **perspectiva** define um conjunto de **editores** e **visões** organizadas de uma forma visual tal que auxilie o trabalho de um determinado papel (projetista, designer, etc).





Conceitos Importantes

- **Editores** permitem a criação e a alteração de **recursos**, e estão associadas a determinados tipos de **recursos**.
 - Editores para JSP, Java, XML, etc



Conceitos Importantes

- Uma **visão** provê uma forma de navegar através de **recursos** ou outras informações.

Conceitos Importantes

The screenshot shows the Eclipse IDE interface. On the left is the **Package Explorer** view, displaying a project named 'Ebts' with a package structure including 'br.com.modulo.ebts' and its sub-packages. The 'EbtsException.java' file is selected. In the center is the **Editor** view, showing the source code of 'EbtsException.java', which is a Java class extending 'Exception'. On the right is the **Tasks** view, which is currently empty, showing a table with columns for 'Description', 'Resource', and 'In Folder'. The bottom status bar indicates the current view is 'Package Explorer'.



Conceitos Importantes

- **Recursos** geralmente são arquivos no HD. Eles ficam no **workspace**, uma pasta especial no sistema de arquivos.
- O **workspace** é o local onde ficam os **recursos**, organizados em **projetos**



Conceitos Importantes

- O **workspace** é compartilhado por todas as aplicações no Eclipse. Sempre que ocorre uma mudança em um **recurso** do **workspace**, todas as aplicações abertas são avisadas.



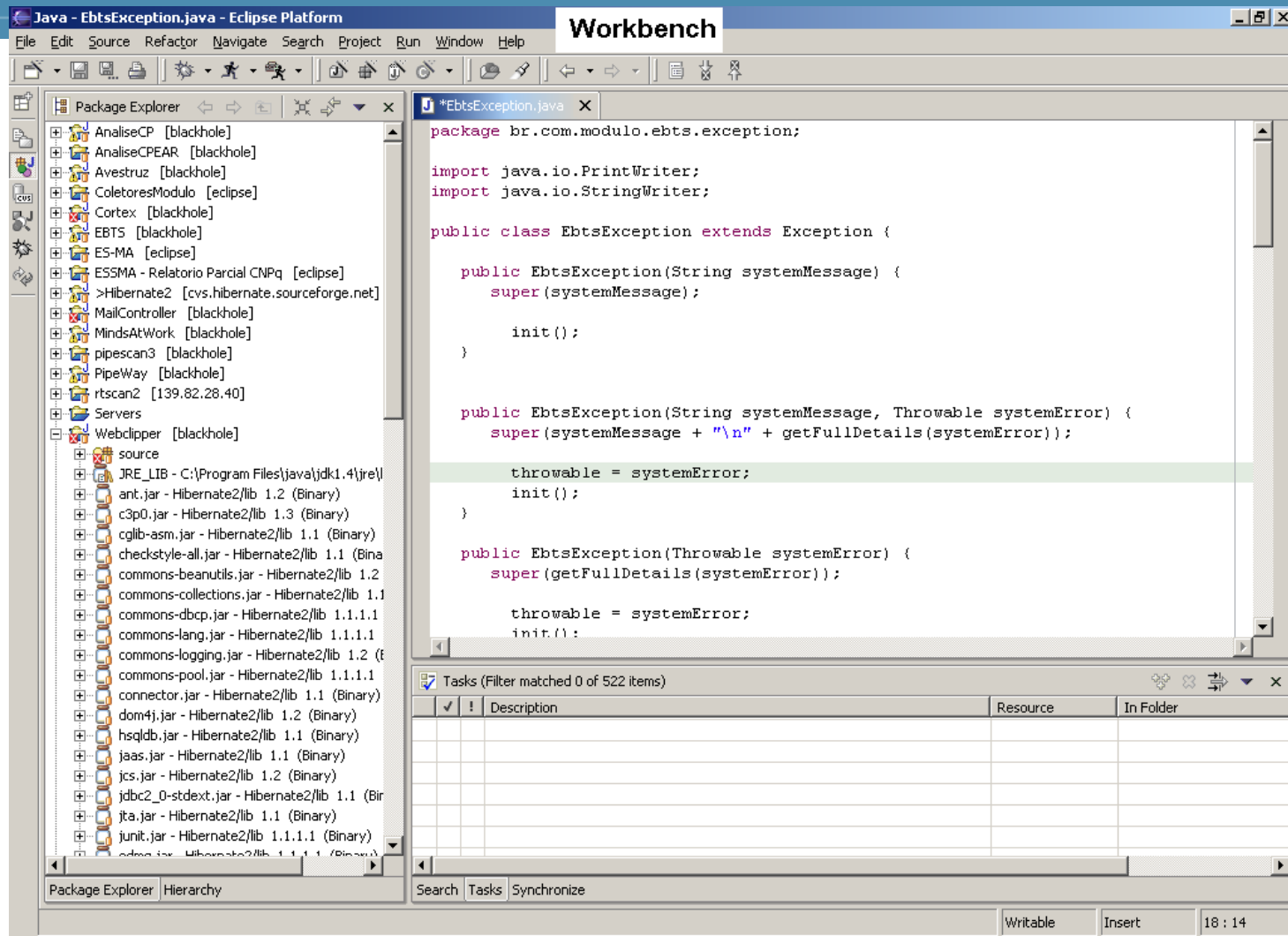
- Um usuário sempre trabalha com o **workbench**, que é a parte visível da GUI do eclipse.
- A **perspectiva** selecionada definirá o que estará visível.



Conceitos Importantes

- Perspectivas mais utilizadas:
 - **Java**: boa para geração de código em Java
 - **Java browsing**: boa para geração de código em Java
 - **Debug**: boa para debug
 - **CVS**: trabalho em equipe
 - **Recursos**: exibe todos os recursos do workspace em um único local

Perspectiva Java





Perspectiva Java Browsing

Java Browsing - TemporalSemaphore.class - IBM WebSphere Studio Application Developer

File Edit Source Refactor Navigate Search Project Run Window Help

Projects Packages Types Members

Charsets.jar - E:\Arquivos de progr...
sunjce_provider.jar - E:\Arquivos d...
dnsns.jar - E:\Arquivos de program...
ldapsec.jar - E:\Arquivos de progra...
localedata.jar - E:\Arquivos de pro...
classes12.jar 1.3 (Binary)
minds-utils.jar 1.2 (Binary)

br.com.mindsatwork.jdbc
br.com.mindsatwork.sync
br.com.mindsatwork.util
br.com.modulo.ebts
br.com.modulo.ebts.admin
br.com.modulo.ebts.client
br.com.modulo.ebts.exception

Semaphore
TemporalSemaphore

TemporalSemaphore()
TemporalSemaphore(int)
TemporalSemaphore(int, String)
TemporalSemaphore(String)
waitFor(long)

Class File Editor

Source not found

The jar file minds-utils.jar has no source attachment.
You can attach the source by clicking Attach Source below:

Attach Source...

```
public class br.com.mindsatwork.sync.TemporalSemaphore extends br.com.mindsatwork.sync.Semaphore {  
    public br.com.mindsatwork.sync.TemporalSemaphore();  
    public br.com.mindsatwork.sync.TemporalSemaphore(int arg);  
    public br.com.mindsatwork.sync.TemporalSemaphore(int arg, java.lang.String arg);  
    public br.com.mindsatwork.sync.TemporalSemaphore(java.lang.String arg);  
    public void waitFor(long arg);  
}
```

Read Only Insert 1 : 1



Perspectiva CVS

CVS Repository Exploring - C1HTMLDoc.java - IBM WebSphere Studio Application Developer

File Edit Source Refactor Navigate Search Project Run Window Help

CVS Repositories

- :extssh:magalha@139.82.28.40:/home/cvs
- :pserver:anonymous@cvs.hibernate.sourceforge.net:/cvs
- :pserver:magalha@blackhole.teccomm.les.inf.puc-rio.br
- :pserver:magalha@blackhole.teccomm.les.inf.puc-rio.br

HEAD

- AdaptiveCompressor
- AnaliseCP
- AnaliseCPEAR
- Avestruz
- BackOfficeEquifax
- BioGenie
- Blk1ToBlk2
- CheckupTool
- Cortex
- CVSROOT
- DB2Tools
- EComercial
- Equifax
- estrutura-comercial-ejb
- estrutura-comercial-web
- FAP-CP-SF
- FAP-CP-SF-Decoder
- FAP-CP-SF-Server
- Joao_pessoal
- MailController
- ObjectCircuits
- Old
- Pipescan
- pipescan3
- PipeWay
- PipewayEAR
- Webclipper
- WebclipperAdmin
- WebclipperAdminWeb
- Webclipper Enviador
- Webclipper-old
- Webclipper RoboBusca
- Win tools

```
/*
 * Copyright 2003 Minds at Work. All rights reserved.
 * MINDS AT WORK PROPRIETARY/CONFIDENTIAL. Use is subject to license terms.
 */

package br.com.mindsatwork.scanner;

import java.net.URL;
import java.util.Collection;

/**
 * Interface que modela um HTMLDoc resultante do processamento de um document
 * necessários para que subclasses possam ser utilizadas pelo motor de busca.
 *
 * @author João Magalhães
 * @version 1.1 - 10/07/2001
 */
public interface C1HTMLDoc
{
    /**
     * Obtém a próxima URL oriunda de um HRef.
     */
    @return A próxima URL oriunda de um HRef. Se não houver mais URLs, ret
    /**
     * Obtém a próxima URL oriunda de um Frame.
     */
    public String getNextHRef();
}
```

Tasks (558 items)

✓ !	Description	Resource	In F
⚠	The method logEvent(String) from the type ILog is deprecated	CIDBAssembler.java	Wel
⚠	The method logEvent(String) from the type ILog is deprecated	CIDBAssembler.java	Wel
⚠	The method logEvent(String) from the type ILog is deprecated	CIDBAssembler.java	Wel
⚠	The method logEvent(String) from the type ILog is deprecated	CIDBAssembler.java	Wel
⚠	The method logEvent(String) from the type ILog is deprecated	CIDBAssembler.java	Wel
⚠	The method logEvent(String) from the type ILog is deprecated	CIDBAssembler.java	Wel

CVS Resource History Tasks

1 item selected



Perspectiva de DEBUG

Debug - Backdoor.Asoxy.txt - IBM WebSphere Studio Application Developer

File Edit Navigate Search Project Run Window Help

Debug

UserManager [Java Application]
E:\Arquivos de programas\Java\jdk\bin\javaw.exe (12/11/03 10:59 PM)

Variables

Variables Registers Storage Storage Mapping Monitors Modules Display Breakpoints Expressions

CIHTMLD... Documen... CISimple... Scanner... TODO.txt CIDBAss... CIWebSe... 06h00m... 07h30m... 14h00m... W32.HLL... Back...

Assunto Backdoor.Asoxy
Descoberto em 2003-11-26 00:00:00.0
Atualizado em 2003-11-26 20:17:13.0
Última Modificação Análise inicial.
Change Log 2003.11.26: Análise inicial.
Sumário Backdoor.Asoxy permite que um sistema comprometido seja usado como um proxy HTTP.
MCID 2298
Risco Muito baixo
Poder de Devastação Baixa
Severidade 1.3
Impacto 2.6

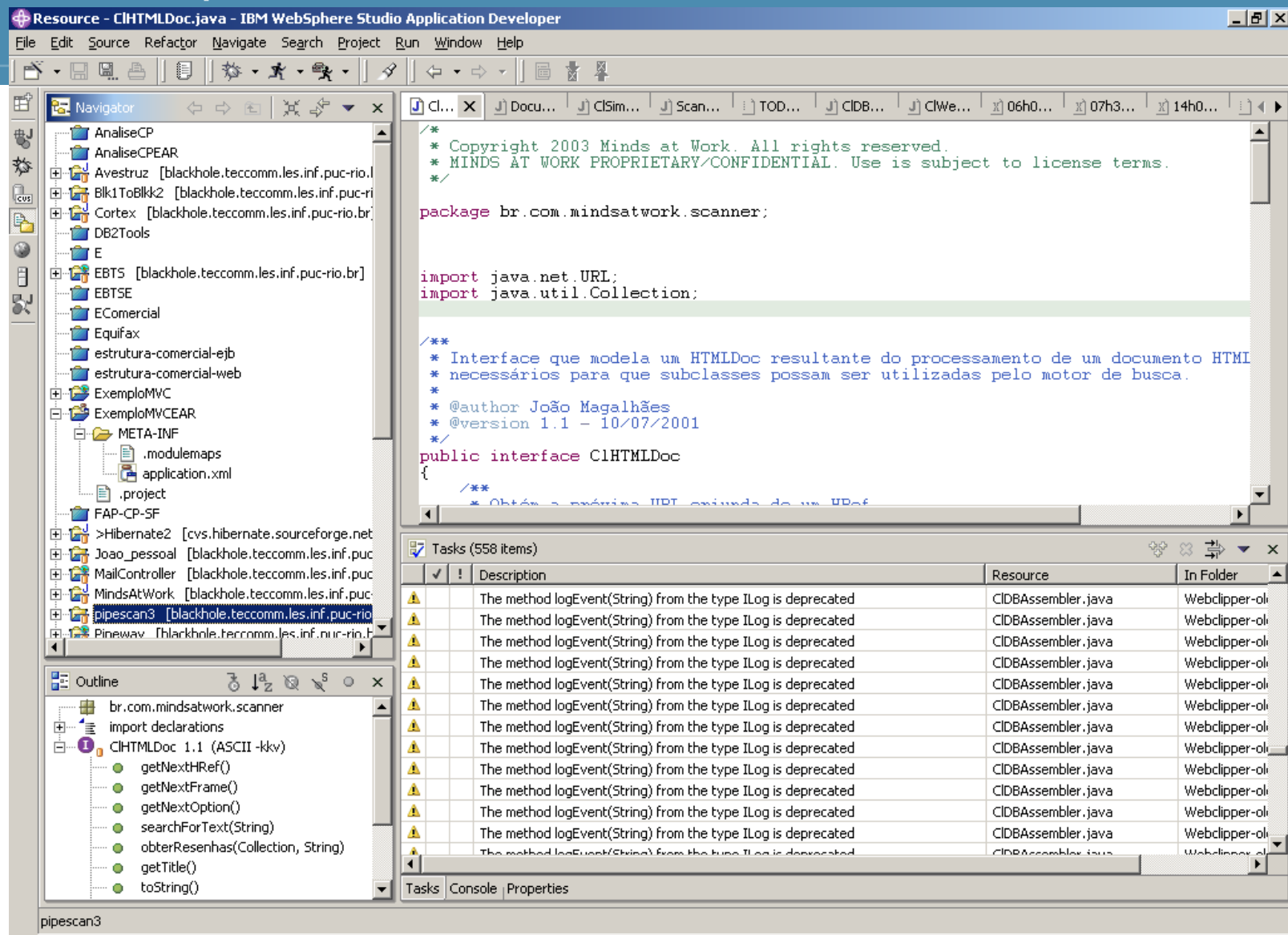
Console [E:\Arquivos de programas\Java\jdk\bin\javaw.exe (12/11/03 10:59 PM)]

Console Search Tasks

Writable Insert 1 : 1



Perspectiva de Recursos





Conceitos Importantes

- O **workspace** do eclipse é baseado em **projetos**.
- Um **projeto** é um agrupamento de **recursos**.
- Um **recurso** é um **arquivo**.



Conceitos Importantes

Java - C1HTMLDoc.java - IBM WebSphere Studio Application Developer

File Edit Source Refactor Navigate Search Project Profile Run Window Help

Package Explorer

- Avestruz [blackhole.teccomm.les.inf.puc-rio.br]
 - source
 - br.com.mindsatwork.avestruz
 - CapacitySmartStorage.java 1.4 (ASCII -kkv)
 - GroupSender.java 1.3 (ASCII -kkv)
 - HtmlSender.java 1.10 (ASCII -kkv)
 - Link.java 1.5 (ASCII -kkv)
 - LinkStorageStrategy.java 1.4 (ASCII -kkv)
 - PagesKeeper.java 1.5 (ASCII -kkv)
 - Pesquisa.java 1.7 (ASCII -kkv)
 - Scanner.java 1.17 (ASCII -kkv)
 - SearchEntry.java 1.1 (ASCII -kkv)
 - SelectiveSender.java 1.5 (ASCII -kkv)
 - Sender.java 1.4 (ASCII -kkv)
 - SmartStorage.java 1.4 (ASCII -kkv)
 - ThreadStopper.java 1.1 (ASCII -kkv)
 - UnableToConnectException.java 1.3 (ASCII -kkv)
 - br.com.mindsatwork.avestruz.doc
 - br.com.mindsatwork.avestruz.unused
 - br.com.mindsatwork.scanner
 - JRE_LIB - E:\Arquivos de programas\Java\jdk\jre\lib
 - WAS_PLUGIN\lib\j2ee.jar - E:\Arquivos de programas\IBM\WebSphere Studio Application Developer\lib\j2ee.jar
 - activation.jar - MindsAtWork/lib 1.1 (Binary)
 - HttpClient.jar - MindsAtWork/lib 1.1 (Binary)
 - mail.jar - MindsAtWork/lib 1.2 (Binary)
 - commons-logging.jar - MindsAtWork/lib 1.1 (Binary)
 - log4j.jar - MindsAtWork/lib 1.1 (Binary)
 - conf
 - design
 - docs
- Blk1ToBlk2 [blackhole.teccomm.les.inf.puc-rio.br]
- Cortex [blackhole.teccomm.les.inf.puc-rio.br]
- EBTS [blackhole.teccomm.les.inf.puc-rio.br]
- ExemploMVC
- ExemploMVCEAR
- >Hibernate2 [cvs.hibernate.sourceforge.net]
- Java - pessoal [blackhole.teccomm.les.inf.puc-rio.br]

Code Editor

```
/*
 * Copyright 2003 Minds at Work. All rights reserved.
 * MINDS AT WORK PROPRIETARY/CONFIDENTIAL. Use is subject to license terms.
 */

package br.com.mindsatwork.scanner;

import java.net.URL;
import java.util.Collection;

/**
 * Interface que modela um HTMLDoc resultante do processamento de um documento
 * necessários para que subclasses possam ser utilizadas pelo motor de busca.
 *
 * @author João Magalhães
 * @version 1.1 - 10/07/2001
 */
public interface C1HTMLDoc
{
    /**
     * Obtém a próxima URL oriunda de um HRef.
     *
     * @return A próxima URL oriunda de um HRef. Se não houver mais URLs, retorna null.
     */
    public String getNextHref();
}
```

Search

Package Explorer Hierarchy Console Styles Tasks Search Synchronize

Writable Insert 14 : 4

Recursos

Projetos



Prática

- Criando um projeto
 - Vamos criar agora um projeto no Eclipse.



Conceitos Importantes

The screenshot shows the Eclipse IDE interface. The 'New' menu is open, displaying options like Project..., Package, Class, Interface, Source Folder, Scrapbook Page, Folder, File, and Other... The 'Welcome' page is visible in the center, providing information about the Eclipse Workbench and installed features. The 'Tasks' list at the bottom shows several missing required libraries for the 'Webclipper' resource.

Java - Welcome - Eclipse Platform

File Edit Navigate Search Project Run Window Help

New

- Close Ctrl+F4
- Close All Ctrl+Shift+F4
- Save Ctrl+S
- Save As...
- Save All Ctrl+Shift+S
- Revert
- Move...
- Rename...
- Refresh
- Print... Ctrl+P
- Import...
- Export...
- Properties Alt+Enter
- Exit

Project...

- Package
- Class
- Interface
- Source Folder
- Scrapbook Page
- Folder
- File
- Other...

Welcome

This page will help familiarize you with the Eclipse Workbench. To get started, read the sections below and click on the related links.

Installed Features

To find out more about the features installed in your workbench, choose [Help > Welcome...](#) and select the feature you are interested in.

Perspectives, views and editors

A window contains one or more perspectives. A perspective consists of views (e.g. Navigator) and editors for working with your resources.

The shortcut bar at the far left of the window allows you to open new perspectives and move between perspectives that are already open. The perspective you are currently working with is shown in the title of the window and in the shortcut bar as a pushed-in icon.

Configuring your perspectives

You can move views and editors around the workbench by dragging their titlebars. You can also add more views to your current perspective by using [Window > Show View](#). To reset the perspective to its original state, choose [Window > Reset Perspective](#).

Once you have arranged your perspective, you can save it using [Window > Save Perspective As...](#). You can customize the views, perspectives and New menu operations that show up for your perspective. To do this choose [Window > Customize Perspective...](#). A view can additionally be converted into a Fast View by dragging it to the shortcut bar (at the far left of the window).

Join a CVS team

To join a CVS development team, open the [CVS Repository Exploring perspective](#). Next, specify

Tasks (132 items)

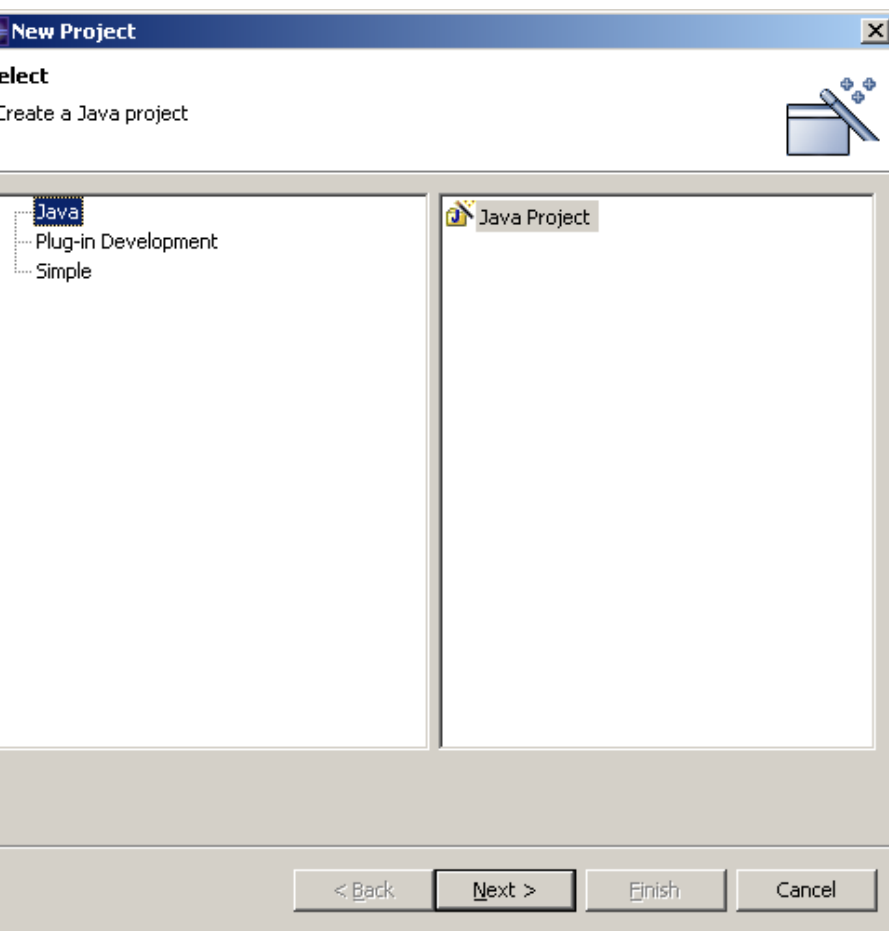
	✓	!	Description	Resource	In Folder	Location
✗			Missing required library: 'Hibernate2/lib/commons-beanutils.jar'.	Webclipper		Build
✗			Missing required library: 'Hibernate2/lib/commons-collections.jar'.	Webclipper		Build
✗			Missing required library: 'Hibernate2/lib/commons-dbc.jar'.	Webclipper		Build
✗			Missing required library: 'Hibernate2/lib/commons-lang.jar'.	Webclipper		Build
✗			Missing required library: 'Hibernate2/lib/commons-logging.jar'.	Webclipper		Build
✗			Missing required library: 'Hibernate2/lib/commons-pool.jar'.	Webclipper		Build

Package Explorer Hierarchy



Conceitos Importantes

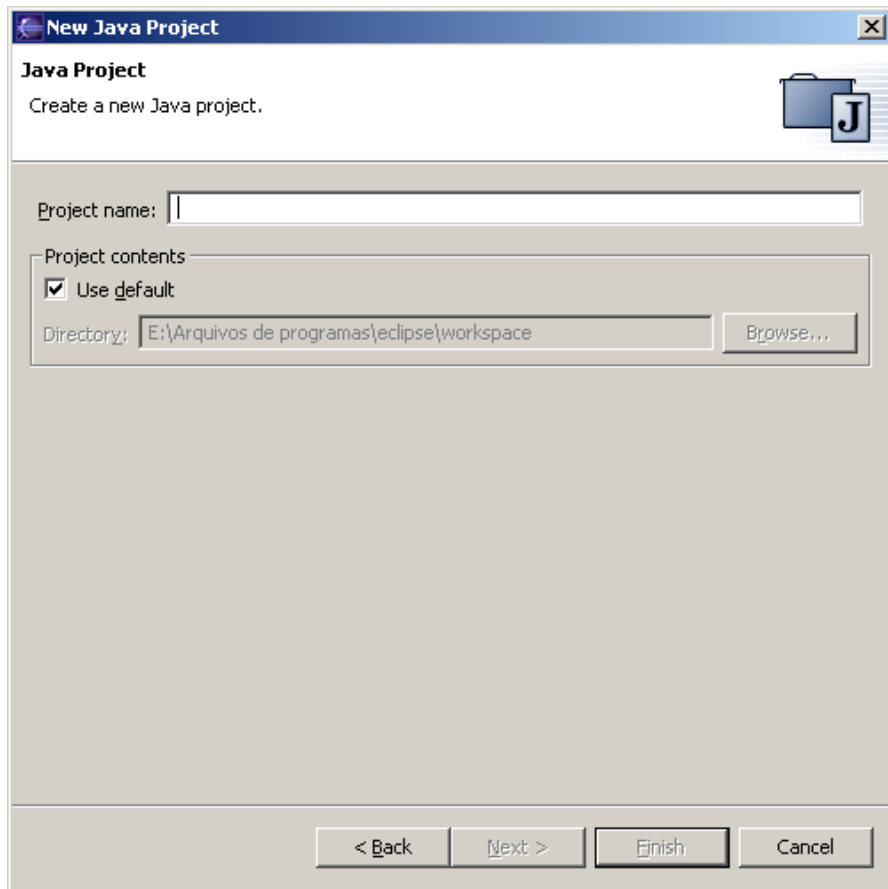
- Existem 3 tipos de projeto:
 - Projeto Java: projeto de uma aplicação stand-alone Java
 - Projeto Simples: projeto não Java
 - Desenvolvimento de plugin: para desenvolver plugins para o eclipse
- Vamos criar, inicialmente, um projeto Java
 - Perspectiva Java



- Nesta janela seleciona-se o tipo de projeto. Vamos iniciar por um projeto Java



Conceitos Importantes



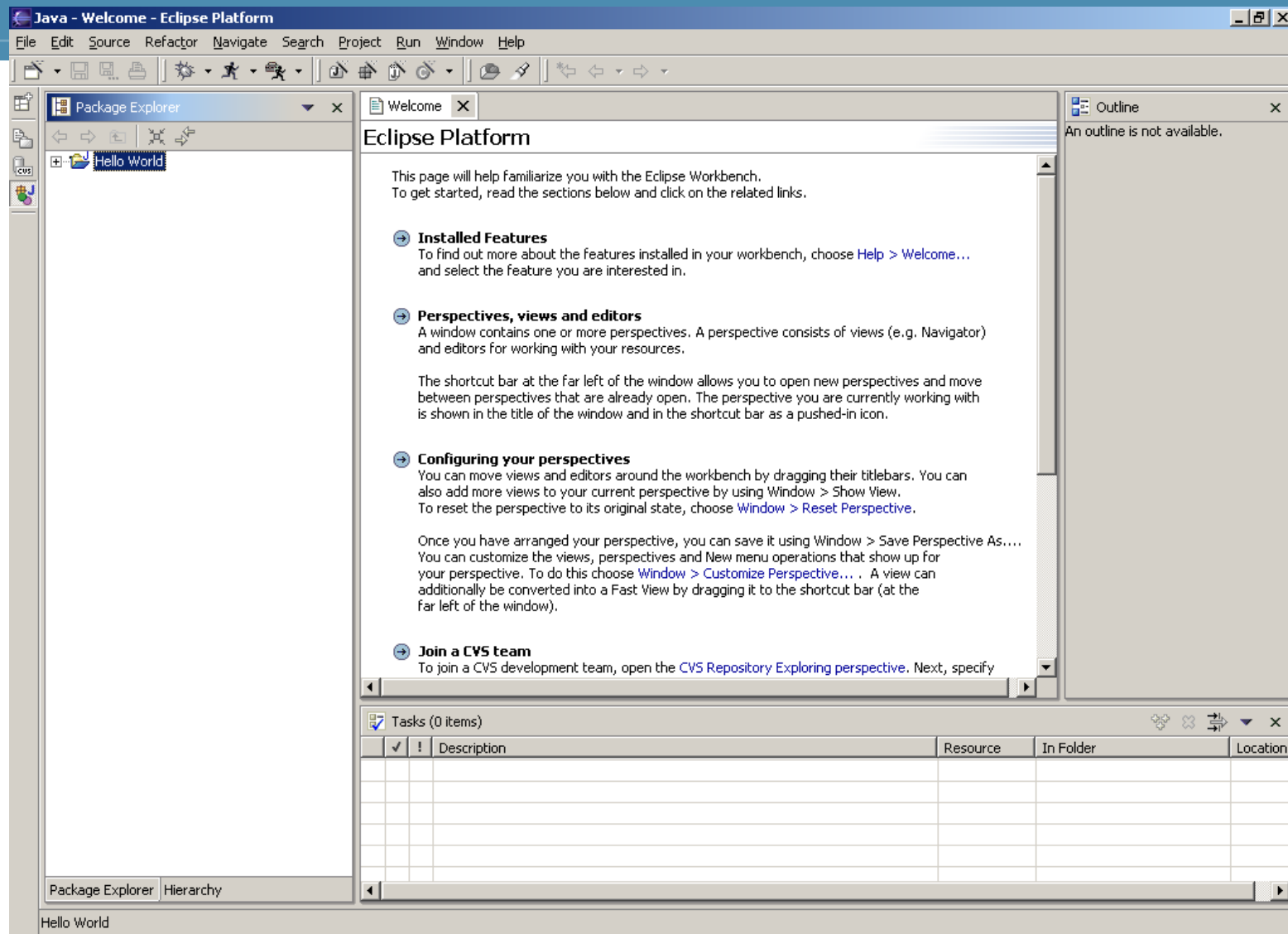
- Nesta janela são definidos o nome e a localização do projeto (um projeto não precisa estar no diretório do workspace!)



Conceitos Importantes

- O workspace na verdade é uma abstração: corresponde ao conjunto de projetos gerenciados pelo Eclipse.

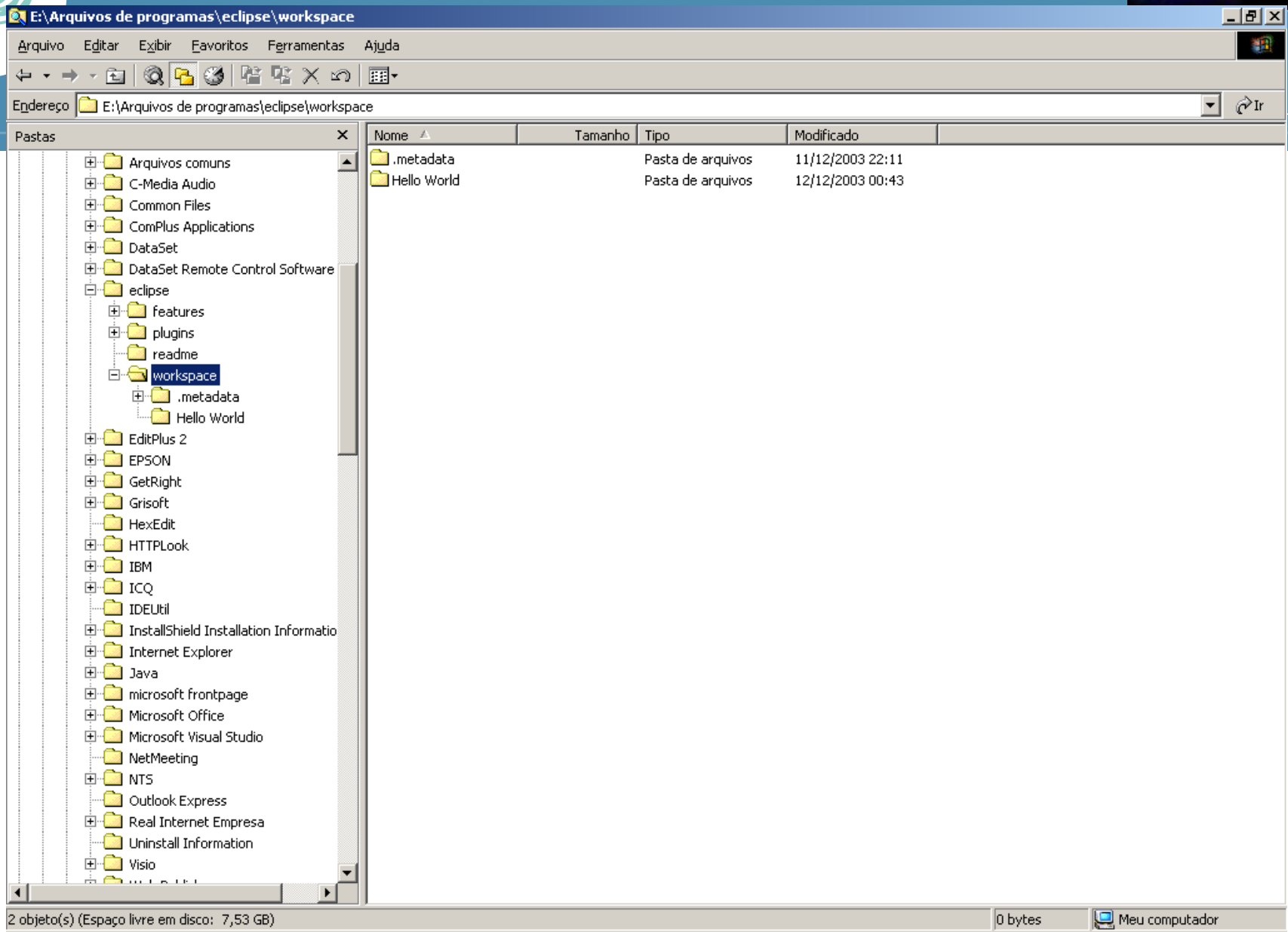
Conceitos Importantes





Conceitos Importantes

- Surgiu uma “nova pasta”. Dentro, ficam os recursos do projeto.
- Olhando o sistema de arquivos...





Conceitos Importantes

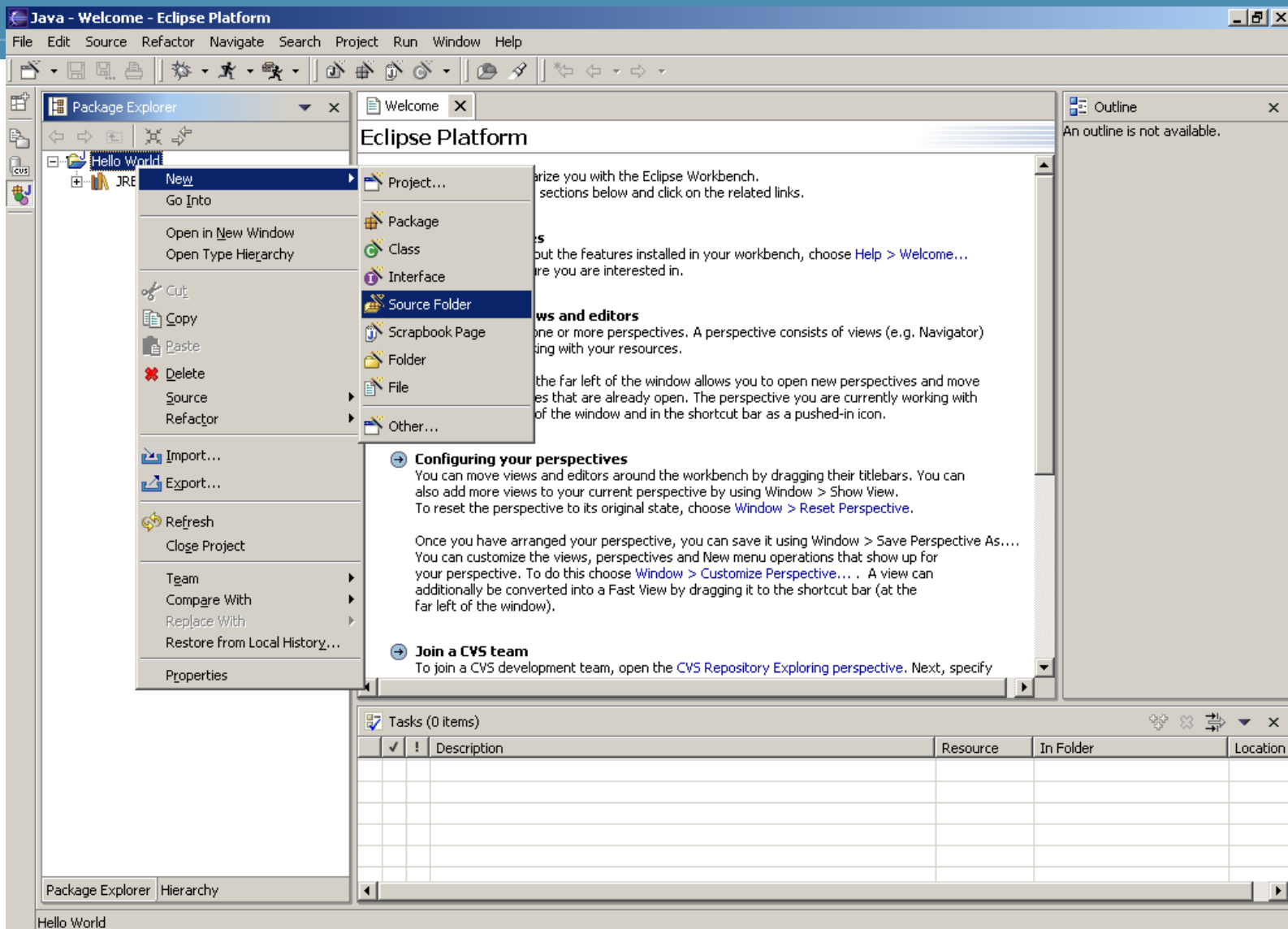
- Apareceu a pasta “Hello World”
 - O sistema de armazenamento do Eclipse é totalmente baseado em sistema de arquivos.
 - A maior parte das configurações do workspace é composta por arquivos XML.



Conceitos Importantes

- Como regra geral, um projeto pode ter arquivos e pastas
- Um projeto Java também pode ter arquivos e pastas, porém uma pasta pode ser de duas naturezas:
 - Folder: contém recursos quaisquer.
 - Source folder: contém código Java.

Hello World





Conceitos Importantes

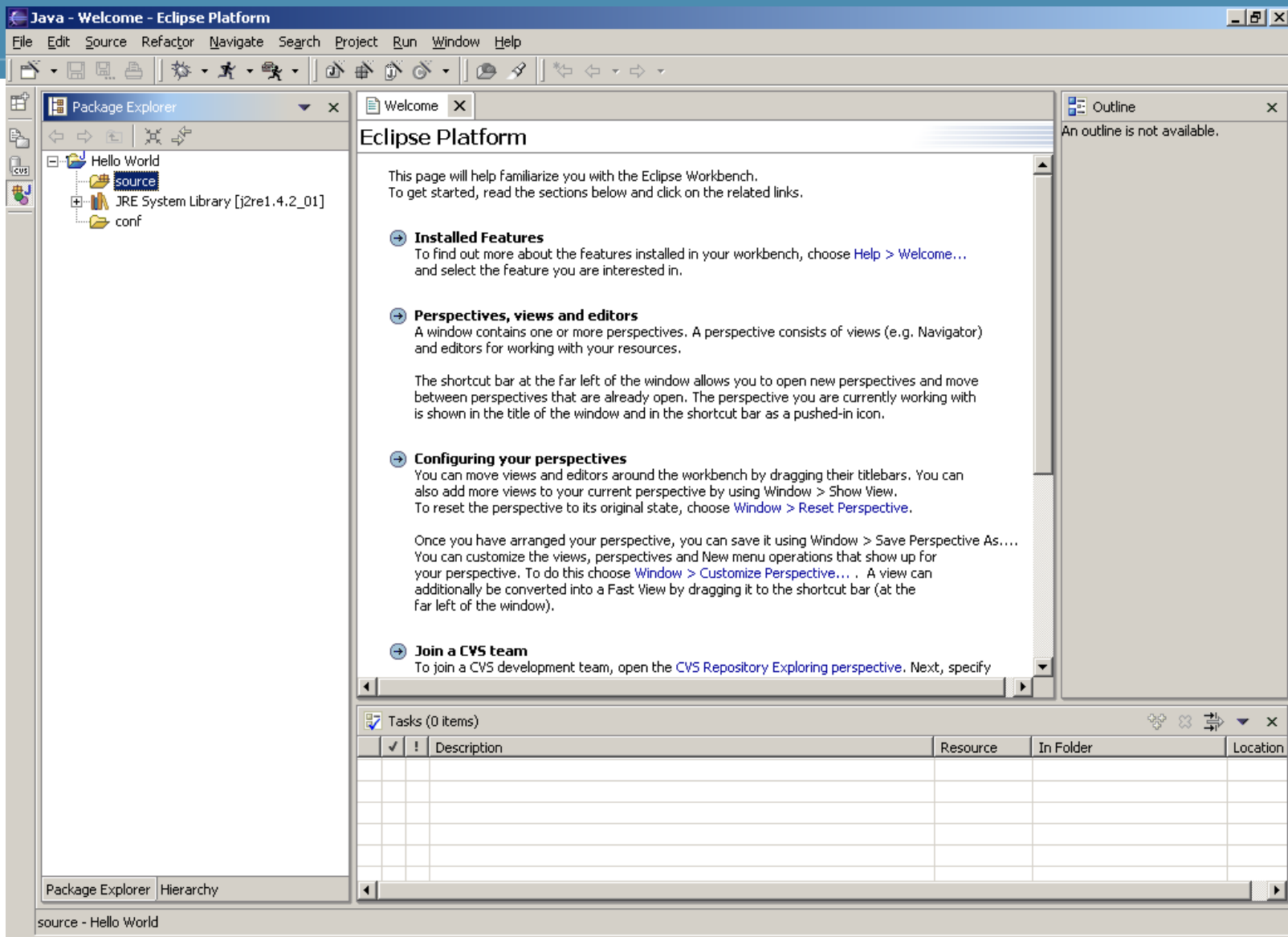
- A diferença é que:
 - O compilador não tenta compilar o que estiver presente numa pasta comum. O seu conteúdo é tratado como pastas e arquivos comuns
 - A source folder é compilada. Seu conteúdo é tratado como uma estrutura de pacotes. Arquivos .java são compilados



Conceitos Importantes

- Vamos criar duas pastas:
 - Uma source folder
 - Uma pasta comum

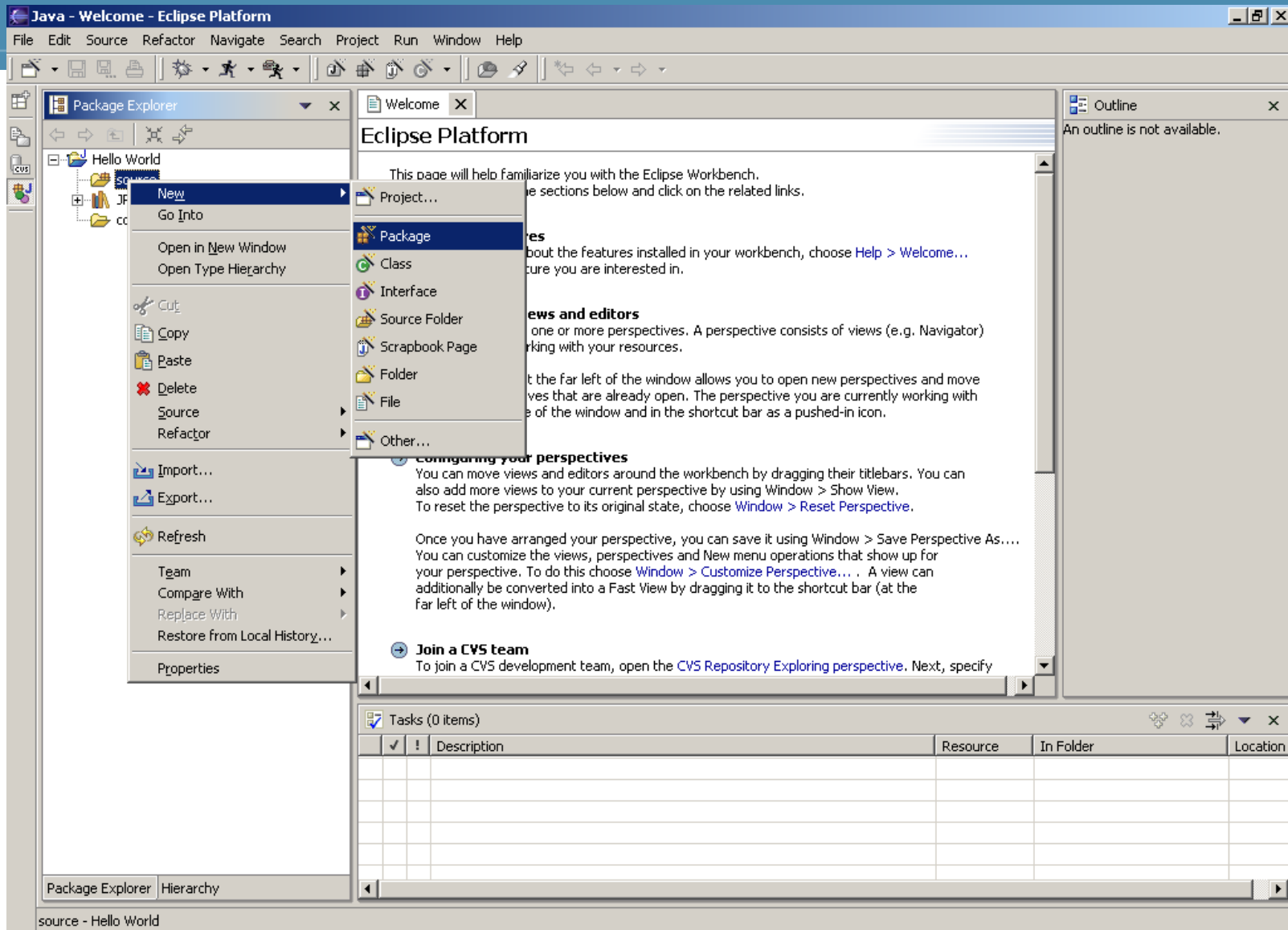
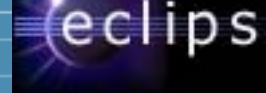
Conceitos Importantes





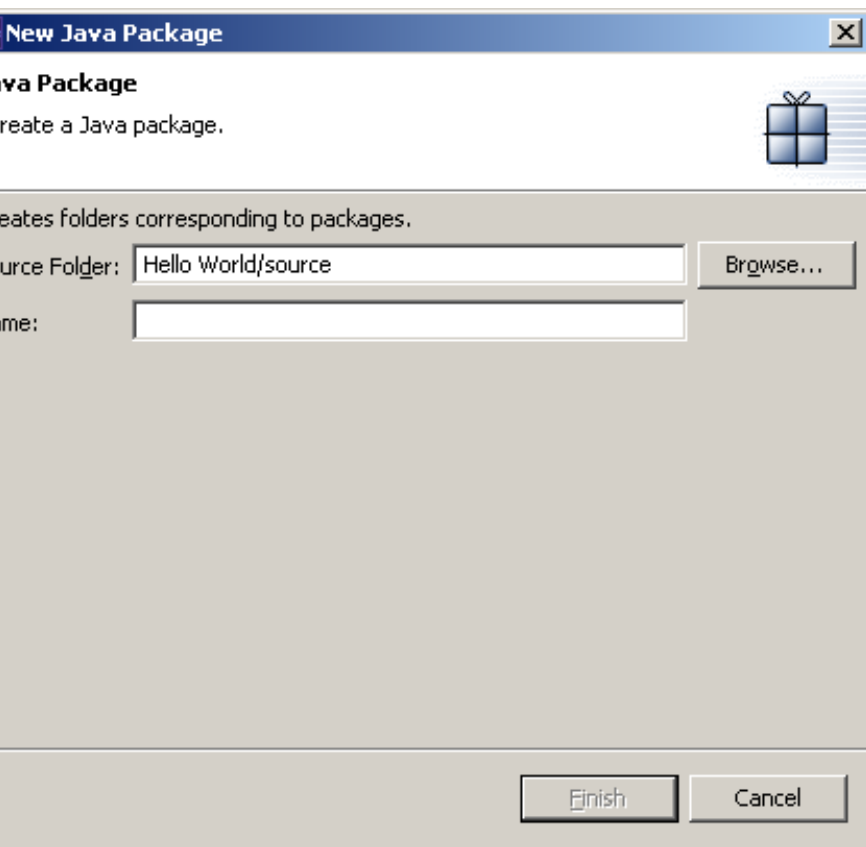
Conceitos Importantes

- Dentro da source folder colocaremos os pacotes e as classes.
- Vamos criar um pacote





Conceitos Importantes



- Basta digitar o nome do pacote



Conceitos Importantes

The screenshot displays the Eclipse IDE interface. The title bar reads "Java - Welcome - Eclipse Platform". The menu bar includes "File", "Edit", "Navigate", "Search", "Project", "Run", "Window", and "Help". The toolbar contains various icons for file operations and development tasks. On the left, the "Package Explorer" shows a project named "Hello World" with sub-elements "source", "br.edu.curso.eclipse", "JRE System Library [j2re1.4.2_01]", and "conf". The main editor area is titled "Eclipse Platform" and contains the following text:

This page will help familiarize you with the Eclipse Workbench. To get started, read the sections below and click on the related links.

- Installed Features**
To find out more about the features installed in your workbench, choose [Help > Welcome...](#) and select the feature you are interested in.
- Perspectives, views and editors**
A window contains one or more perspectives. A perspective consists of views (e.g. Navigator) and editors for working with your resources.

The shortcut bar at the far left of the window allows you to open new perspectives and move between perspectives that are already open. The perspective you are currently working with is shown in the title of the window and in the shortcut bar as a pushed-in icon.
- Configuring your perspectives**
You can move views and editors around the workbench by dragging their titlebars. You can also add more views to your current perspective by using [Window > Show View...](#). To reset the perspective to its original state, choose [Window > Reset Perspective...](#)

Once you have arranged your perspective, you can save it using [Window > Save Perspective As...](#). You can customize the views, perspectives and New menu operations that show up for your perspective. To do this choose [Window > Customize Perspective...](#). A view can additionally be converted into a Fast View by dragging it to the shortcut bar (at the far left of the window).
- Join a CVS team**
To join a CVS development team, open the [CVS Repository Exploring perspective](#). Next, specify

On the right, the "Outline" view displays the message "An outline is not available." At the bottom, the "Tasks" view shows "0 items" and a table with columns for "Description", "Resource", "In Folder", and "Location".

	✓	!	Description	Resource	In Folder	Location

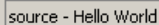
At the bottom left, the status bar shows "Package Explorer" and "Hierarchy".



Conceitos Importantes

- Note que o pacote apareceu em branco. Isso porque não há classes no pacote. Vamos criar uma classe

source - Hello World





Conceitos Importantes

New Java Class

Create a new Java class.

Source Folder:

Package:

☐ Enclosing type:

Name:

Modifiers: ☒ public ☐ default ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass:

Interfaces:

Which method stubs would you like to create?

☐ public static void main(String[] args)

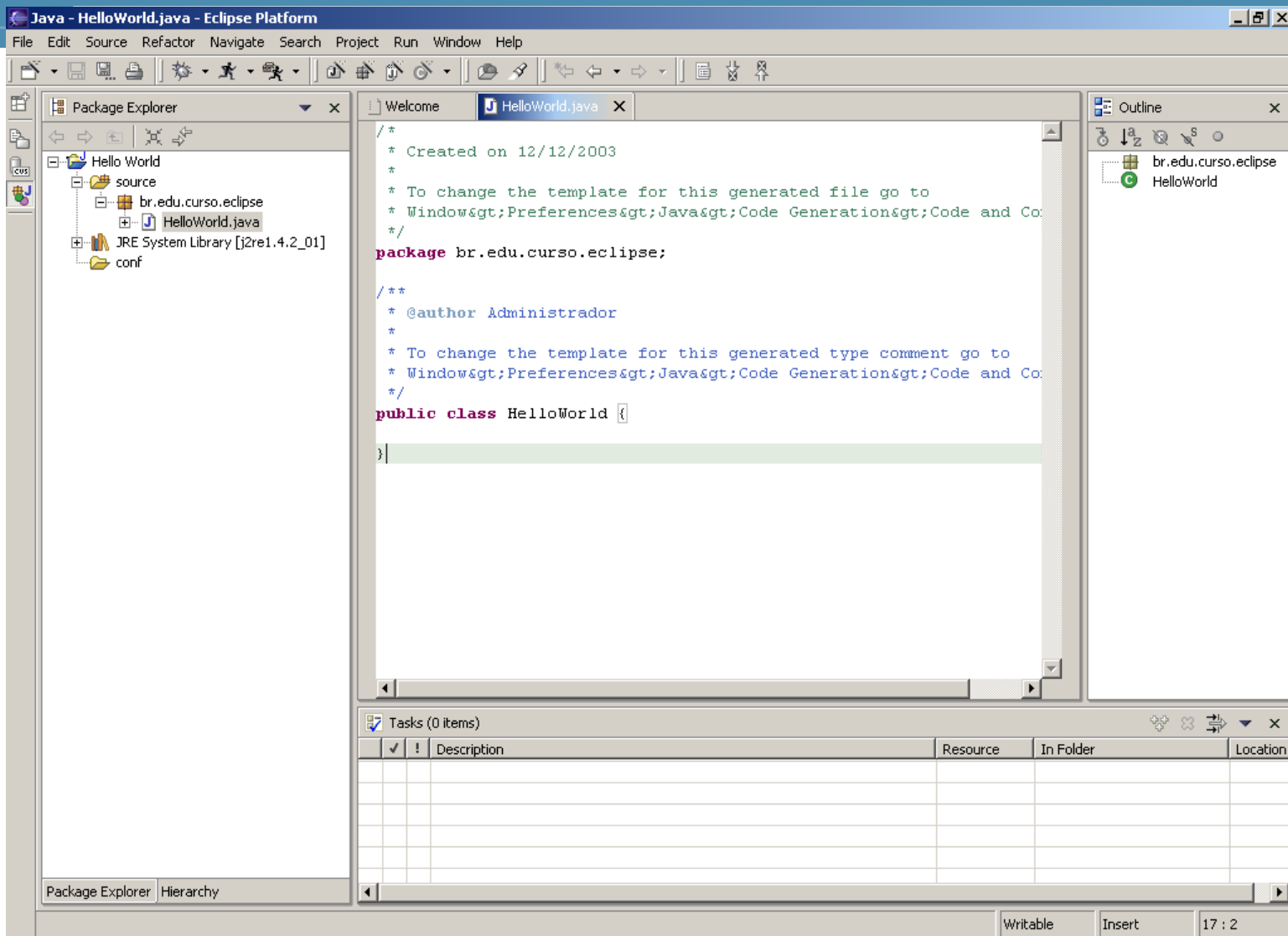
☐ Constructors from superclass

☒ Inherited abstract methods

- Existem Wizards para a criação de classes e interfaces. Essa é outra característica do Eclipse: a automatização da geração de código
- Note que várias características da classe podem ser definidas
 - Interfaces implementadas
 - Classe herdada



Conceitos Importantes





Conceitos Importantes

- Note que foi automaticamente criado um javadoc template e um comentário template.
 - Note que, no próprio comentário-padrão, há instruções sobre como e onde mudá-lo.
- Vamos olhar de novo o sistema de arquivos...



Conceitos Importantes

E:\Arquivos de programas\eclipse\workspace\Hello World\source\br\edu\curso\eclipse

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Endereço E:\Arquivos de programas\eclipse\workspace\Hello World\source\br\edu\curso\eclipse

Pastas	Nome	Tamanho	Tipo	Modificado
Arquivos de programas	HelloWorld.java	1 KB	Arquivo Java	12/12/2003 01:34

1 objeto(s) (Espaço livre em disco: 7,52 GB) 414 bytes Meu computador



Conceitos Importantes

- Note que as pastas que criamos apareceram lá, bem como o pacote com sua respectiva estrutura de classes.
- Apareceu também a pasta bin. Essa é a “pasta de compilação”, por default, o local onde os .class são inseridos.



Conceitos Importantes

- Note que os `.class` apareceram na pasta de compilação, sem que solicitássemos nenhuma operação de compilação.
- O Eclipse compila as nossas classes em tempo de criação! Dessa forma, erros podem ser detectados antes da compilação. Vamos inserir um método `main`

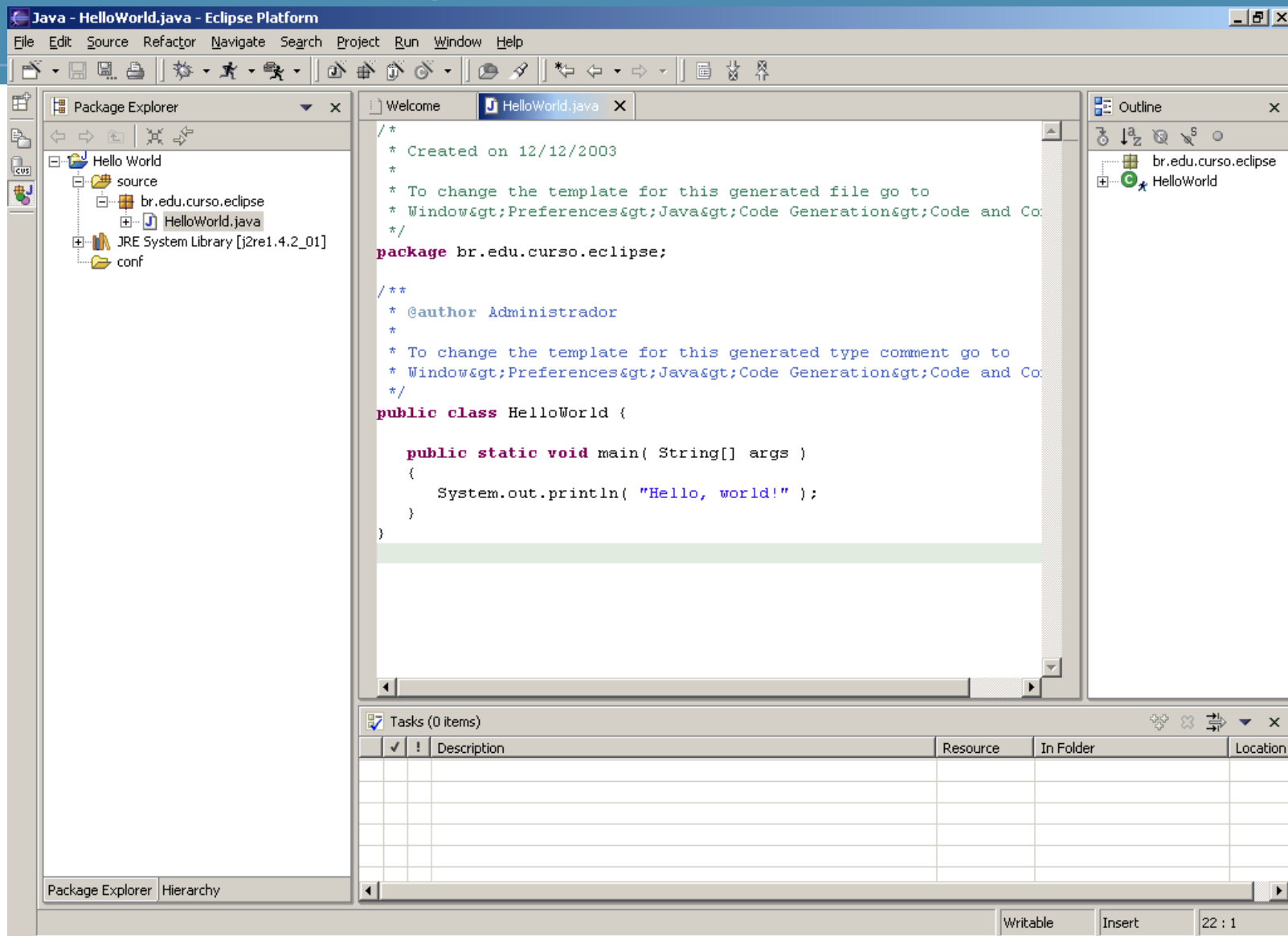


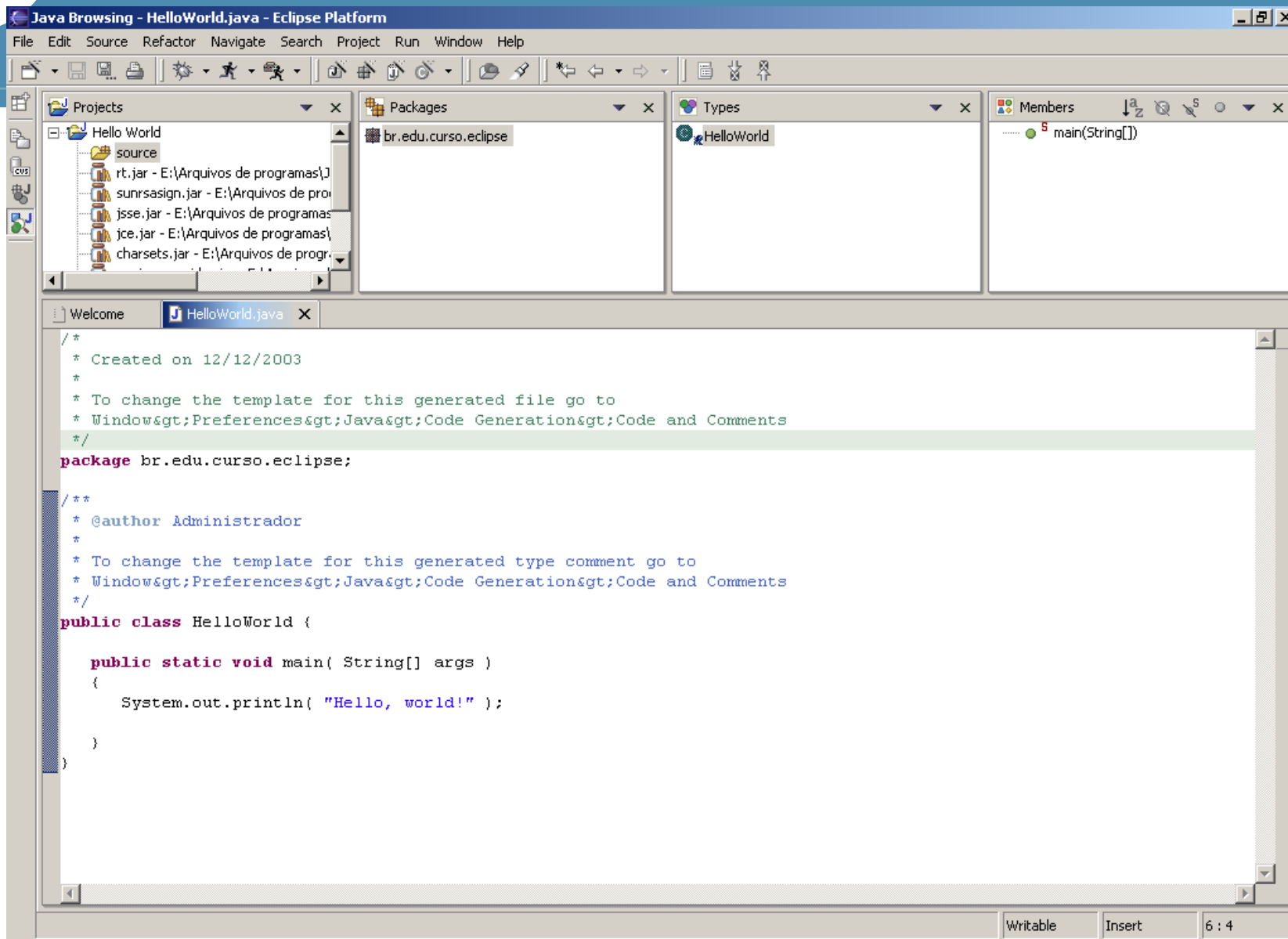
Conceitos Importantes

```
public static void main( String[] args )  
{  
    System.out.println( "Hello, world!" );  
}
```



Conceitos Importantes







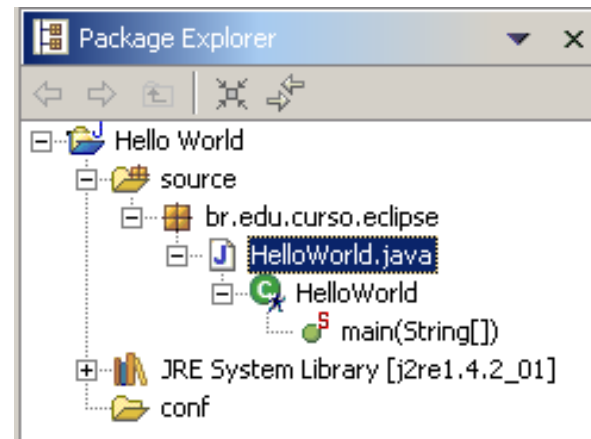
Conceitos Importantes

- Imediatamente, a classe é recompilada para a geração do .class



Conceitos Importantes

- Percebam que existe um sinal de + ao lado da classe. Perceba que os métodos apareceram na classe. O mesmo aconteceria para atributos. Um duplo-clique em cima deles leva diretamente ao código.





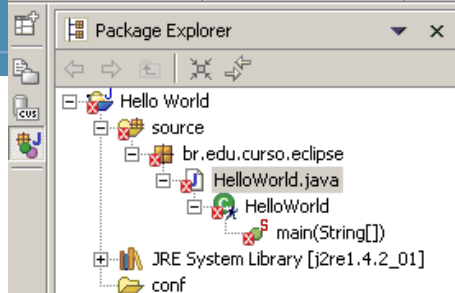
Conceitos Importantes

- E se inseríssemos um erro de compilação?
- O Eclipse exibiria, na hora, o erro ocorrido, bem como uma sugestão de como consertá-lo.



Java - HelloWorld.java - Eclipse Platform

File Edit Source Refactor Navigate Search Project Run Window Help

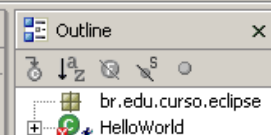


```
Welcome HelloWorld.java x
```

```
/*
 * Created on 12/12/2003
 *
 * To change the template for this generated file go to
 * Window>Preferences>Java>Code Generation>Code and Co
 */
package br.edu.curso.eclipse;

/**
 * @author Administrador
 *
 * To change the template for this generated type comment go to
 * Window>Preferences>Java>Code Generation>Code and Co
 */
public class HelloWorld {

    public static void main( String[] args )
    {
        Sytem.out.println( "Hello, world!" );
    }
}
```



Tasks (1 item)

	✓	!	Description	Resource	In Folder	Location
	✗		Sytem cannot be resolved	HelloWorld...	Hello World/source/br/edu/...	line 19

Package Explorer Hierarchy

Writable

Insert

19 : 9

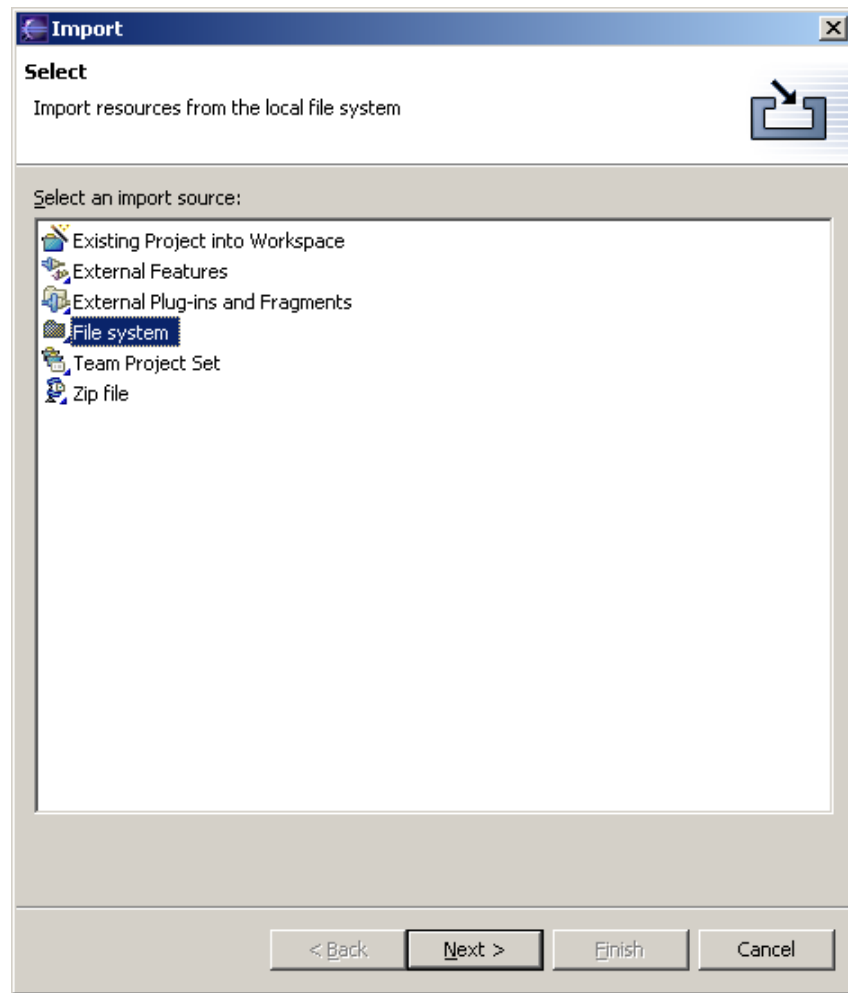
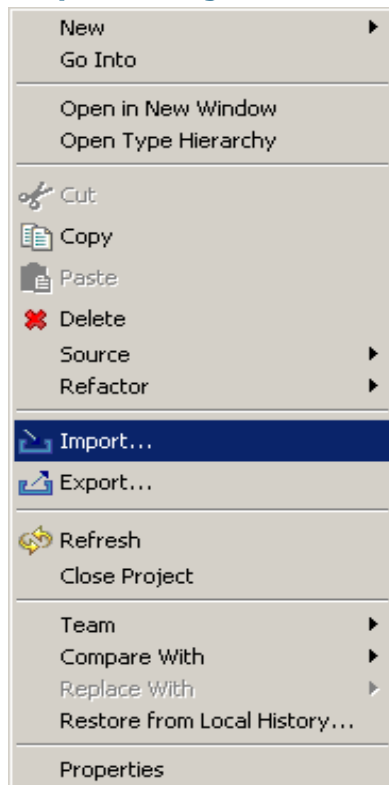


- Não há mais um passo de geração de código e um passo de compilação. Ambos estão condensados em um único passo.

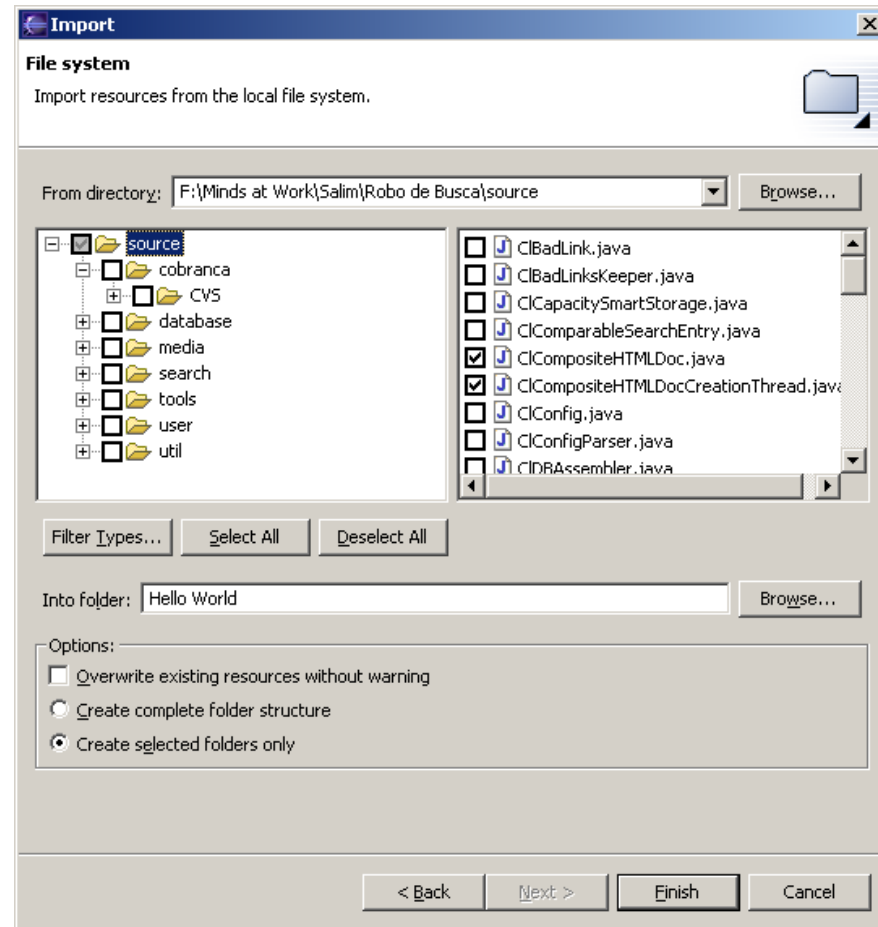


- E se você já tiver recursos feitos, e que deseja incorporar ao Eclipse?
 - Classes prontas
 - Pacotes definidos

- O Eclipse possui uma ferramenta de importação





- Basta selecionar o diretório raiz de importação em "From Directory"
- Na janela à esquerda aparecem os diretórios, tratados como pacotes
- Na janela à direita, os arquivos
- **Arquivos importados são fisicamente copiados para o projeto**

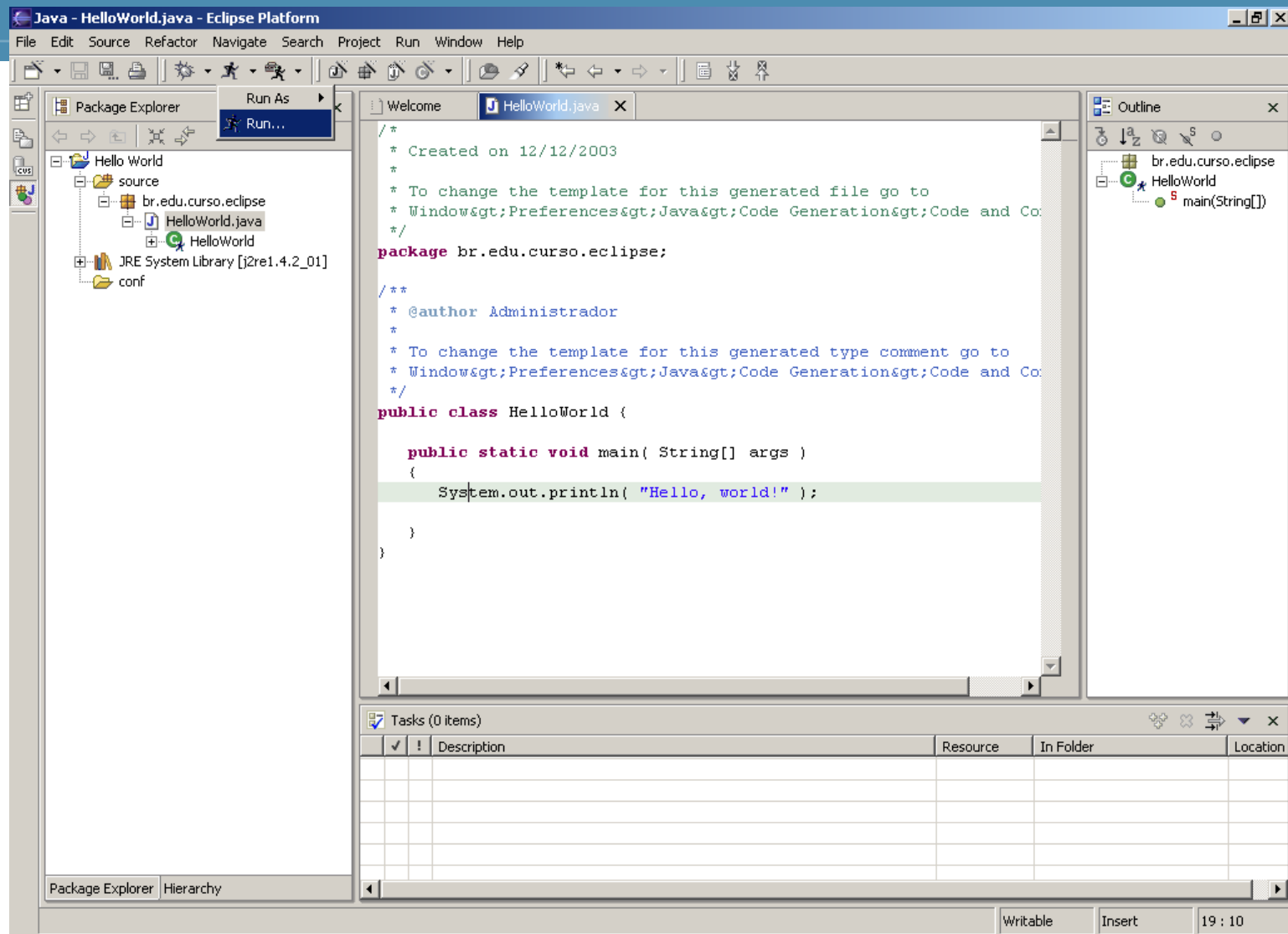




Conceitos Importantes

- Agora, a execução da aplicação. Há duas formas:
 - a execução simples: 
 - modo debug: 

Conceitos Importantes

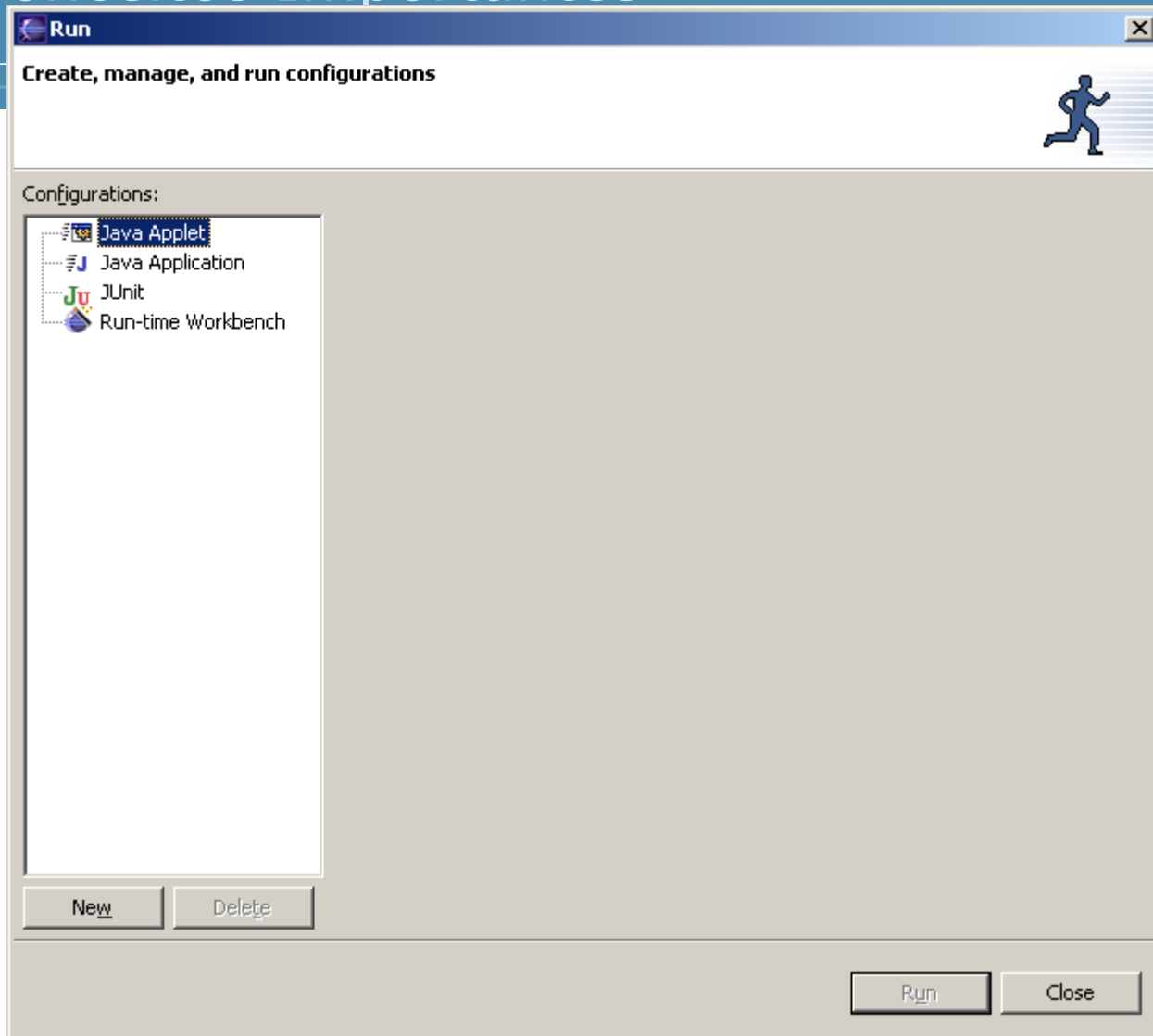


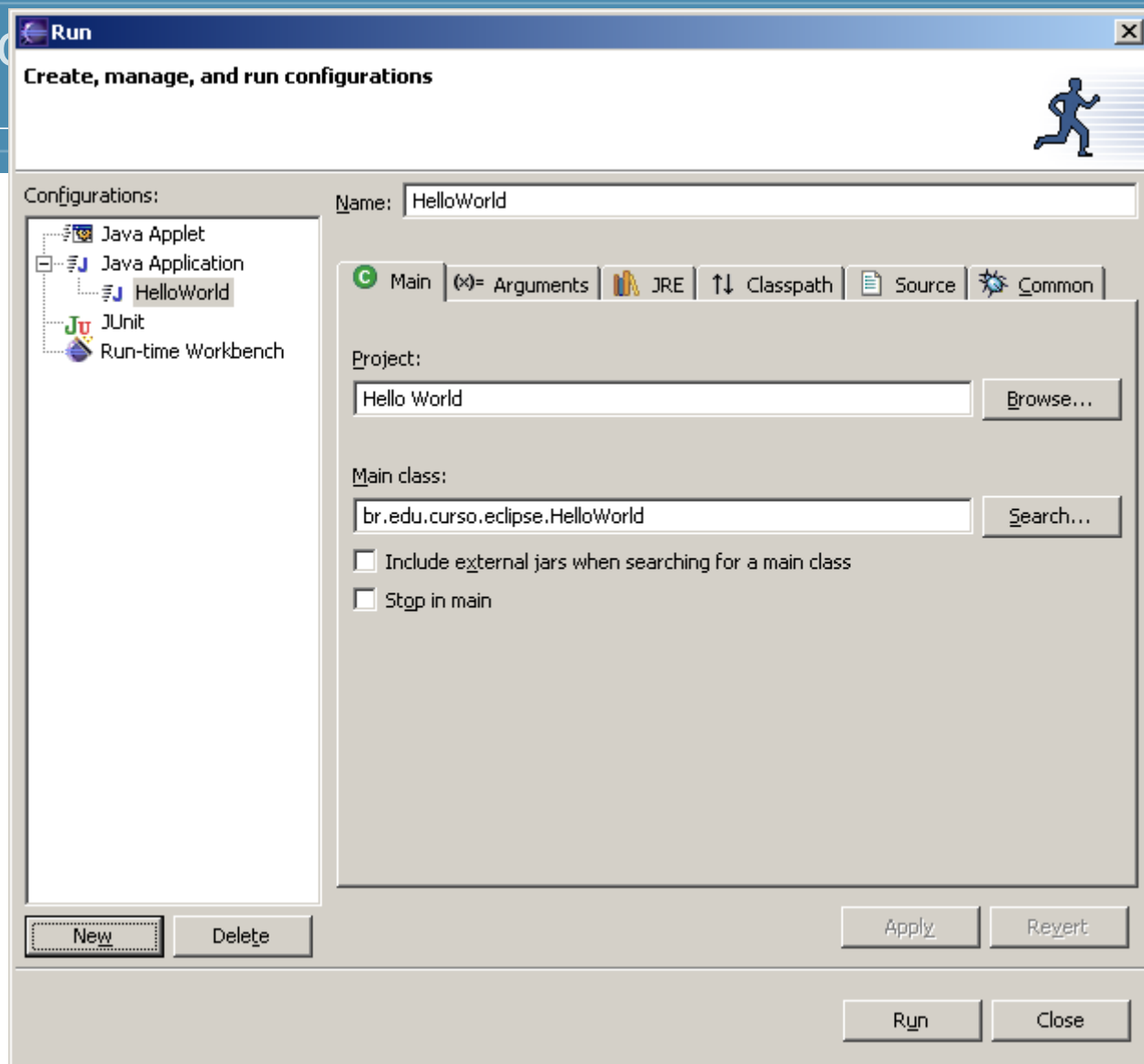


Conceitos Importantes

- É possível selecionar a natureza da aplicação.
- Vamos selecionar Java Application

Conceitos Importantes







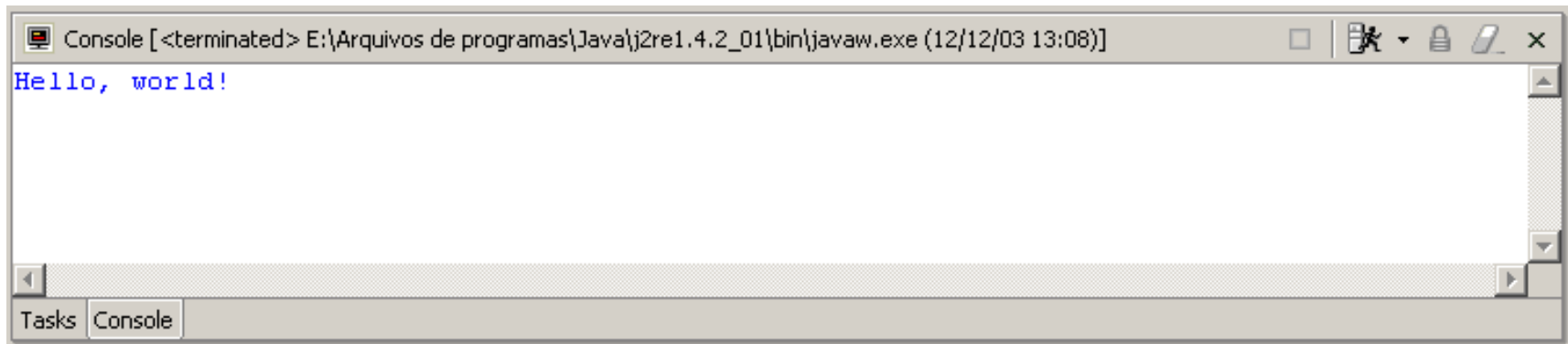
Conceitos Importantes

- É possível definir diversos parâmetros de execução, como:
 - JRE utilizado;
 - Parâmetros de execução;
 - Local dos arquivos fonte (usado para debug);
 - Classpath específico.

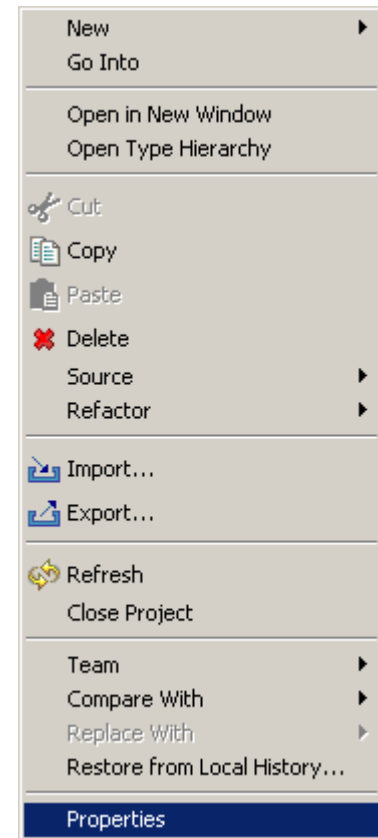


Conceitos Importantes

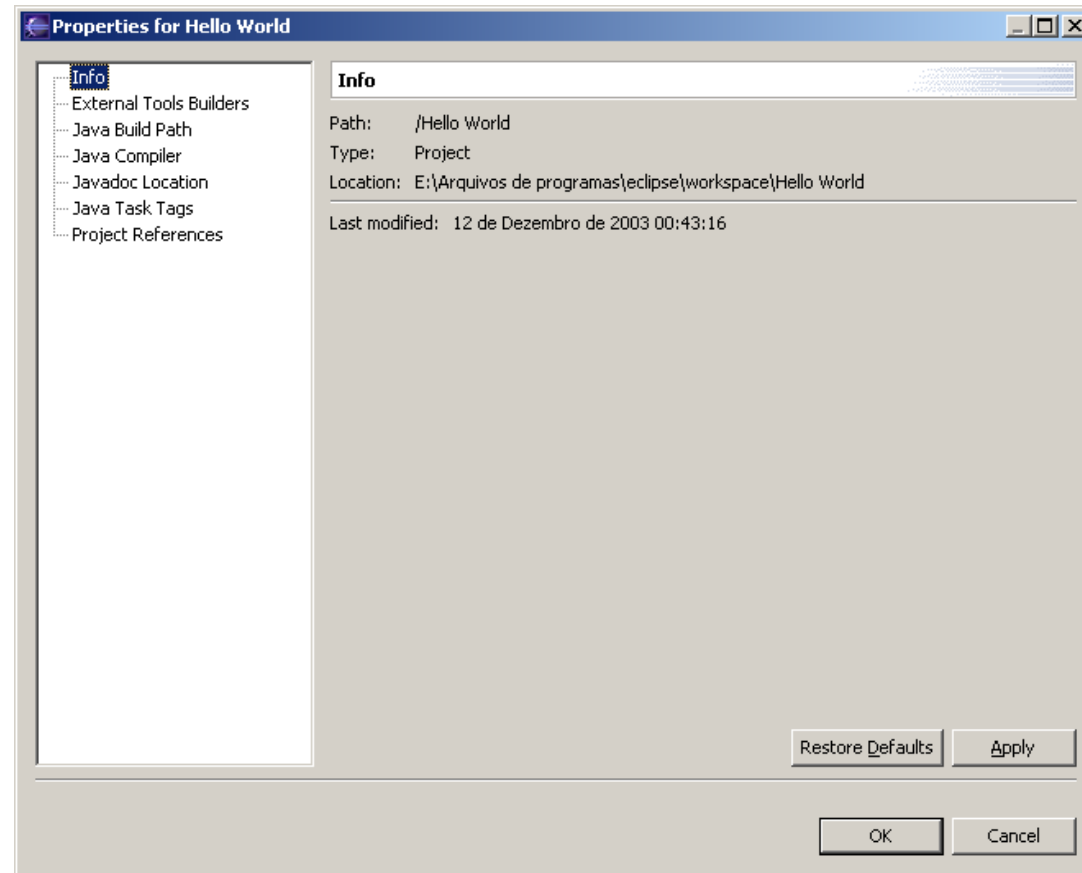
- Uma nova visão, a de console, exibe os resultados da execução



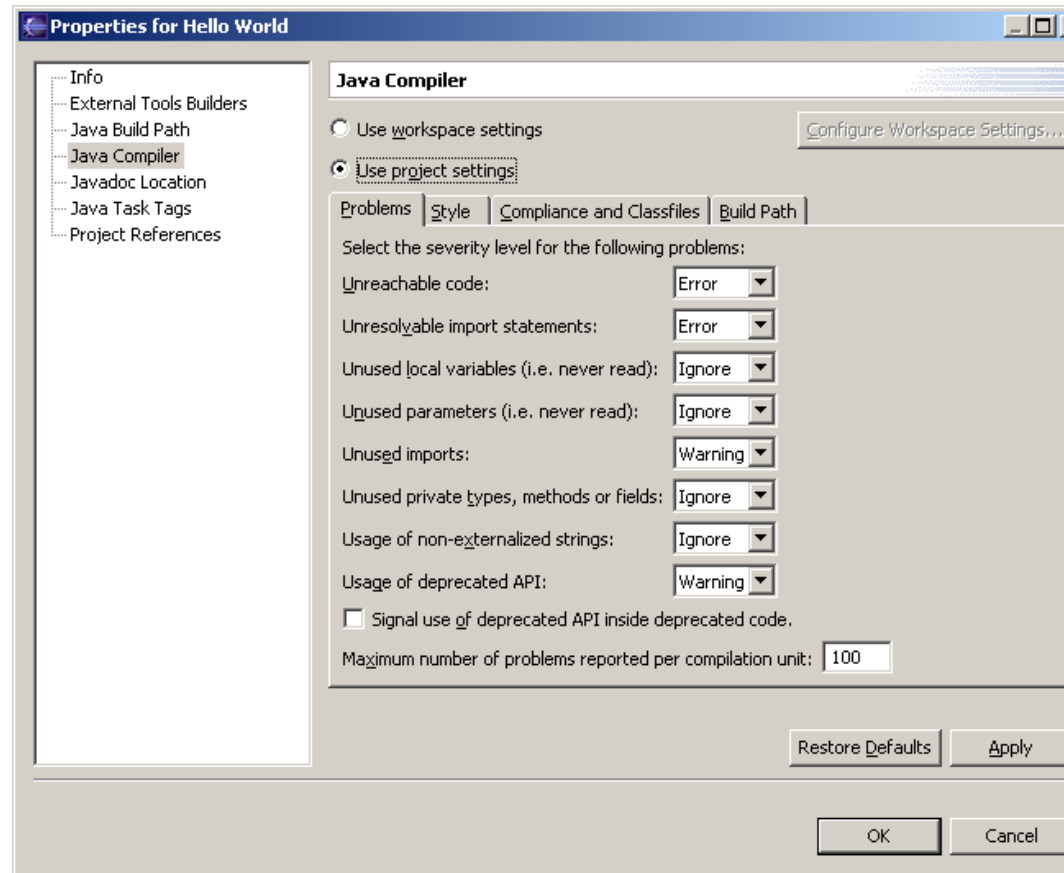
- O local da “pasta de compilação” é configurável nas propriedades do projeto.
- Botão direito do mouse em cima do projeto abre o menu aí ao lado
- Diferentes tipos de projeto possuem diferentes opções de configuração



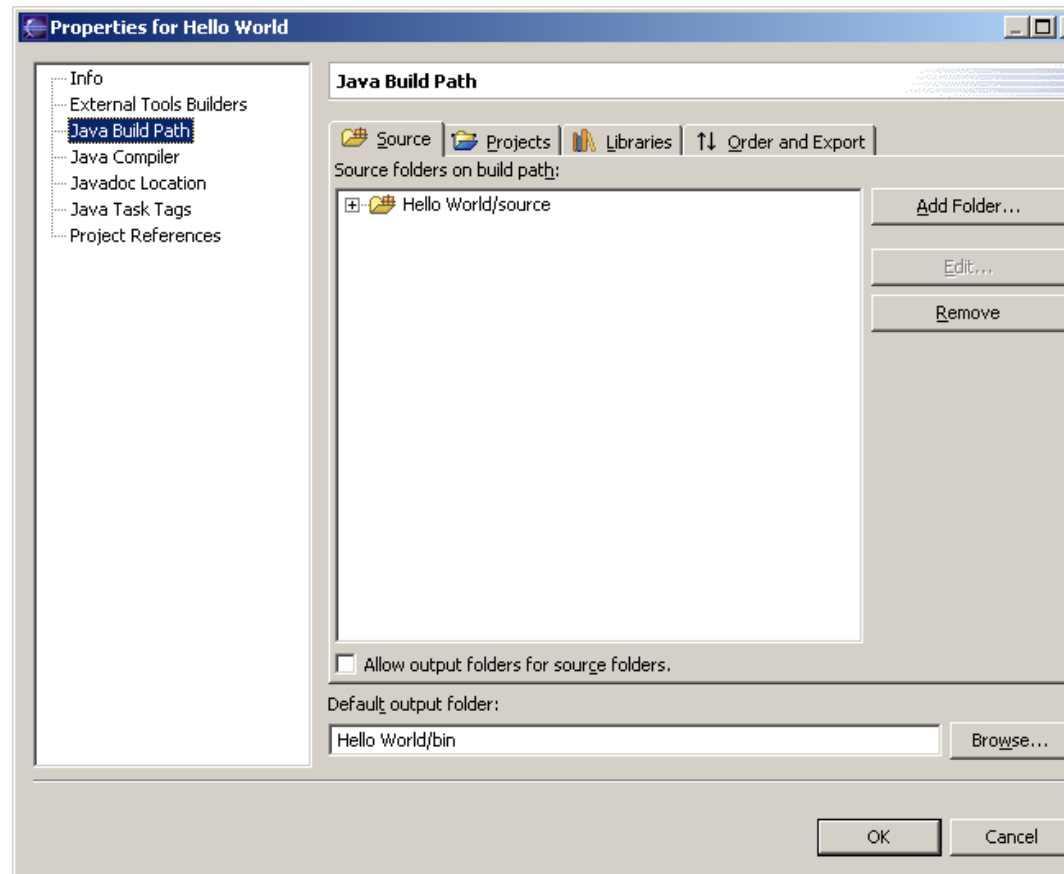
- Através da janela de propriedades do projeto, é possível configurar:
 - Caminho de construção (Classpath);
 - Ferramentas externas;
 - Referências entre projetos;
 - Compilador Java usado



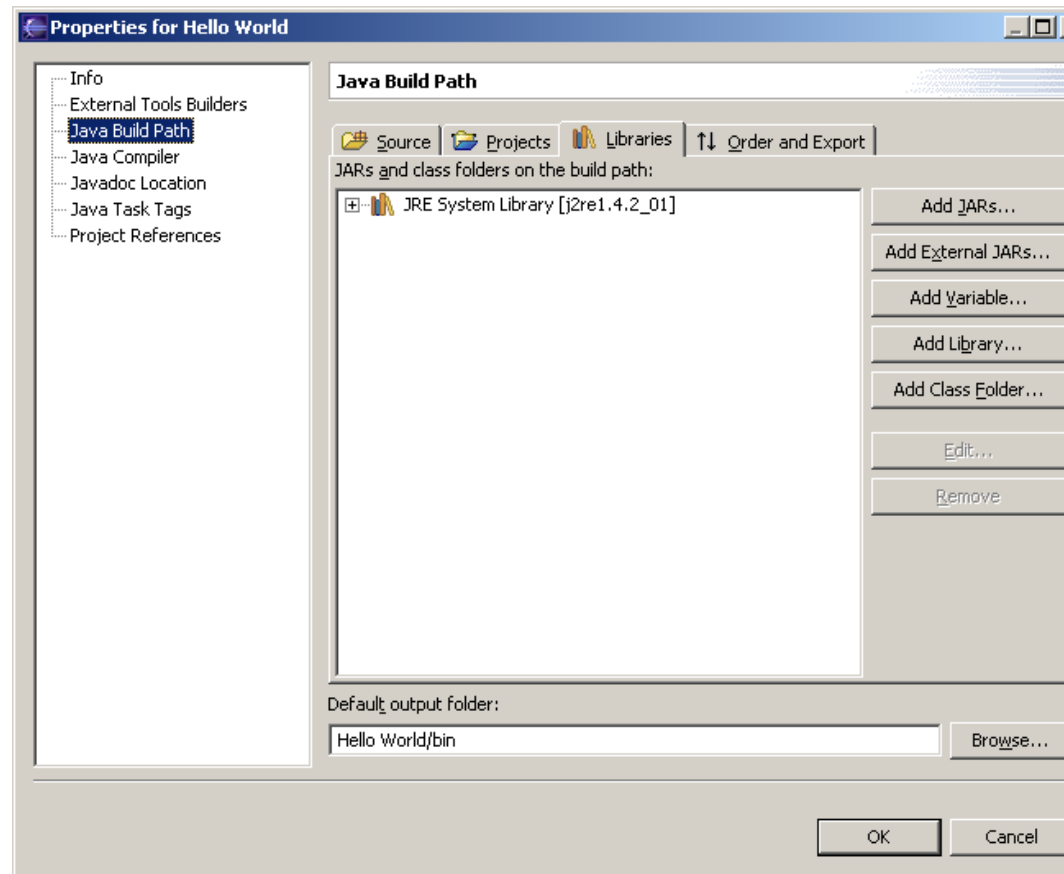
- O Compilador Java
 - Aba “Java Compiler” permite selecionar alguns critérios de compilação
 - Níveis de erros e warnings
 - Compatibilidade de código gerado
 - Esses critérios irão sobrescrever as configurações gerais (veremos isso mais à frente)



- O Caminho de Construção
 - Aba “Source” permite selecionar source folders e a “pasta de compilação”



- O Caminho de Construção
 - Aba “Libraries” permite selecionar as bibliotecas importadas pelo projeto
 - Add JARs e Add variable são os mais importantes



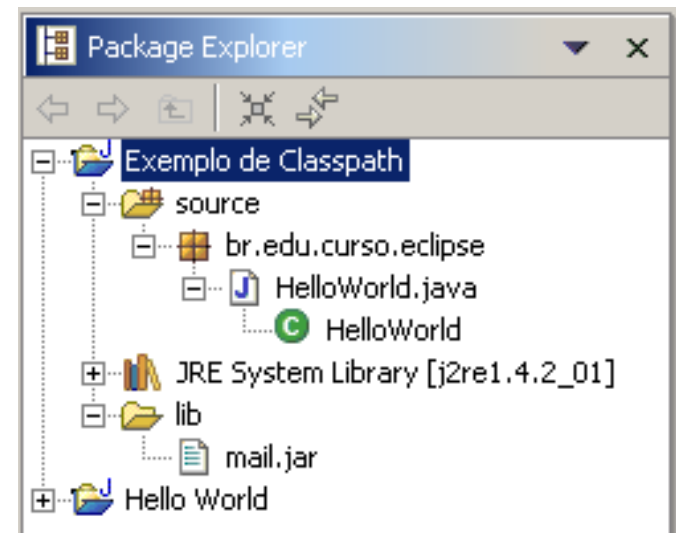


- Por exemplo, imagine que nosso projeto fizesse uso da api JavaMail da Sun.
- É através do caminho de construção que isso é feito.

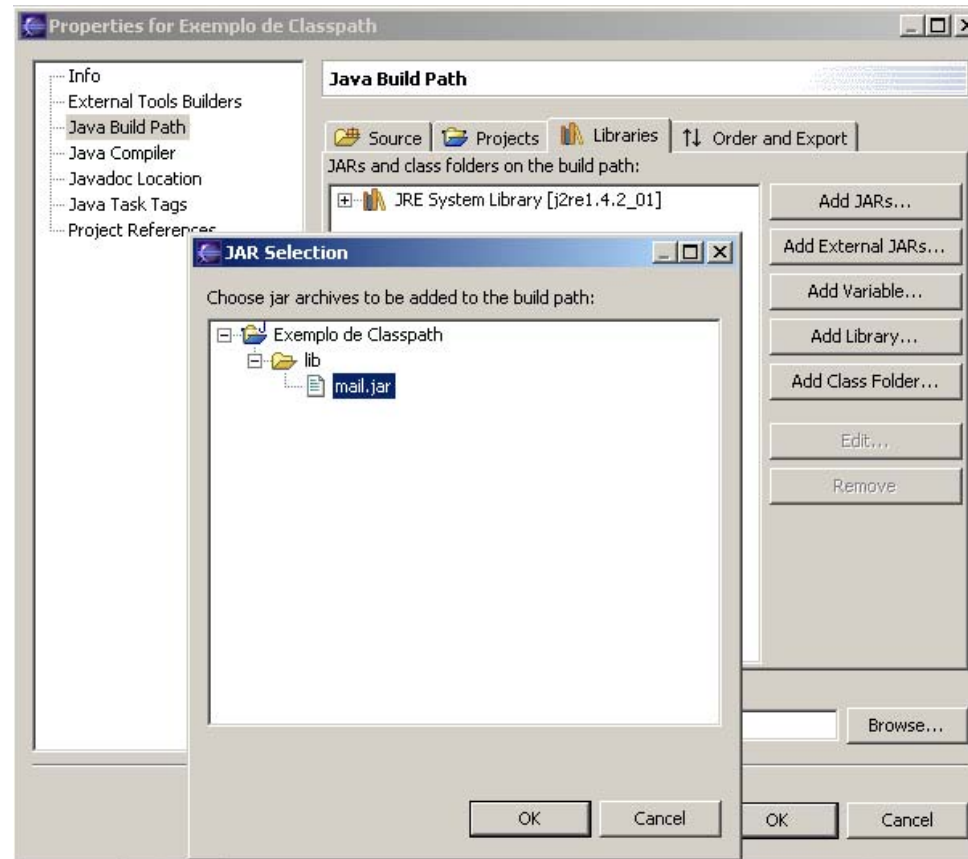


- Primeiro, criaremos um novo projeto, com uma nova classe HelloWorld
- Vamos apenas criar uma pasta, de nome lib. Nessa pasta colocaremos o arquivo mail.jar, que contém as classes do JavaMail.

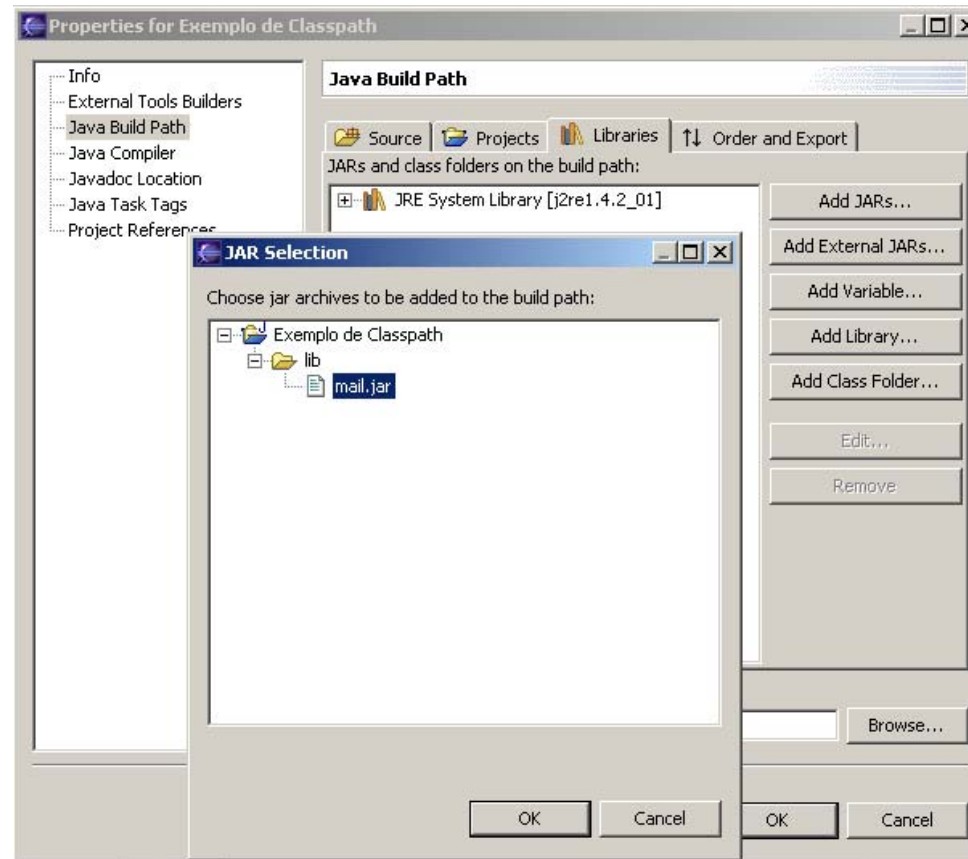
- Agora, é preciso incluir o mail.jar no classpath desse projeto. Cada projeto tem o seu classpath.



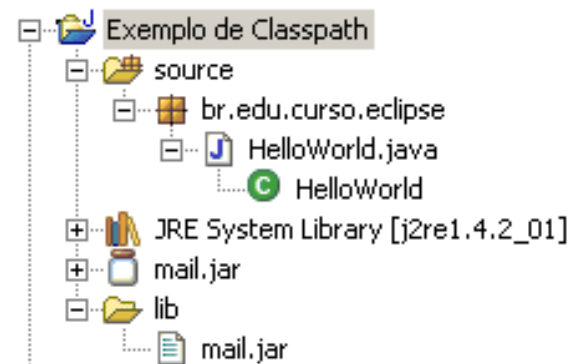
- Basta selecionar “Properties” dando clique com botão direito do mouse no projeto, selecionar “Java Build Path” e, na aba “Libraries”, selecionar o diretório lib.



- Todos os diretórios que possuam candidatos a bibliotecas (zip, jar, etc) serão listados, **mesmo diretórios de outros projetos**

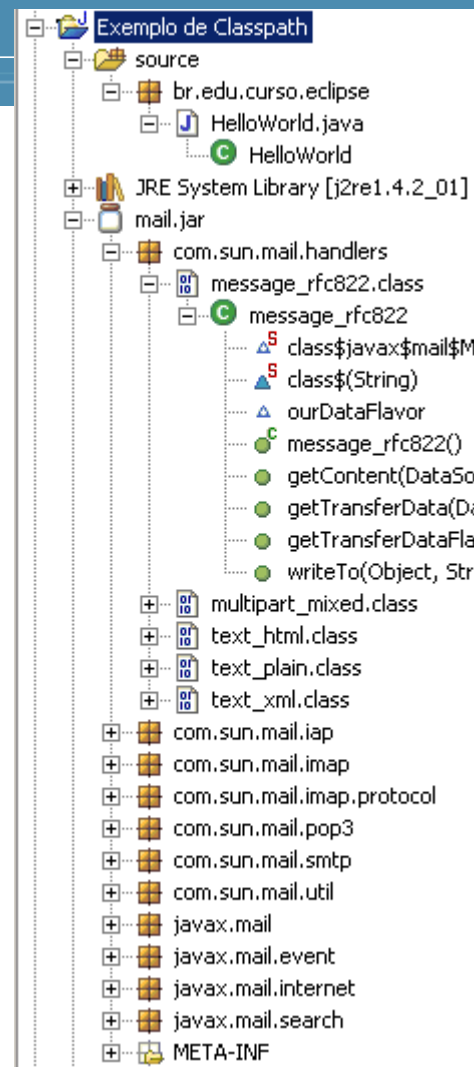


- Note que apareceu um “pote” (.jar) com o nome do .jar no projeto. Isso indica a dependência criada.



Conceitos Importantes

- Agora temos acesso a todas as classes do jar mail.jar





Conceitos Importantes

- E também podemos usá-las em nossas próprias classes. O esquema de code assistance do Eclipse passará a sugerir-las.
- Notaram o recurso de auto-completion do eclipse? Acostumem-se a usar o CTRL + espaço, é mais prático e menos propenso a erros!

```
package br.edu.curso.eclipse;
/*
 * Created on 13/12/2003
 *
 * To change the template for this generated file go to
 * Window>Preferences>Java>Code Generation>Code and Comments
 */
/**
 * @author Administrador
 *
 * To change the template for this generated type comment go to
 * Window>Preferences>Java>Code Generation>Code and Comments
 */
public class HelloWorld
{
    Mes
}
```

Message - com.sun.corba.se.internal.iiop.messages
Message - javax.mail
Message_1_0 - com.sun.corba.se.internal.iiop.messages
Message_1_1 - com.sun.corba.se.internal.iiop.messages
Message_1_2 - com.sun.corba.se.internal.iiop.messages
MessageAware - javax.mail
MessageBase - com.sun.corba.se.internal.iiop.messages
MessageCatalog - org.apache.crimson.util
MessageChangedEvent - javax.mail.event



Conceitos Importantes

- Existem outras formas para inserir bibliotecas.
 - O botão “Add JARs” adiciona bibliotecas do próprio workspace ao projeto.
 - O botão “Add External JARs” adiciona bibliotecas de todo o sistema de arquivos.
 - Não é recomendado devido ao desenvolvimento em equipe



Conceitos Importantes

- Existem outras formas para inserir bibliotecas.
 - O botão “Add Variable” veremos mais adiante...
 - O botão “Add Library” adiciona bibliotecas de sistema
 - O botão “Add Class Folder” é similar ao “Add JARs”, só que com pastas e classes.



Conceitos Importantes

- E se houvesse algum erro de execução, como a exceção seria exibida?

```
public static void main( String[] args )  
{  
    System.out.println( args[ 0 ] );  
}
```



Conceitos Importantes

Java - HelloWorld.java - Eclipse Platform

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- Hello World
 - source
 - br.edu.curso.eclipse
 - HelloWorld.java
 - HelloWorld
 - JRE System Library [j2re1.4.2_01]
 - conf

Welcome

HelloWorld.java

```
/*
 * Created on 12/12/2003
 *
 * To change the template for this generated file go to
 * Window>Preferences>Java>Code Generation>Code and Co
 */
package br.edu.curso.eclipse;

/**
 * @author Administrador
 *
 * To change the template for this generated type comment go to
 * Window>Preferences>Java>Code Generation>Code and Co
 */
public class HelloWorld {

    public
    {
        Syst
    }
}
```

Outline

- br.edu.curso.eclipse
 - HelloWorld
 - main(String[])

Java Virtual Machine Launcher

Fatal exception occurred. Program will exit.

OK

Console [E:\Arquivos de programas\Java\j2re1.4.2_01\bin\javaw.exe (12/12/03 13:31)]

```
java.lang.ArrayIndexOutOfBoundsException: 0
    at br.edu.curso.eclipse.HelloWorld.main(HelloWorld.java:19)
```


Package Explorer Hierarchy

Tasks Console

Writable Insert 17 : 2



Conceitos Importantes

- Isso lembra o processo de DEBUG...
- Existe uma perspectiva específica para isso – é a perspectiva de debug.
- Vamos debugar esse programa. Basta clicar sobre o ícone 



Conceitos Importantes

Debug - HelloWorld.java - Eclipse Platform

File Edit Source Refactor Navigate Search Project Run Window Help

Debug

- HelloWorld [Java Application]
 - br.edu.curso.eclipse.HelloWorld at localhost:13595
 - System Thread [Finalizer] (Running)
 - System Thread [Reference Handler] (Running)
 - Thread [main] (Suspended (exception ArrayIndexOutOfBoundsException))
 - HelloWorld.main(String[]) line: 19
 - System Thread [Signal Dispatcher] (Running)
 - E:\Arquivos de programas\Java\j2re1.4.2_01\bin\javaw.exe (12/12/03 13:56)

Variables

args= String[0] (id=16)

Variables Breakpoints Expressions Display

Welcome HelloWorld.java

```
*
 * To change the template for this generated type comment go to
 * Window>Preferences>Java>Code Generation>Code and Comments
 */
public class HelloWorld {

    public static void main( String[] args )
    {
        System.out.println( args[ 0 ] );
    }
}
```

Outline

- br.edu.curso.eclipse
 - HelloWorld
 - main(String[])

Console [br.edu.curso.eclipse.HelloWorld at localhost:13595]

Console Tasks

Writable Insert 20 : 5

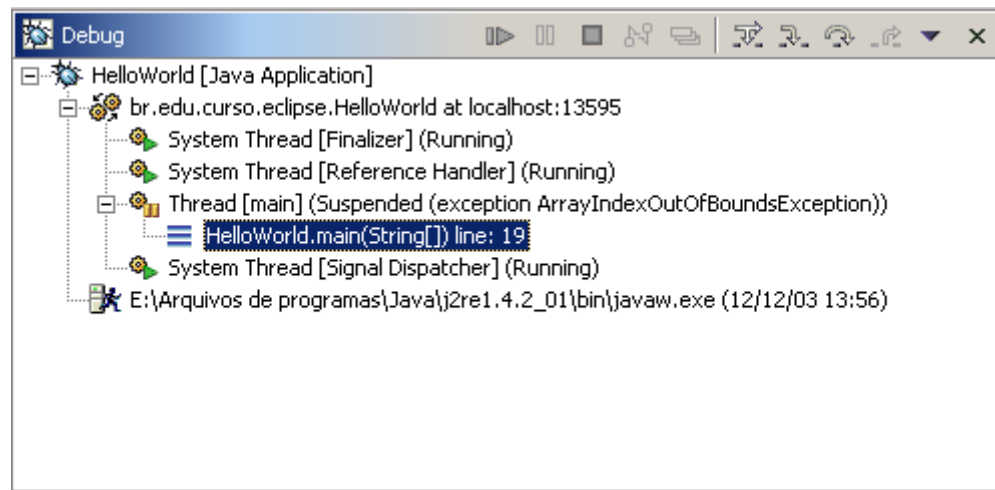


Conceitos Importantes

- Existem 5 áreas nessa perspectiva:
 - Visão de programas debugados;
 - Visão de variáveis, breakpoints, etc;
 - Editor de código;
 - Visão de outline;
 - Visão de console.

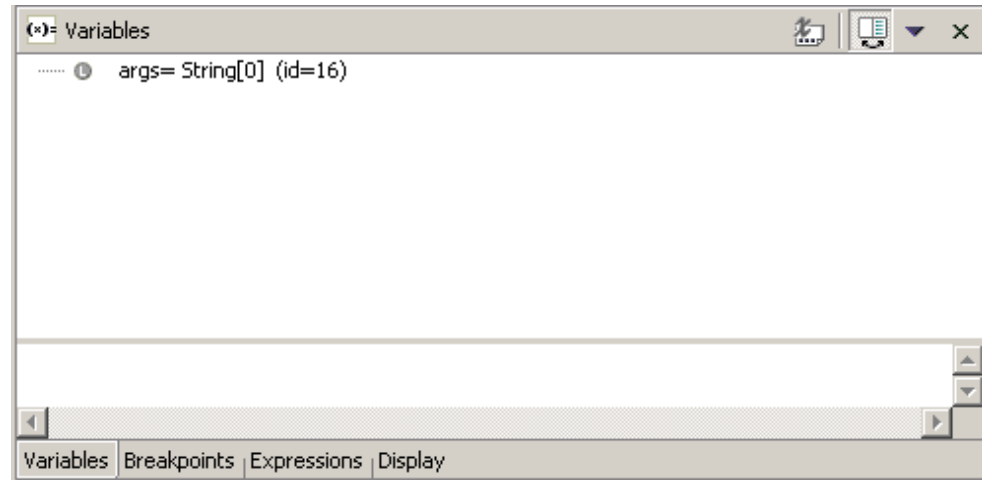
Visão de programas debugados

- Nesta visão, é possível ver os programas sendo debugados, bem como a pilha de ativação de cada um



Visão de variáveis e etc.

- Exibe variáveis locais, breakpoints, expressões e informações sobre a execução do programa





Editor de código

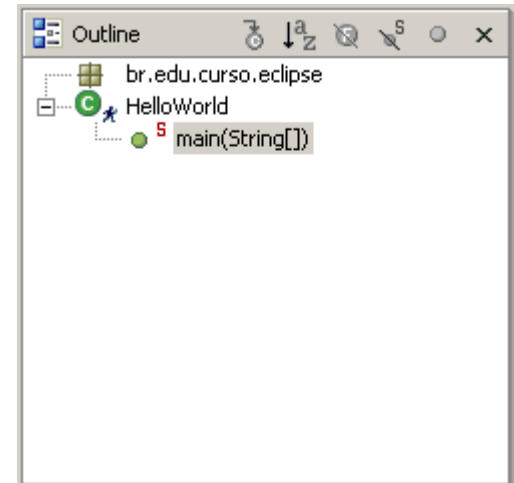
- Exibe o código da classe sendo visualizada no momento na visão de programas debugados

```
*  
* To change the template for this generated type comment go to  
* Window>Preferences>Java>Code Generation>Code and Comments  
*/  
public class HelloWorld {  
  
    public static void main( String[] args )  
    {  
        System.out.println( args[ 0 ] );  
    }  
}
```



Visão de outline

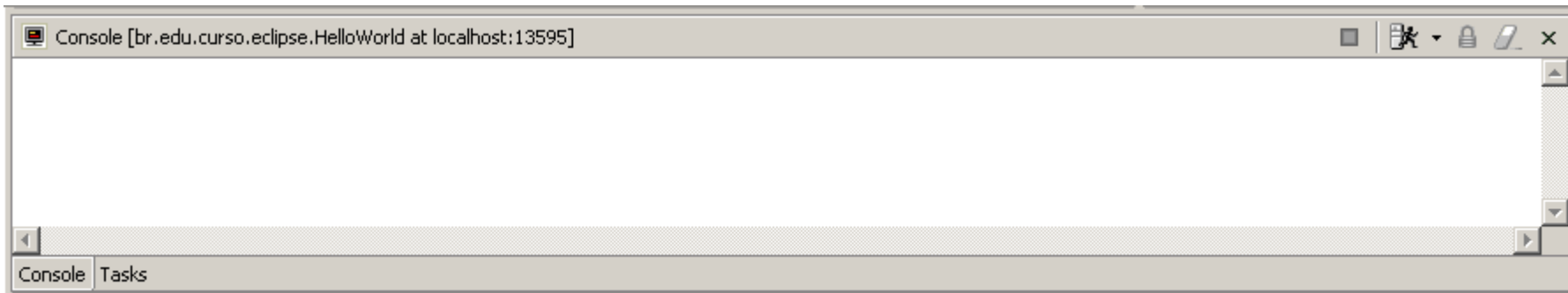
- Exibe informações sobre a classe sendo editada





Visão de console

- Recebe o stdout e o stderr da aplicação (stderr em vermelho, stdout em preto)



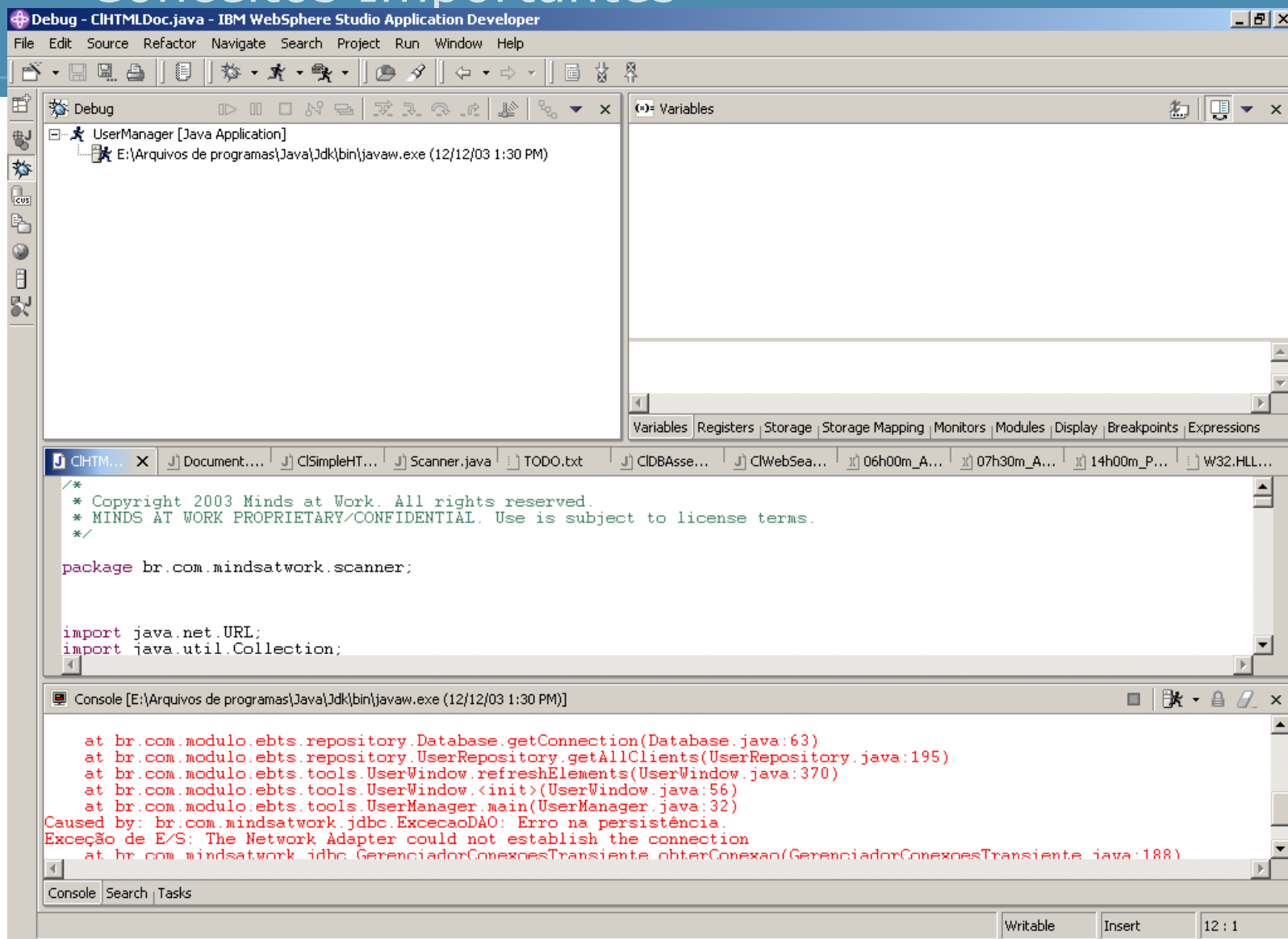


Conceitos Importantes

- Lembre-se que você pode customizar uma perspectiva! Assim, padrão é que haja essas 5 janelas abertas, mas nada impede que você altere (uma idéia é, por exemplo, fechar o outline para ficar com mais espaço no código)



Conceitos Importantes





Conceitos Importantes

- O processo de debug em Java...
 - Passo-a-passo na execução do programa
 - Edição e gerenciamento dos breakpoints na visão de breakpoints
 - Exame de variáveis na visão de variáveis
 - Verificação da saída do programa na visão de console



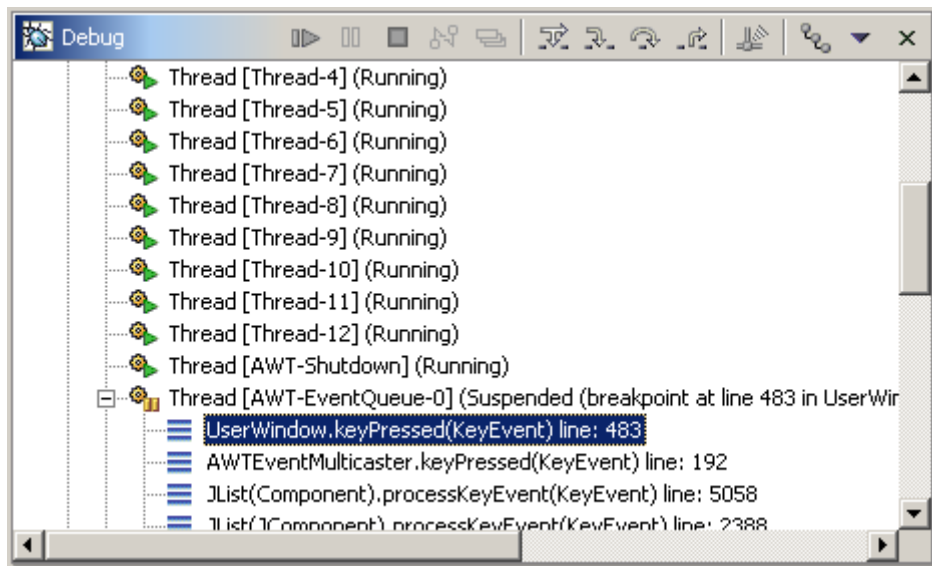
Conceitos Importantes

- Para definir um breakpoint:
 - Duplo-clique sobre a posição esquerda do editor de código Java.

```
CIHTMLDo... Docum... X CISimpleHT... Scanner.java TODO.txt CIDBAsse... CIWebSea... 06h00m_A... 07h30m_A...
* @param text
* @param documentDate
*
*/
protected Document( Link link, URL url, String text, Date documentDate )
{
    this.link      = link;
    this.url       = url;
    this.text      = text;
    this.documentDate = documentDate;
}
```

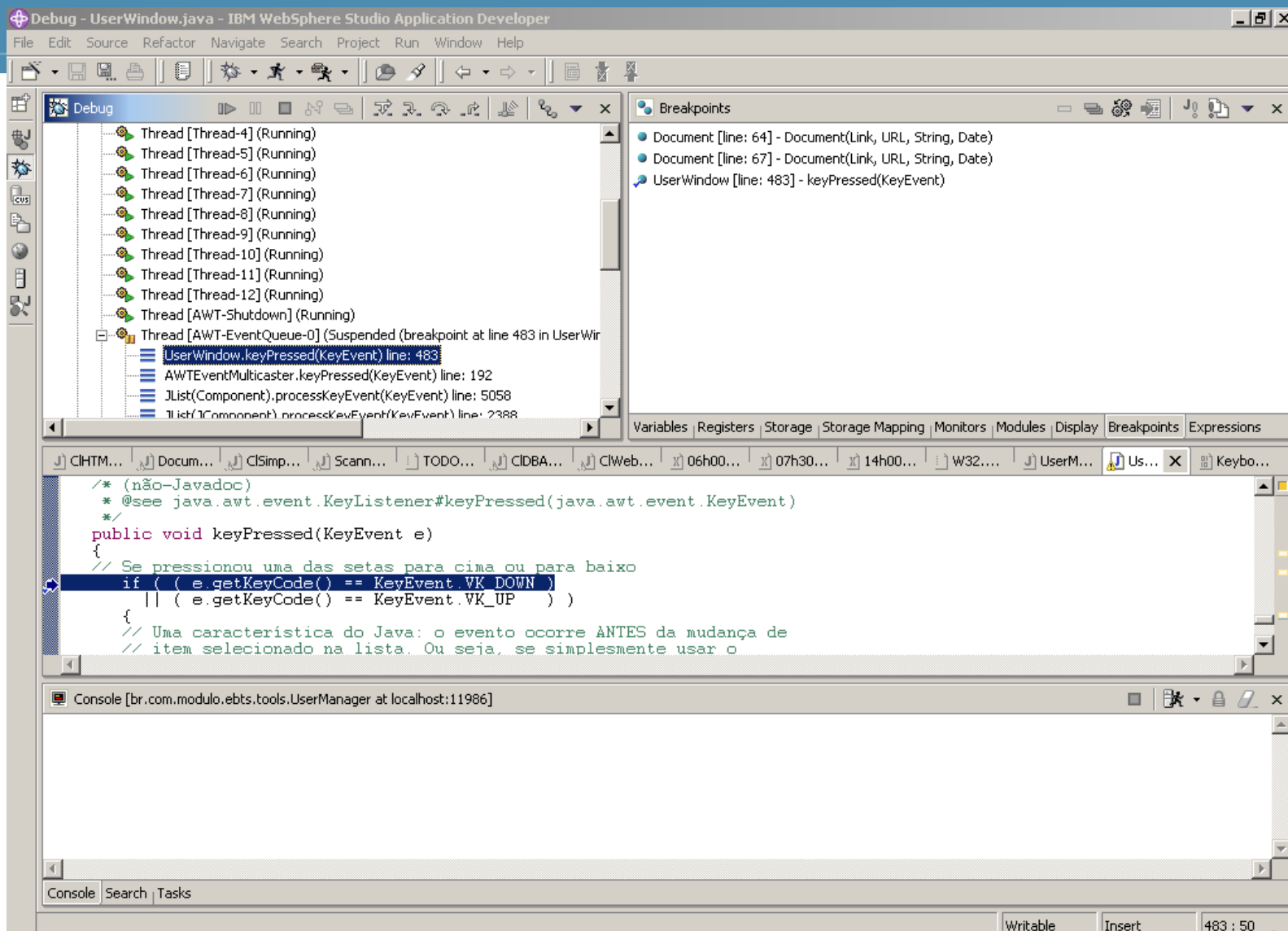
Conceitos Importantes

- Controles de execução:



- F5: Step Into
- F6: Step Over
- F7: Step Return
- F8: Resume
- CTRL + R: Run to line

Conceitos Importantes



Debug - UserWindow.java - IBM WebSphere Studio Application Developer

File Edit Source Refactor Navigate Search Project Run Window Help

Debug

- Thread [Thread-4] (Running)
- Thread [Thread-5] (Running)
- Thread [Thread-6] (Running)
- Thread [Thread-7] (Running)
- Thread [Thread-8] (Running)
- Thread [Thread-9] (Running)
- Thread [Thread-10] (Running)
- Thread [Thread-11] (Running)
- Thread [Thread-12] (Running)
- Thread [AWT-Shutdown] (Running)
- Thread [AWT-EventQueue-0] (Suspended (breakpoint at line 483 in UserWindow.java))

Breakpoints

- Document [line: 64] - Document(Link, URL, String, Date)
- Document [line: 67] - Document(Link, URL, String, Date)
- UserWindow [line: 483] - keyPressed(KeyEvent)

Variables Registers Storage Storage Mapping Monitors Modules Display Breakpoints Expressions

```
/* (não-Javadoc)
 * @see java.awt.event.KeyListener#keyPressed(java.awt.event.KeyEvent)
 */
public void keyPressed(KeyEvent e)
{
    // Se pressionou uma das setas para cima ou para baixo
    if ( ( e.getKeyCode() == KeyEvent.VK_DOWN )
        || ( e.getKeyCode() == KeyEvent.VK_UP ) )
    {
        // Uma característica do Java: o evento ocorre ANTES da mudança de
        // item selecionado na lista. Ou seja, se simplesmente usar o
```

Console [br.com.modulo.ebts.tools.UserManager at localhost:11986]

Console Search Tasks

Writable Insert 483 : 50



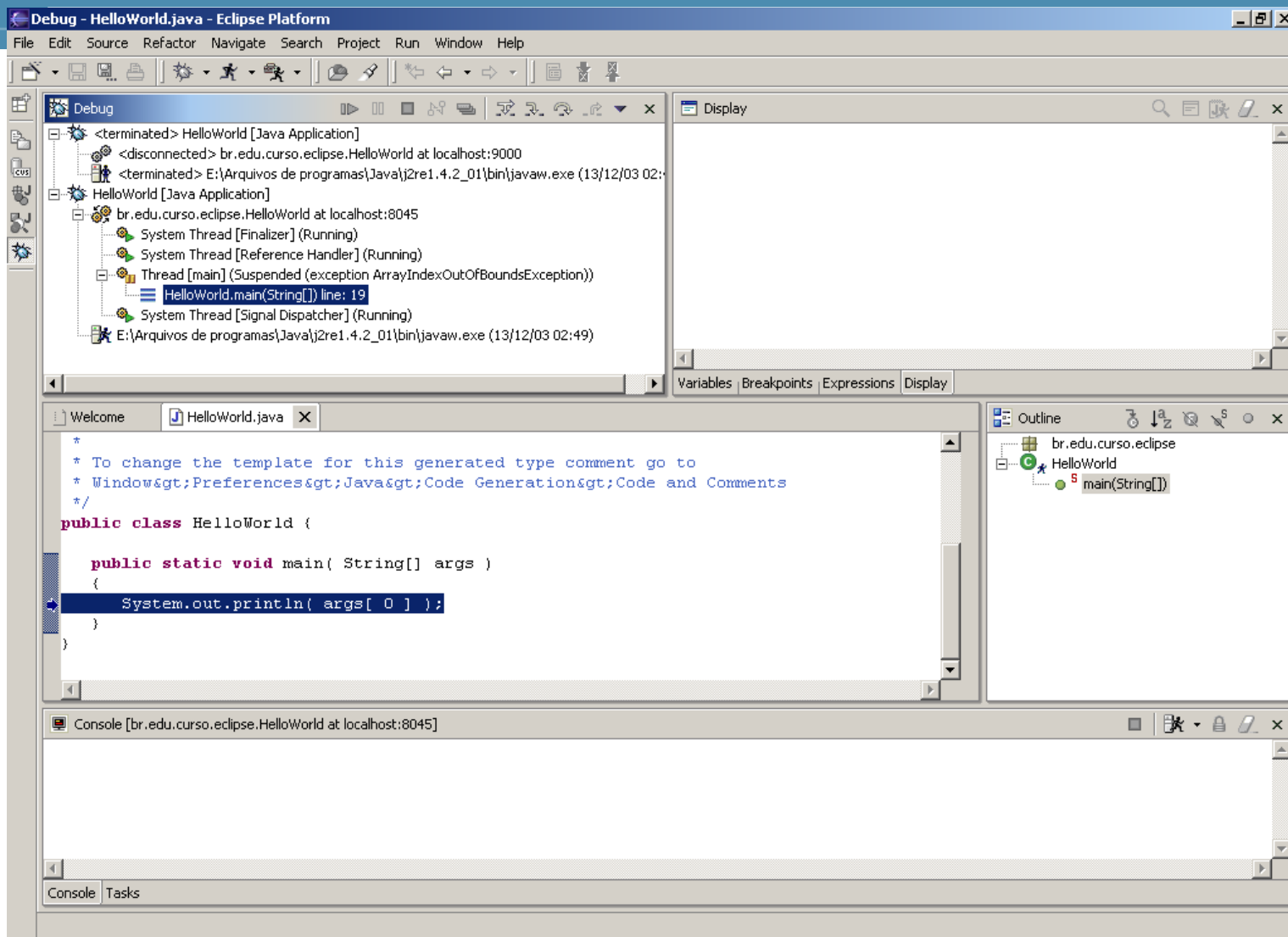
Conceitos Importantes

- Vamos estudar um exemplo prático de como o debugger se comporta com a ocorrência de uma exceção

```
public static void main( String[] args )  
{  
    System.out.println( args[ 0 ] );  
}
```



Conceitos Importantes





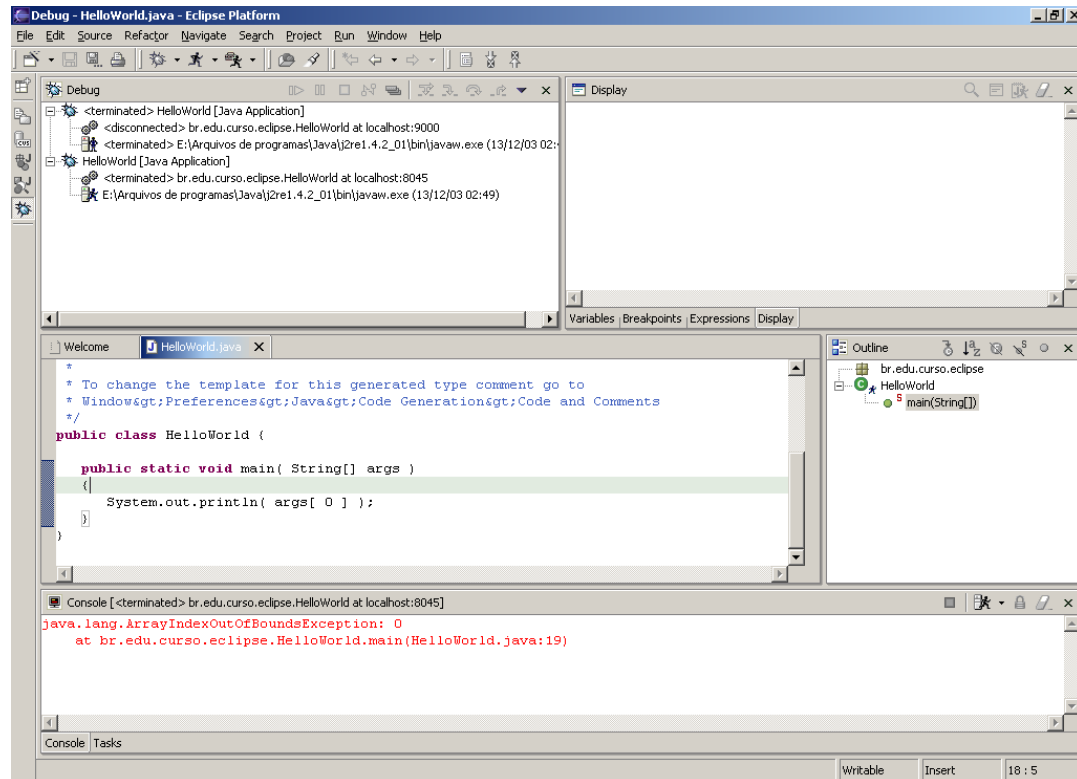
Conceitos Importantes

- O programa foi interrompido na linha em que ocorre a exceção, **imediatamente antes dela ser levantada.**
- A classe da exceção e a pilha de ativação (stack trace) encontram-se na visão de programas debugados.



Conceitos Importantes

- Se clicarmos em Resume (F8), a exceção será levantada



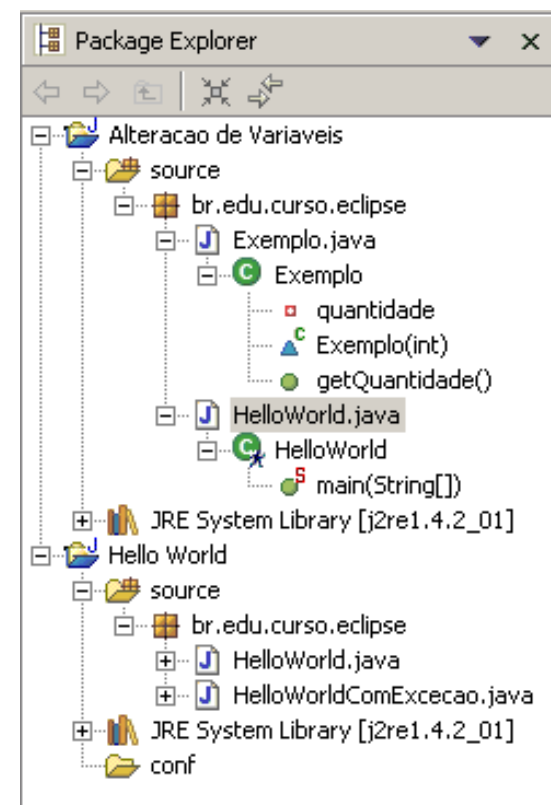


Conceitos Importantes

- É possível alterar os valores de atributos e variáveis locais através da visão de variáveis

Conceitos Importantes

- Vamos criar um novo projeto para testar essa feature
 - Um JavaBean com um atributo int de nome quantidade, com um método getQuantidade correspondente
 - Uma classe com main usando esse JavaBean





Conceitos Importantes

```
public class HelloWorld {  
  
    public static void main( String[] args )  
    {  
        int i = 0;  
        Exemplo ex = new Exemplo( 3 );  
        System.out.println( ex.getQuantidade() );  
    }  
}
```



Conceitos Importantes

```
public class Exemplo
{
    private int quantidade;

    Exemplo( int qtde )
    {
        quantidade = qtde;
    }

    public int getQuantidade()
    {
        return quantidade;
    }
}
```



Conceitos Importantes

- Executando em modo debug com um breakpoint em `System.out.println...`



Conceitos Importantes

Debug - HelloWorld.java - Eclipse Platform

File Edit Source Refactor Navigate Search Project Run Window Help

Debug

- HelloWorld [Java Application]
 - br.edu.curso.eclipse.HelloWorld at localhost:14071
 - System Thread [Finalizer] (Running)
 - System Thread [Reference Handler] (Running)
 - Thread [main] (Suspended (breakpoint at line 21 in HelloWorld))
 - HelloWorld.main(String[]) line: 21
 - System Thread [Signal Dispatcher] (Running)

E:\Arquivos de programas\Java\j2re1.4.2_01\bin\javaw.exe (13/12/03 14:30)

Variables

- args= String[0] (id=14)
- i= 0
- ex= Exemplo (id=16)
 - quantidade= 3

3

Variables Breakpoints Expressions Display

HelloWorld.java

```
/* Window>Preferences>Java>Code Generation>Code and Comments */
public class HelloWorld {

    public static void main( String[] args )
    {
        int i = 0;
        Exemplo ex = new Exemplo( 3 );
        System.out.println( ex.getQuantidade() );
    }
}
```

Outline

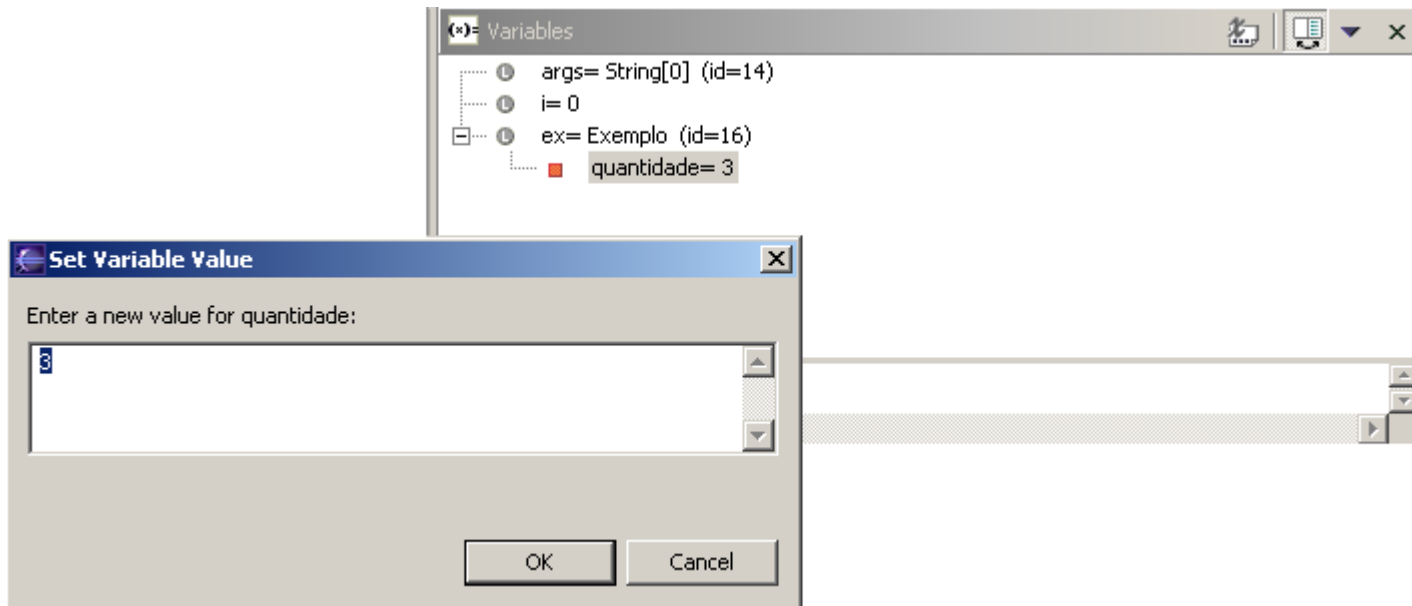
- br.edu.curso.eclipse
 - HelloWorld
 - main(String[])

Console [br.edu.curso.eclipse.HelloWorld at localhost:14071]

Console Tasks

Conceitos Importantes

- Dando duplo-clique em quantidade, é possível alterar seu conteúdo.





Conceitos Importantes

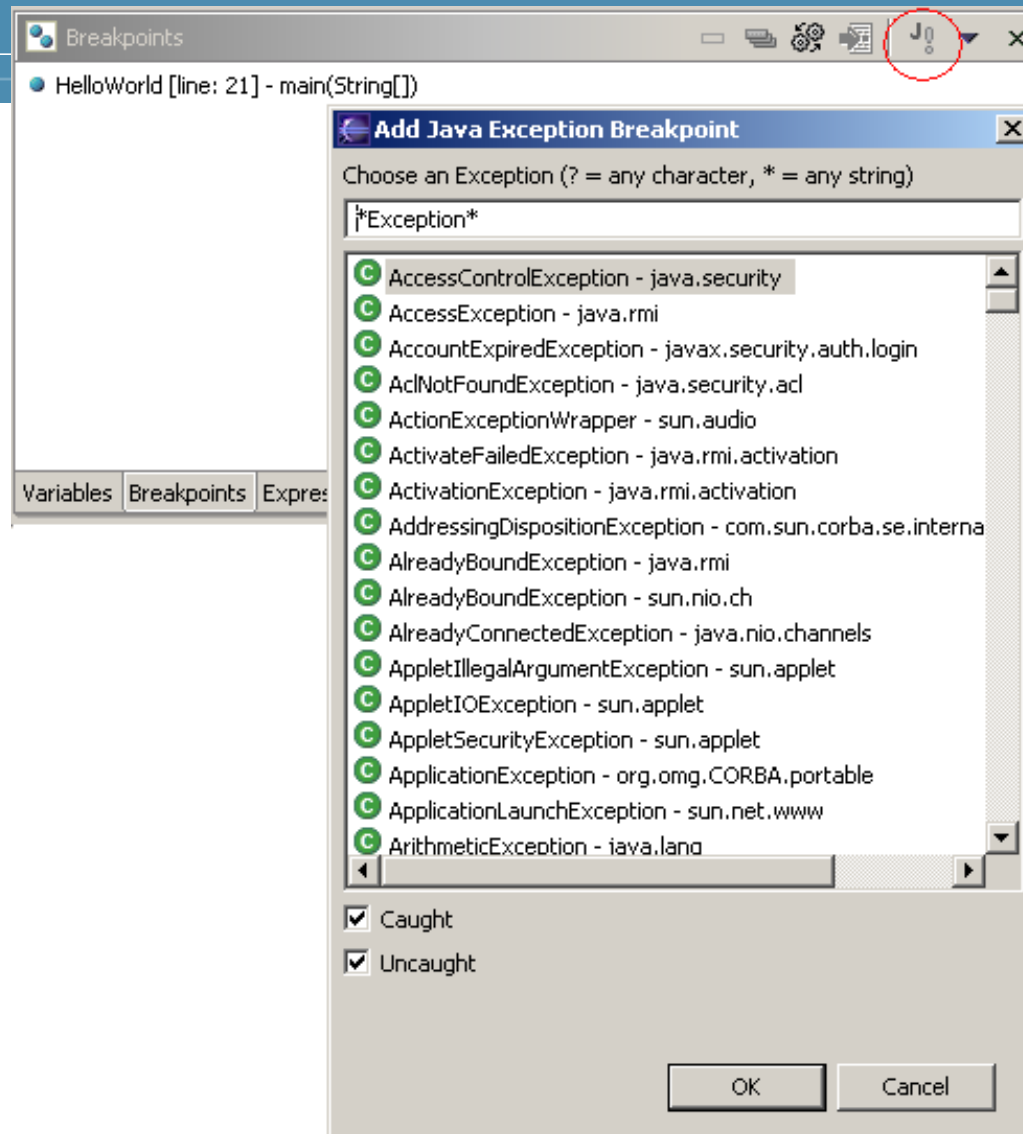
- Existem diversos tipos de breakpoints muito úteis
 - Nós vimos os de linha
 - Há o de método, atributo (watchpoint) e exceção



Conceitos Importantes

- Breakpoints
 - Método e atributo: clicar com botão direito do mouse sobre o elemento e selecionar a opção
 - Exceção: selecionar a opção na visão de breakpoints

Conceitos Importantes



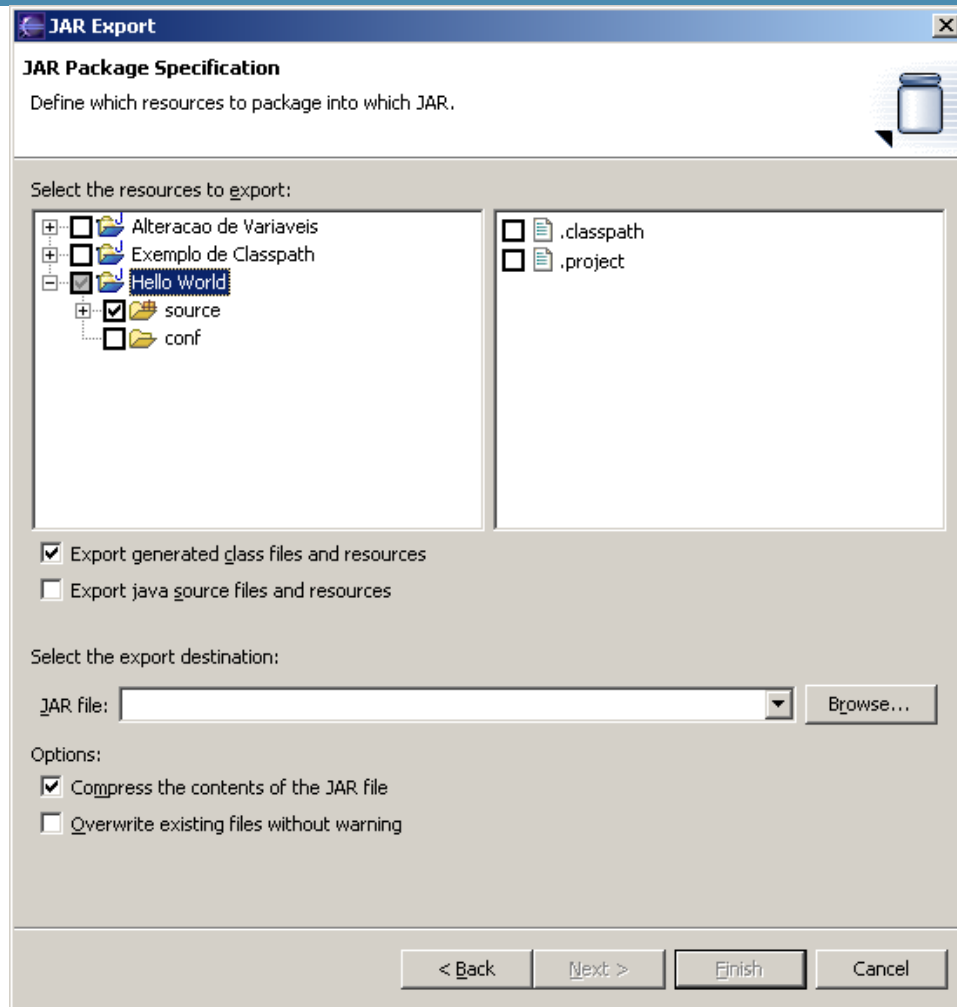


Conceitos Importantes

- E depois que todo o desenvolvimento está feito? Basta utilizar a ferramenta de exportação do Eclipse.
- O funcionamento é análogo ao da ferramenta de importação.
- Botão direito do mouse em cima do recurso que quer exportar

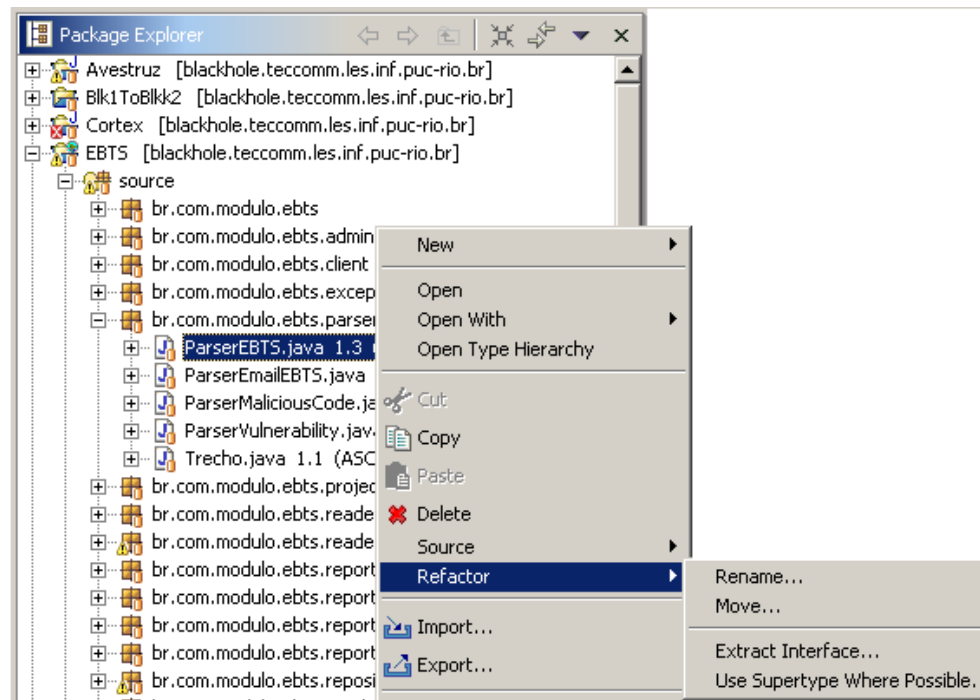
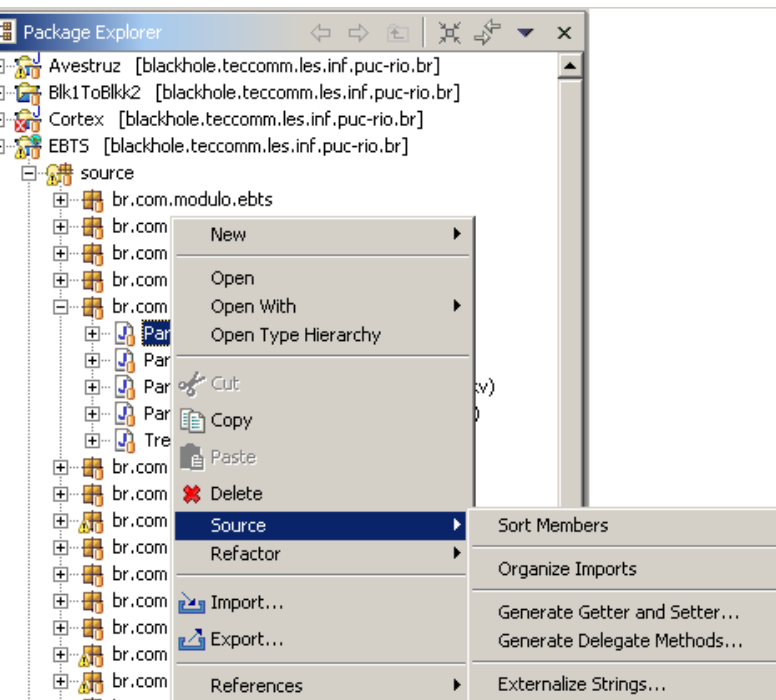
Conceitos Importantes

- Por exemplo, vamos gerar um .jar de um de nossos projetos





- Alguns padrões de reengenharia de código estão presentes no Eclipse. Eles são muito úteis!
 - Geração automática de getters e setters
 - Alteração do nome de classes, métodos, atributos e pacotes;
 - Mudança de métodos entre classes



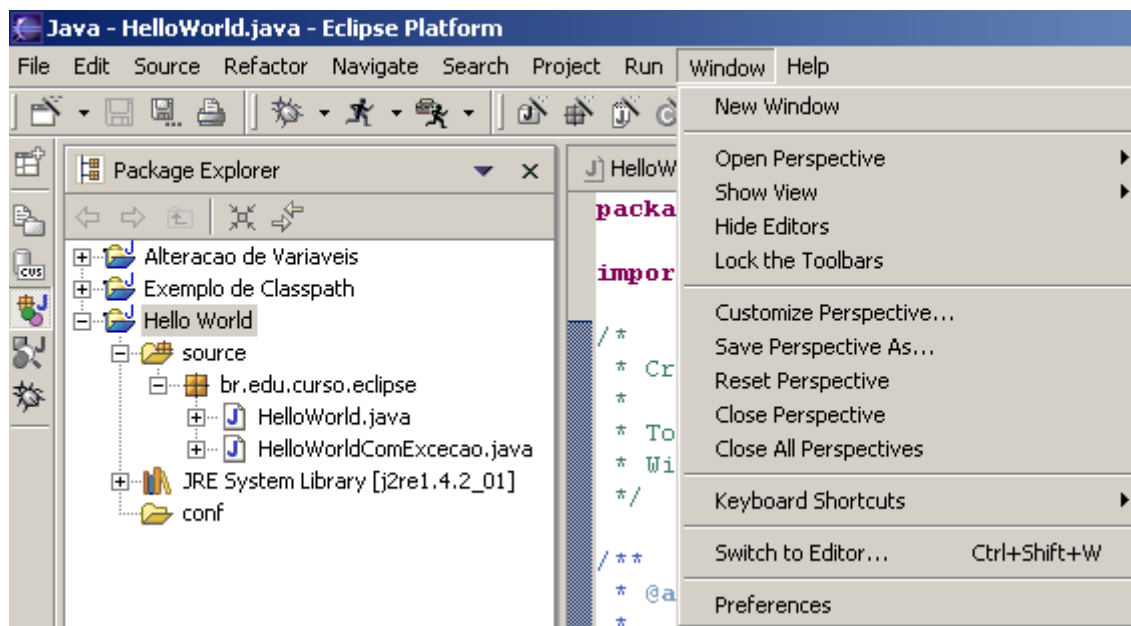


Conceitos Importantes

- O Eclipse possui diversas configurações gerais, que definem:
 - Aparência dos editores;
 - Aparência das visões;
 - Tipo de código objeto gerado;
 - Instruções para o code assistant;

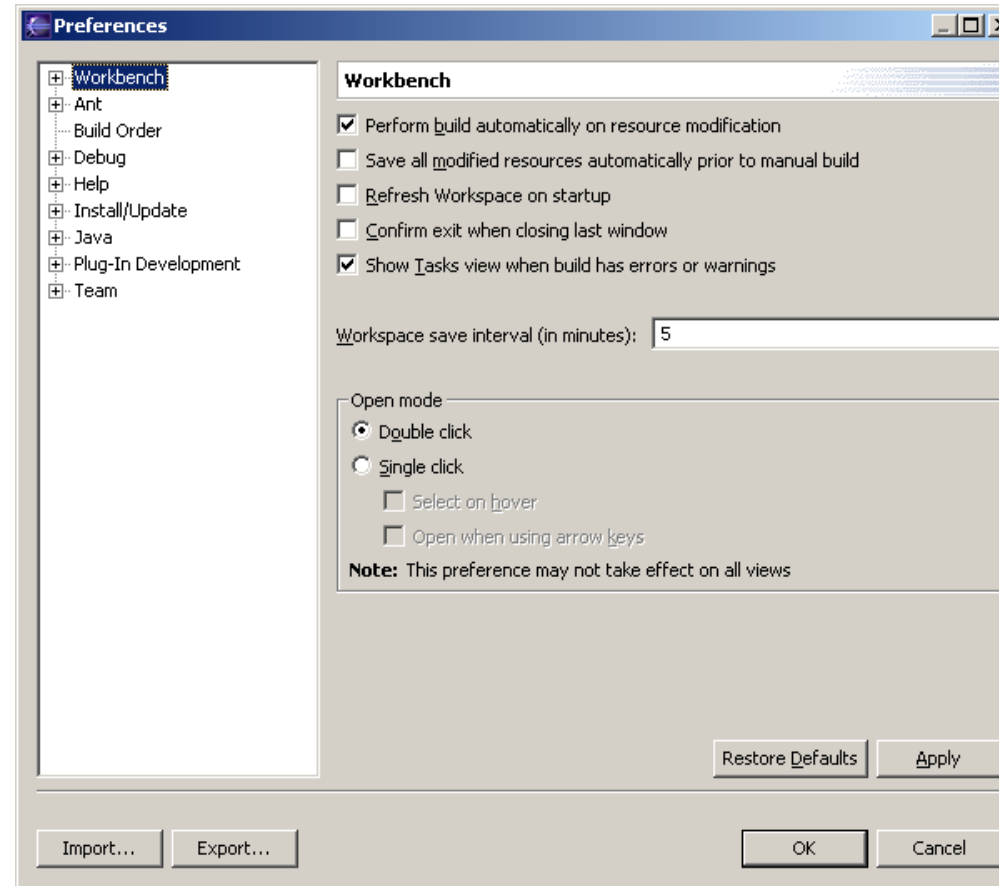
Conceitos Importantes

- Todas as configurações ficam no menu Window



Conceitos Importantes

- A opção “Preferences” é a mais utilizada. Ela define diversos parâmetros de configuração do ambiente Eclipse, dos editores, do compilador, etc.





Conceitos Importantes

- No mundo do desenvolvimento de software, é raro o desenvolvimento stand alone.
- É necessário compartilhar seu trabalho com outros desenvolvedores



Conceitos Importantes

- A dificuldade consiste em manter o código-fonte sempre atualizado
- Uma solução open-source para esse gerenciamento é o CVS (Concurrent Versions System)



Conceitos Importantes

- O CVS é um sistema de controle de versão de projetos.
- Ele manipula arquivos anotando as alterações neles realizadas. Cada mudança é marcada com um timestamp e um id de usuário. Um comentário permite ao usuário comunicar a razão das alterações.



Conceitos Importantes

- Essas informações são gravadas no **repositório** (que é um diretório de arquivos gerenciado pelo servidor CVS). Os dados são sempre inseridos em um **ramo** (branch) (o **ramo** principal é o **HEAD**).
- Cada alteração receberá um número de revisão individual. Novos arquivos começam com o número 1.1. É possível comparar ou substituir arquivos por outras revisões a qualquer momento.



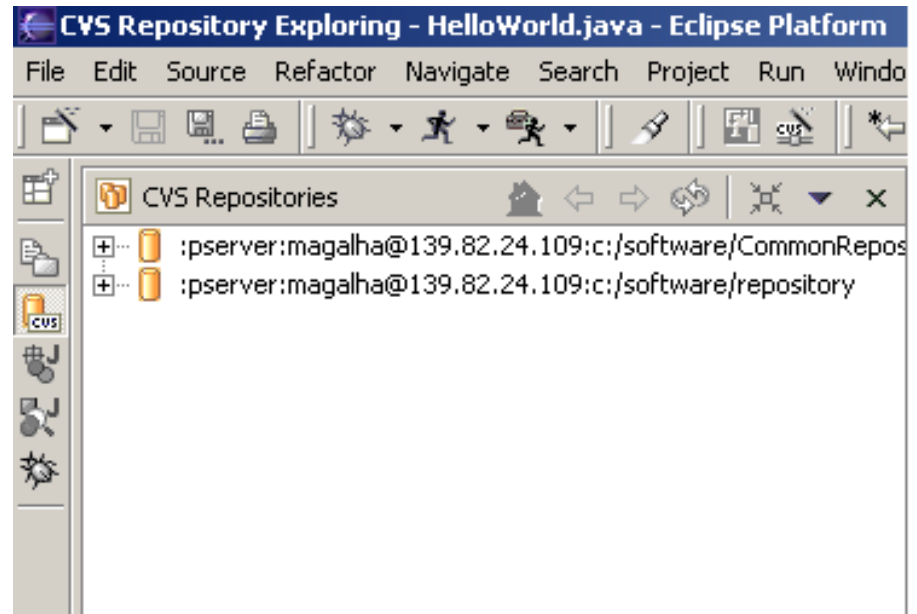
Conceitos Importantes

- Caso os recursos de seu projeto (que consistem de arquivos, possivelmente com números de revisão diferentes) tiverem um estado tal que valha a pena ser armazenado, é possível criar uma versão. Uma versão é um conjunto de arquivos com determinadas revisões.



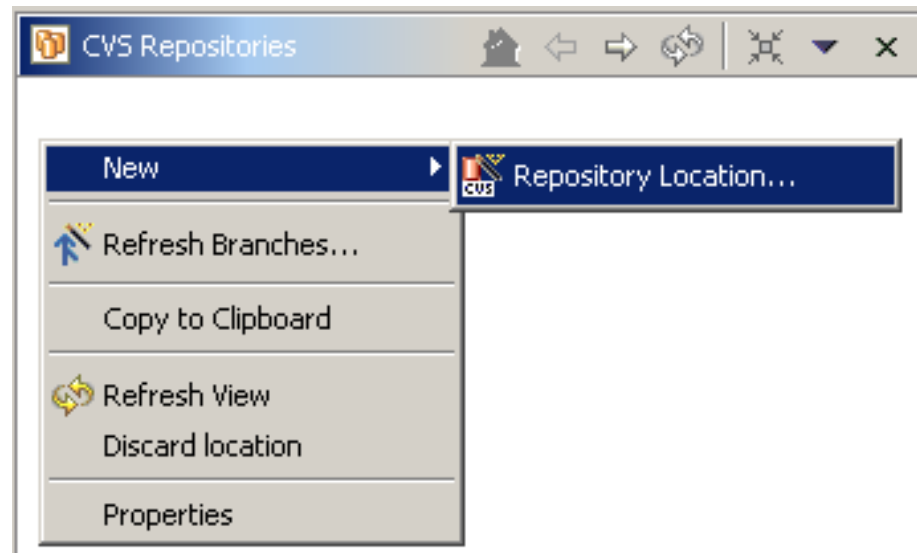
Conceitos Importantes

- O Eclipse já vem com suporte ao CVS: é a perspectiva de CVS



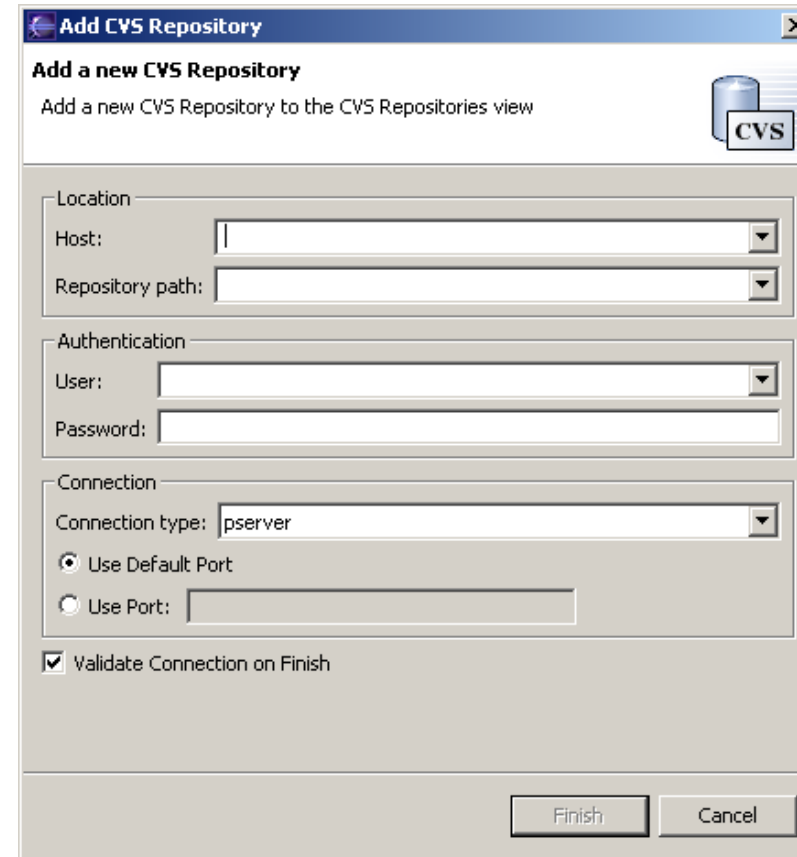
Conceitos Importantes

- Tudo começa com o cadastramento de um repositório
 - Botão direito do mouse sobre a visão de repositórios



Conceitos Importantes

- Host: nome da máquina servidora de CVS
- Repository path: caminho para o repositório (um diretório!)
- User: usuário no CVS
- Password: sua senha
- Connection type
 - pserver para password server
 - ext para sistemas de arquivos ext
 - extssh para autenticação ssh



Add CVS Repository

Add a new CVS Repository to the CVS Repositories view

Location

Host:

Repository path:

Authentication

User:

Password:

Connection

Connection type:

☒ Use Default Port

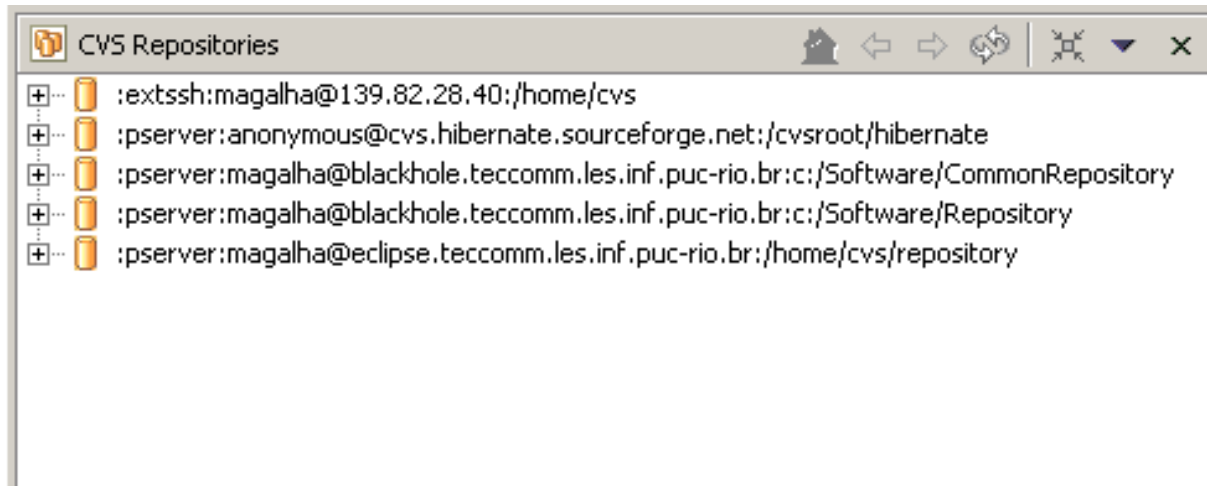
☐ Use Port:

☒ Validate Connection on Finish

Finish Cancel

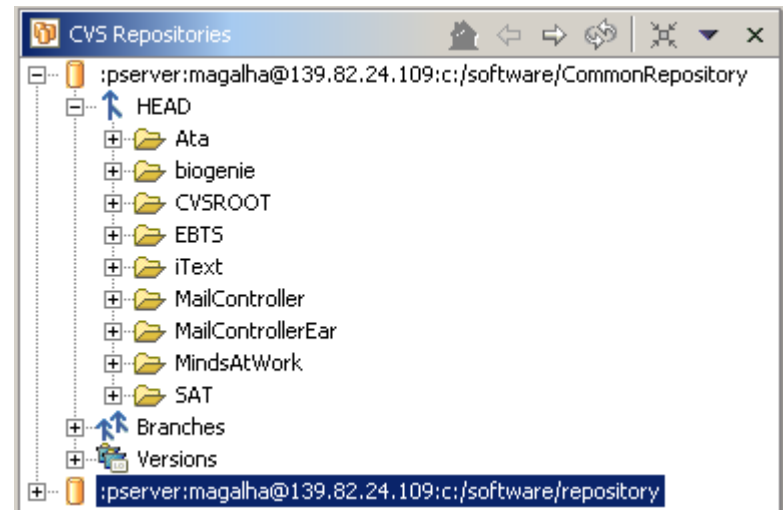
Conceitos Importantes

- É possível ter diversos repositórios



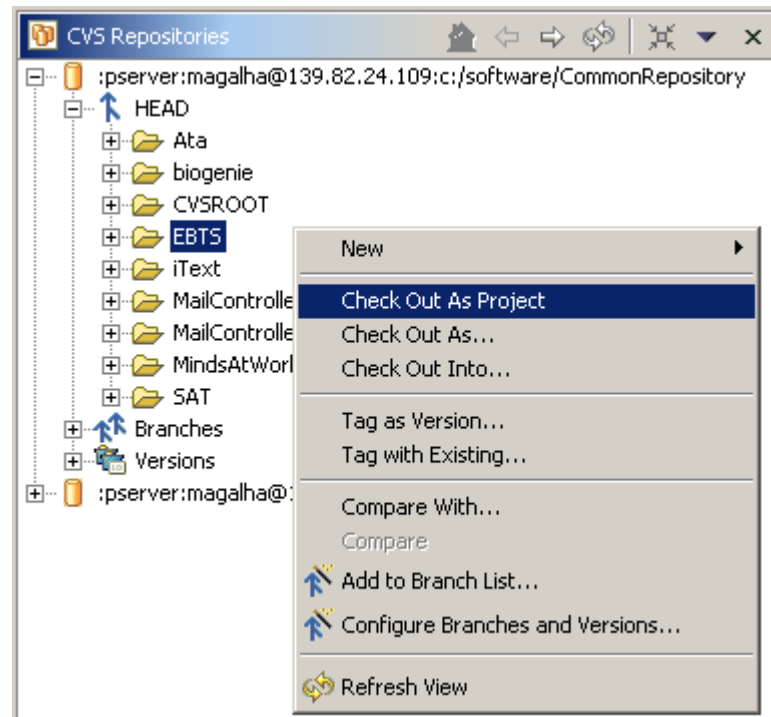
Conceitos Importantes

- Expandido um repositório, podemos ver suas ramificações.

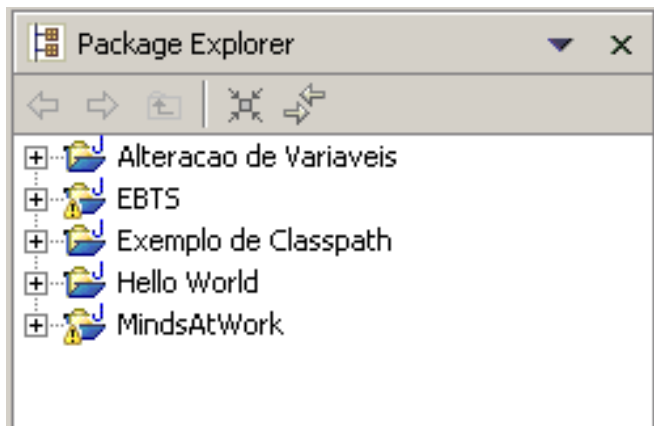
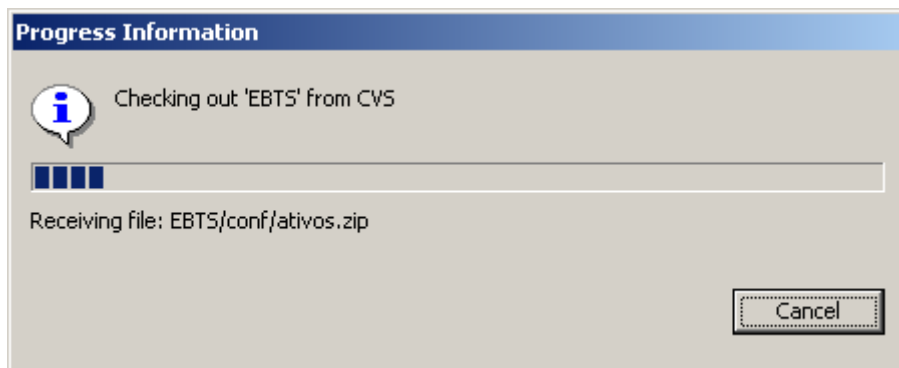


Conceitos Importantes

- É possível fazer “check out” nos projetos
 - É o momento em que se pega um projeto para o workspace

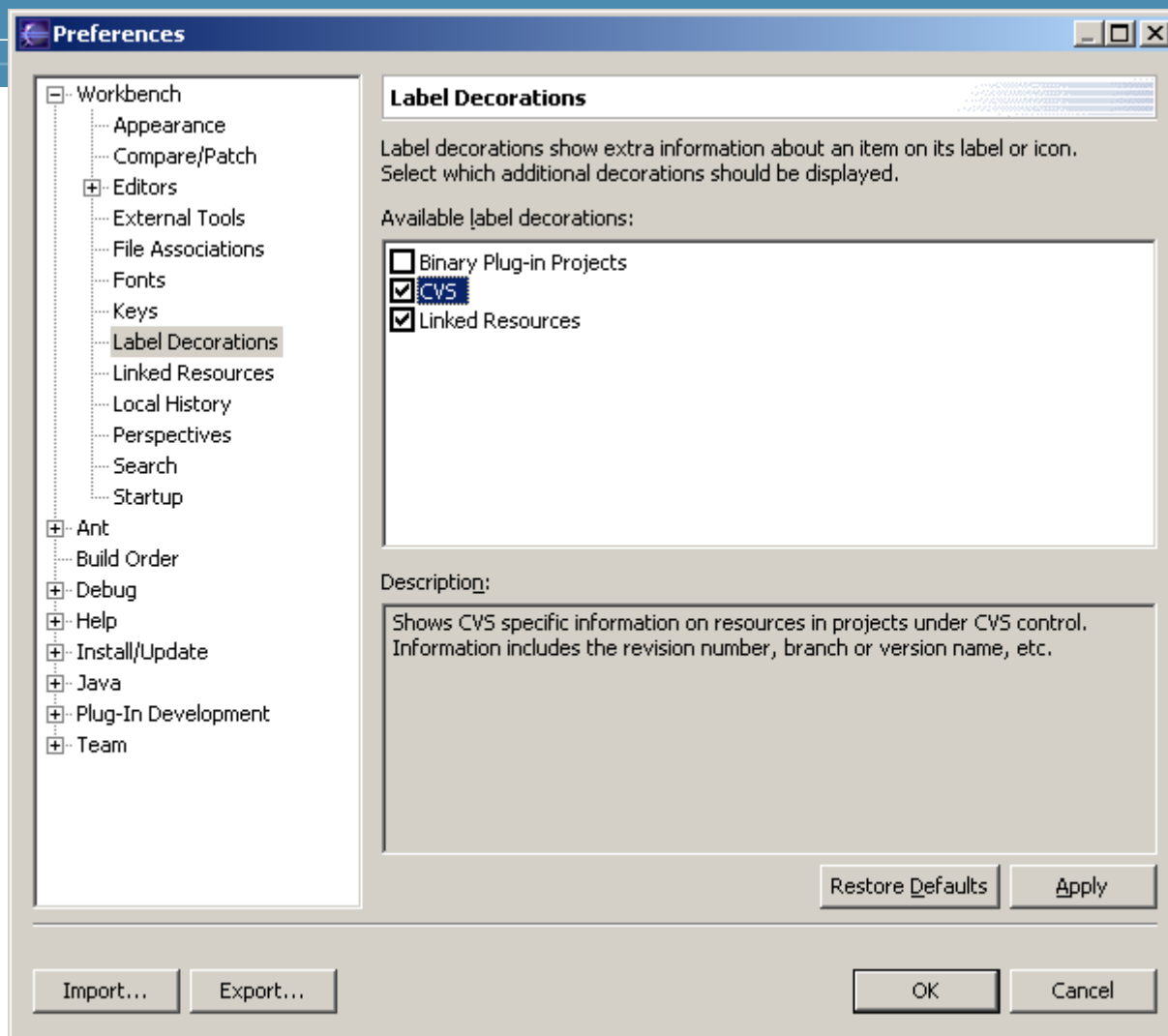


Conceitos Importantes



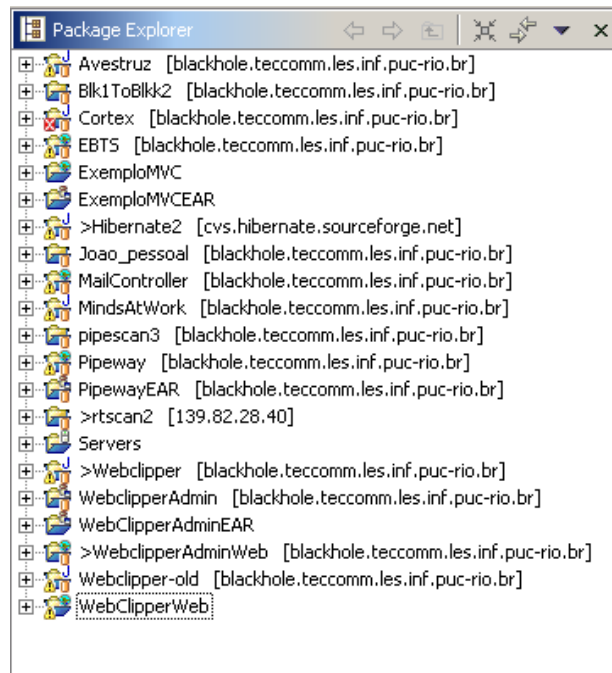
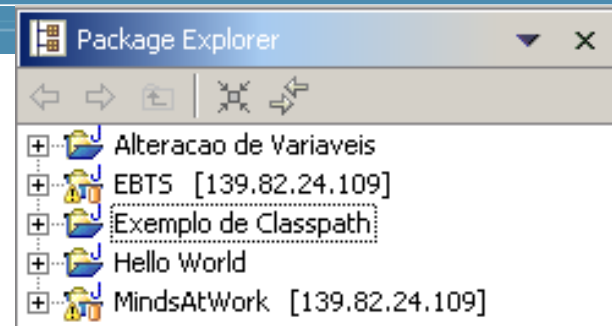
- O novo projeto aparece no workspace, junto com os demais projetos existentes.
- Porém, como saber quais projetos estão em um CVS, e quais não estão? Basta usar o label decoration do CVS!

Conceitos Importantes



Conceitos Importantes

- Note que o nome do servidor de CVS aparece ao lado de cada projeto
- Note ainda que, assim que algum recurso local sai de sincronia com o repositório, um sinal de ">" aparece ao lado do projeto





Conceitos Importantes

- Os comandos do CVS estão sob a opção “Team”
 - Mais usados: commit, update, synchronize, compare with
- Assim como é possível pegar projetos de um CVS, também é possível enviar projetos a um CVS
 - Desde que você tenha direitos para isso



Conceitos Importantes

- Ainda que você esteja desenvolvendo no estilo “euquipe”: **use sempre um sistema de controle de versão de seus projetos.**



Conceitos Importantes

- Dicas sobre como tirar o máximo de proveito do Eclipse:
 - Automatização de geração de código. Use-a!
 - Procure manter seus desenvolvimentos em projetos organizados
 - Pode ser interessante criar um projeto seu somente com as libs usadas (para que outros projetos façam referência).
 - Faça uso de um CVS. Sempre, mesmo que em desenvolvimento stand-alone



Conceitos Importantes

- Dicas sobre como tirar o máximo de proveito do Eclipse:
 - Fique atento aos plugins oferecidos (parte web)
 - Plugin para rodar tomcat
 - Plugin para modelagem em banco de dados
 - Plugin para o hibernate (conhecem? Vale a pena!)
 - Plugin para desenvolvimento web (LomboZ)
 - Plugin para uso do struts



Conceitos Importantes

- Dúvidas? Comentários?



Conceitos Importantes

- Muito obrigado!
 - magalha@mindsatwork.com.br