



eclipse

(c) Copyright Eclipse contributors and others, 2000, 2005. All rights reserved. Java and all Java-related trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S., other countries, or both. Eclipse is a trademark of the Eclipse Foundation, Inc.

Índice

Índice	2
Download do Eclipse	3
Instalação do Eclipse	4
Primeira execução do Eclipse	5
Instalação de plugins	6
Criação de projeto	7
Criação de um pacote	11
Criação de uma classe.	12
Execução de uma aplicação Java dentro do Eclipse	14
Javadoc no code completion	17
Debug	20
Export	23
Importando biblioteca	26
Criando e executando teste JUnit	29
Executando um Ant build	34
Utilizando o CVS	36
Dicas de configurações personalizadas e outras funcionalidades	46

Download do Eclipse

O Eclipse pode ser baixado em: <http://download.eclipse.org/>

Atualmente, a última versão estável do Eclipse é a 3.1.2 e pode ser encontrada em: <http://download.eclipse.org/eclipse/downloads/drops/R-3.1.2-200601181600/index.php>

Nota: Para desenvolvimento JEE, recomenda-se o download do Web Tools Project (WTP):

<http://download.eclipse.org/webtools/downloads/>

É recomendado, neste caso, o download do Eclipse já com WTP integrado (chamada all-in-one).

Atualmente, a versão mais nova do Eclipse com WTP é a 1.01:

<http://download.eclipse.org/webtools/downloads/drops/R-1.0.1-200602171228/>

O Eclipse tem cerca de 100Mb e com o WTP passa a ter cerca de 180Mb.

Instalação do Eclipse

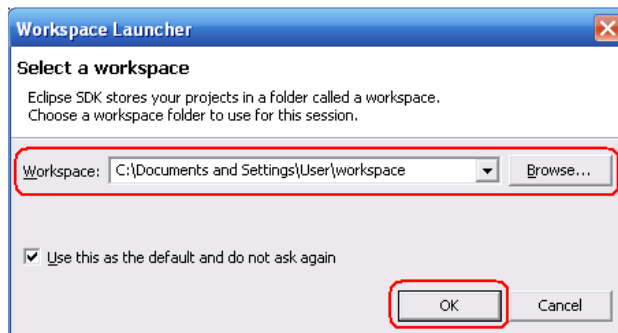
O Eclipse é distribuído em um arquivo compactado, em zip ou tar.gz. Para ser utilizado, ele deve ser descompactado, utilizando qualquer descompactador que suporte estes formatos, como o 7-Zip (<http://www.7-zip.org/>).

O Eclipse não necessita de instalação. Depois de descompactado, mova os arquivos para o diretório em que desejar manter o Eclipse.

Primeira execução do Eclipse

O Eclipse deve ser executado a partir do diretório onde foi instalado, através do arquivo "eclipse.exe" no Windows ou "eclipse" no caso de ter sido baixada a versão para Linux ou através de algum link para estes arquivos. No caso do Linux, o arquivo "eclipse" deve ter permissão de execução (utilize o comando `chmod +x eclipse`).

Na primeira execução do Eclipse, este pedirá o caminho para um diretório "Workspace". É neste diretório que o Eclipse salvará as preferências de utilização e também será o lugar padrão para os projetos. Uma mesma instalação do Eclipse pode utilizar diferentes workspaces, com configurações diferentes, por exemplo, para cada usuário.



Este diretório precisa necessariamente de permissão de escrita, ao contrário do diretório onde o Eclipse foi instalado. Porém, para que o Eclipse possa recuperar a informação da localização do último workspace utilizado e não perguntar esta informação na próxima instalação é necessário que o diretório de instalação do Eclipse também tenha permissão de escrita. Caso contrário o Eclipse perguntará a localização do workspace toda vez que for iniciado, embora tenha sido definido um Workspace default. Apesar disto, o Eclipse funciona corretamente.

A localização do workspace do Eclipse pode ser alterada posteriormente, através do próprio Eclipse.

Instalação de plugins

O Eclipse é conhecido pela vasta variedade de plugins que estendem sua funcionalidade. Os plugins para o Eclipse podem ir desde examinadores de aspectos de qualidade de código, refactoring, UML, construtores de interface gráfica WYSIWYG até tocadores de MP3 e instant messengers.

O Eclipse por padrão já traz instalados plugins para Ant (<http://ant.apache.org/>), JUnit (<http://www.junit.org>) e outros, mas muitos outros plugins podem ser encontrados nos sites Eclipse Plugin Central (<http://www.eclipseplugincentral.com/>), EclipsePlugins (<http://www.eclipse-plugins.info/>), SourceForge.net (<http://sourceforge.net/>) além de sites dos desenvolvedores de outros plugins.

É importante, antes de qualquer coisa, observar no site do plugin, sua compatibilidade com a versão do Eclipse que você esteja usando. Como tanto o site Eclipse Plugin Central quanto o EclipsePlugins apenas centralizam informações sobre plugins desenvolvidos por terceiros, é possível que as informações sobre alguns plugins possam estar desatualizadas e portanto sempre convém olhar o site do desenvolvedor do plugin desejado, encontrado na mesma página que fala sobre o plugin.

A instalação dos plugins costuma ser, em geral, bastante simples, bastando que o Eclipse seja fechado e o plugin copiado para o diretório "plugins" dentro do diretório onde o Eclipse foi instalado. Alguns plugins pedem ainda que alguns diretórios sejam copiados para o diretório "features" também dentro do diretório onde o Eclipse foi instalado.

Alguns poucos plugins acompanham instalador e diferentes formas de instalação. As instruções de instalação devem vir com o plugin ou estar presentes no site do desenvolvedor.

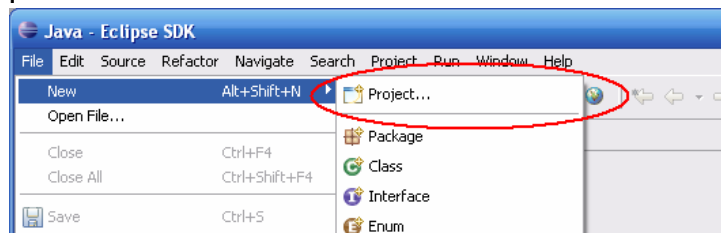
Criação de projeto

Para iniciar o desenvolvimento no Eclipse, primeiro é necessário criar e configurar um projeto.

Um projeto é, essencialmente, um conjunto de classes, separadas em pacotes e as bibliotecas necessárias para seu desenvolvimento.

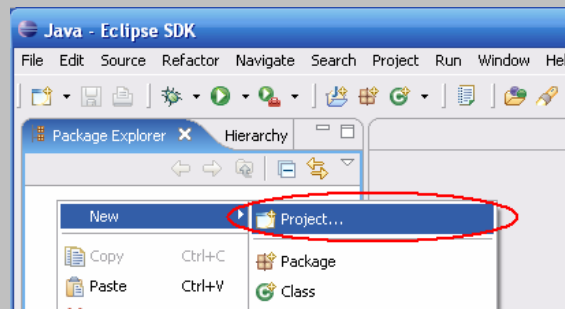
As telas seguintes podem se apresentar ligeiramente diferentes em instalações diferentes do Eclipse, por possuírem outros plugins. Estas telas foram capturadas da versão 3.1.2 do Eclipse com Web Tools Project 1.0.1 integrado.

Para criar um projeto, deve-se selecionar no menu "File" a opção "New" e em seguida "Project...".

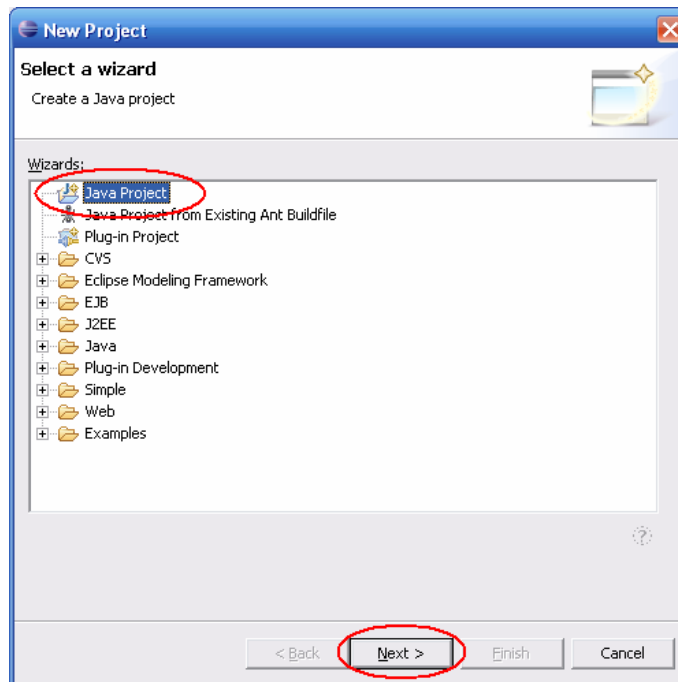


Nota: O menu de contexto traz muitas opções interessantes e é mais acessível que os menus.

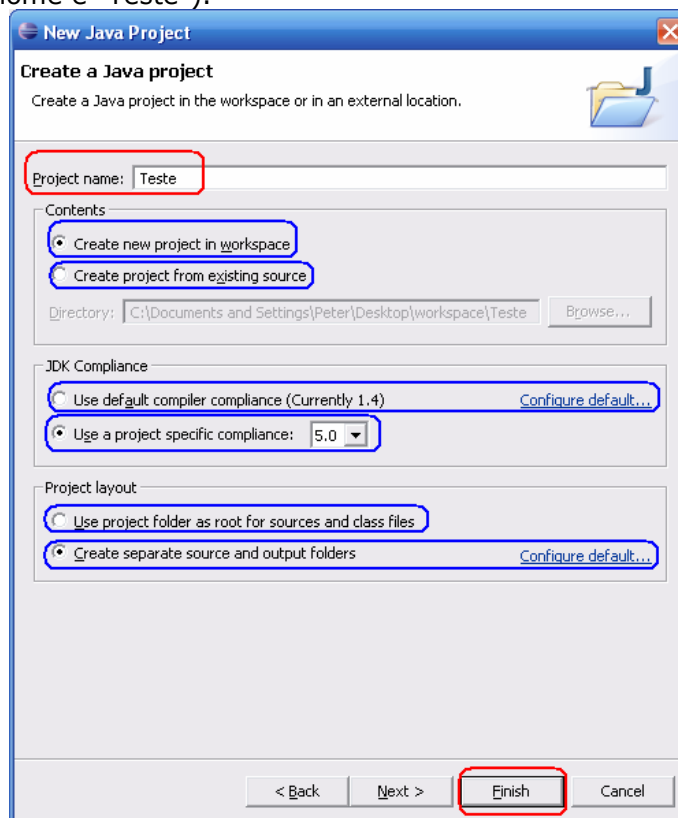
Como alternativa, o menu de contexto do painel "Package Explorer" fornece esta opção.



Selecione "Java Project" e clique em "Next".



Na tela que se segue há muitas informações úteis a serem fornecidas, mas o mínimo exigido é o preenchimento do nome do projeto, circulado em vermelho (neste exemplo, o nome é "Teste").



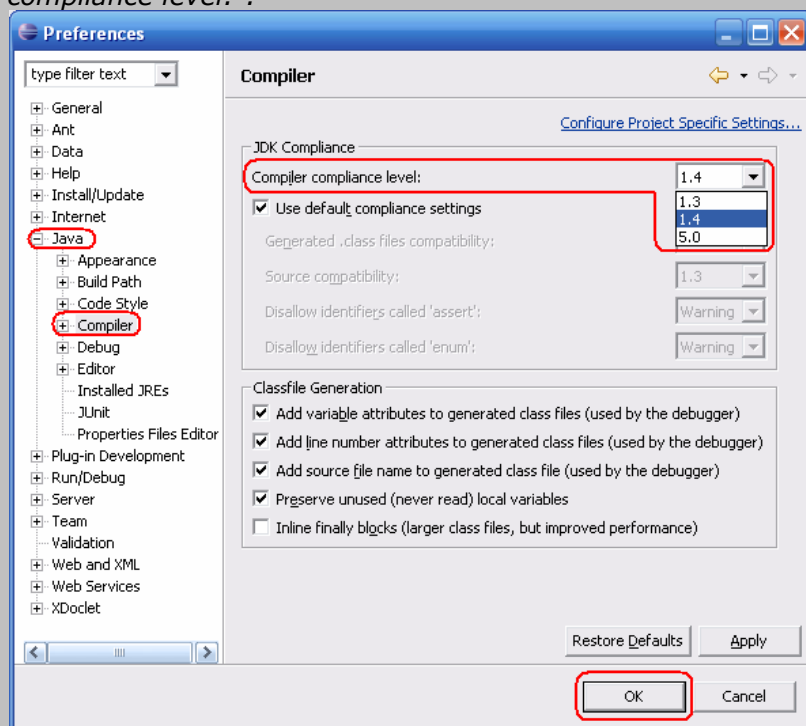
Em azul, dentro do painel "Contents", temos "Create new project in workspace", que criará um diretório com o nome do projeto dentro do workspace. É neste diretório que ficarão os arquivos do projeto.

Nota: caso haja um projeto do eclipse com o mesmo nome, ele aproveitará as informações e criará um projeto a partir daqueles arquivos. Isto pode acontecer caso um projeto tenha sido copiado de outro workspace. É uma forma simples e eficiente de copiar um projeto para outro workspace, mantendo todas as suas configurações. Caso o projeto tenha configurações "default" do workspace, essas configurações não serão mantidas, sendo substituídas pelas configurações "default" do workspace atual.

Selecionando "Create project from existing source" pode-se informar o caminho de um projeto até mesmo fora de um workspace.

No painel "JDK Compliance" temos "Use default compiler compliance (Currently 1.4)" que fará com que o projeto seja compilado para a versão configurada como sendo a versão padrão para compilação neste workspace. Para modificar o default, clique em "Configure default...". Esta alteração se refletirá em todos os outros projetos neste workspace que estiverem usando a configuração "default". Em "Use a project specific compliance:" pode ser selecionada uma configuração específica para este projeto, igual ou diferente da "default".

Nota: a configuração default também pode ser modificada no menu "Window", selecionando "Properties...". Em seguida, no nó "Java" selecionando "Compiler", a opção correspondente é "Compiler compliance level:".

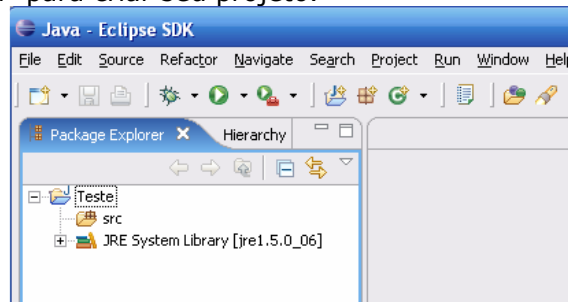


No painel "Project Layout", podemos modificar a estrutura do projeto. É recomendado que se separe o código fonte dos binários gerados, selecionado "Create separate source and output folders", ao invés da opção padrão, que colocará os dois

tipos de arquivos na raiz do projeto, juntos na mesma estrutura de pastas. Esta recomendação se deve ao fato de que o processo de geração de releases fica mais simplificado, especialmente quando se utilizará algum tipo de ferramenta para automatizá-lo.

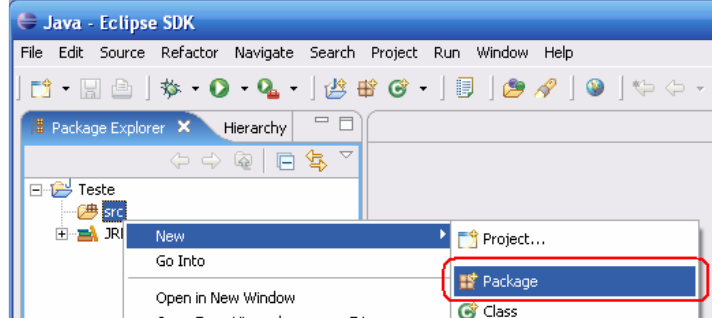
Também se pode modificar a estrutura de pastas de código-fonte e binários (clicando em "Configure default..."), mas é recomendado manter a estrutura padrão, pois esta é a mais usada pelos desenvolvedores. Esta estrutura se trata de um diretório "src" onde ficará o código-fonte da sua aplicação e um diretório "bin", onde ficarão os arquivos binários gerados a partir do código-fonte presente em "src", ambas na raiz do diretório do projeto. O Eclipse cuidará automaticamente da compilação e de manter o diretório "bin" em sincronia com o "src", de forma totalmente transparente para o desenvolvedor.

Clique em "Finish" para criar seu projeto.

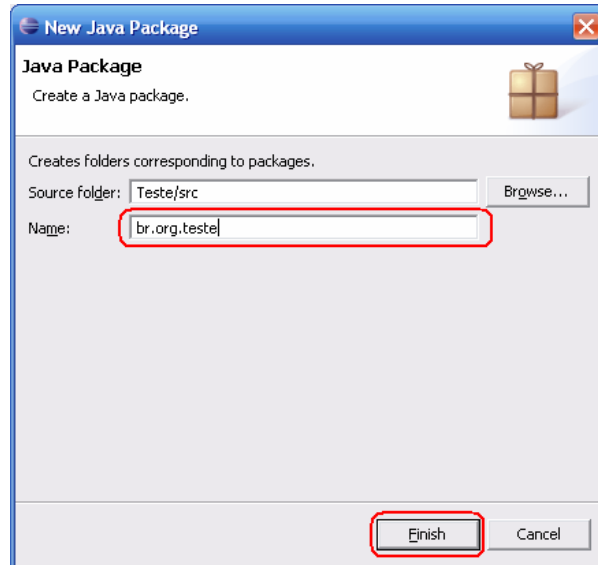


Criação de um pacote

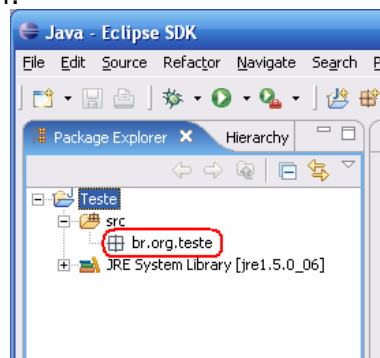
Para criar um pacote, deve-se clicar com o botão direito na pasta "src", selecionar "New" e em seguida "Package". (Isto pode ser feito também selecionando a pasta "src" e selecionando no menu "File" a opção "New" e "Package").



Na tela que se segue, deve ser preenchido o nome do pacote, que pode ter quantos tokens forem necessários.

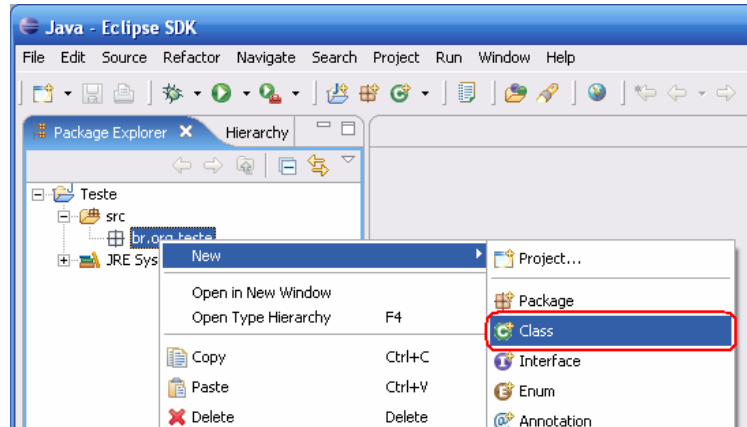


Em seguida, apertando o botão "Finish" será concluída a criação do pacote, que deverá se apresentar assim:

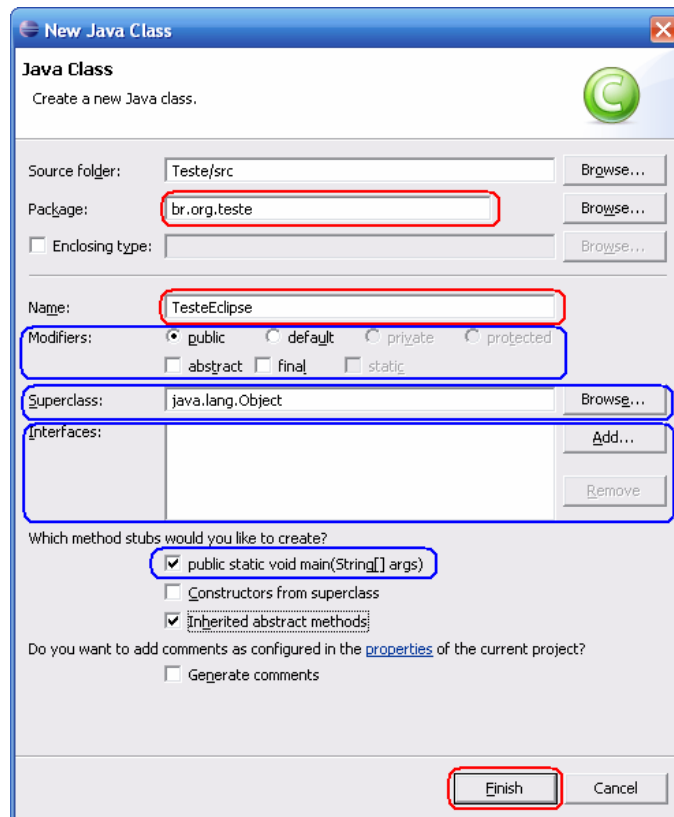


Criação de uma classe.

Clicando com o botão direito sobre o pacote onde deseja criar a classe (neste caso foi escolhido "br.org.teste", que foi o pacote criado no tópico anterior), selecionando "New" e em seguida "Class". Novamente, de forma análoga, no menu "File" encontra-se "New" e "Class".



Na tela que se abrirá, deve ser preenchido o nome da classe no campo identificado com "Name". O campo "Package" já vem preenchido com o nome do pacote que foi clicado para se criar essa classe. Este campo pode ser modificado livremente e caso o pacote não exista, será criado no momento em que a classe for criada.



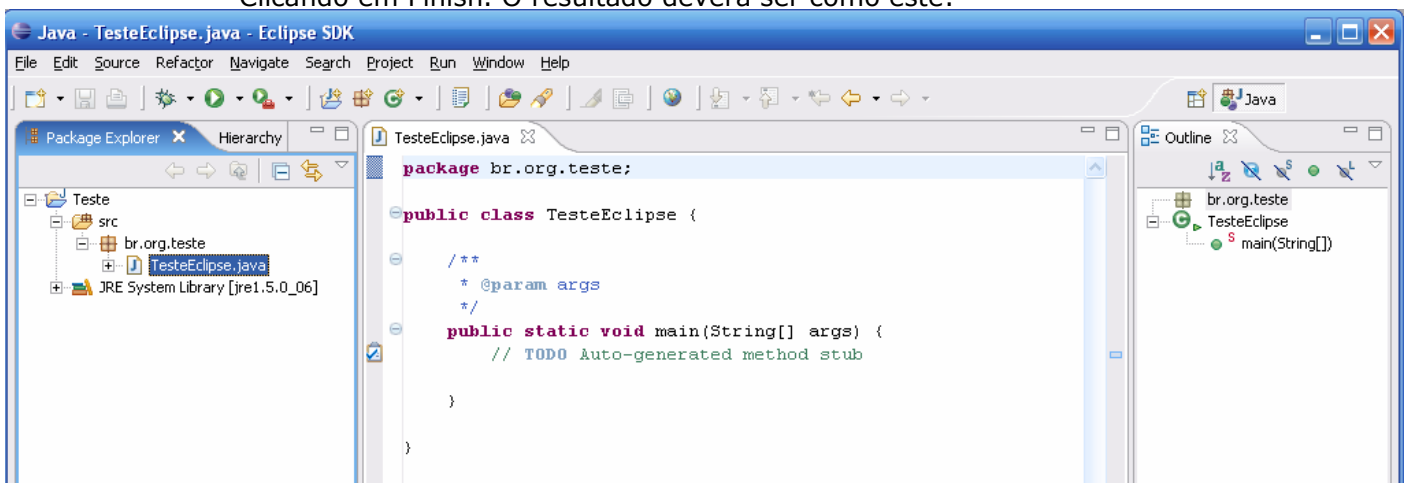
Em azul pode-se selecionar os "Modifiers" (modificadores da classe).

“Superclass” é a classe da qual a classe que estamos criando herdará. Por padrão este campo vem preenchido com “Object”. Clicando em “Browse”, se pode escolher qualquer uma classe da API do Java, de bibliotecas utilizadas no desenvolvimento deste projeto e de classes do projeto em si.

“Interfaces” têm a mesma finalidade de “Superclass”, mas desta vez para adicionar interfaces ao invés de classes. “Add” tem a mesma funcionalidade que browse, mas neste caso, “Add” acrescenta uma interface, diferente do “Browse”, que altera a classe que será estendida. “Add” acrescenta mais uma interface à lista de interfaces. Para remover uma interface, esta deve ser selecionada e em seguida deve ser clicado o botão “Remove”.

“Which method stubs would you like to create?” Ajuda na declaração de métodos. Neste caso, está sendo selecionado o método principal “public static void main(String[] args)”, para que se possa, em um próximo tópico, executar esta classe.

Clicando em Finish. O resultado deverá ser como este:



A criação de interfaces e “enums” é muito semelhante e por isso não será mostrada aqui.

Execução de uma aplicação Java dentro do Eclipse

Antes de executar a classe que foi criada no item anterior, um código deve ser inserido no método principal para que algum resultado da execução possa ser mostrado.

Um código de exemplo segue aqui:

```
if (args.length==0){
    System.out.println("Hello Eclipse!");
} else {
    System.out.println("Hello " + args[0] + " " + args[1]);
}
```

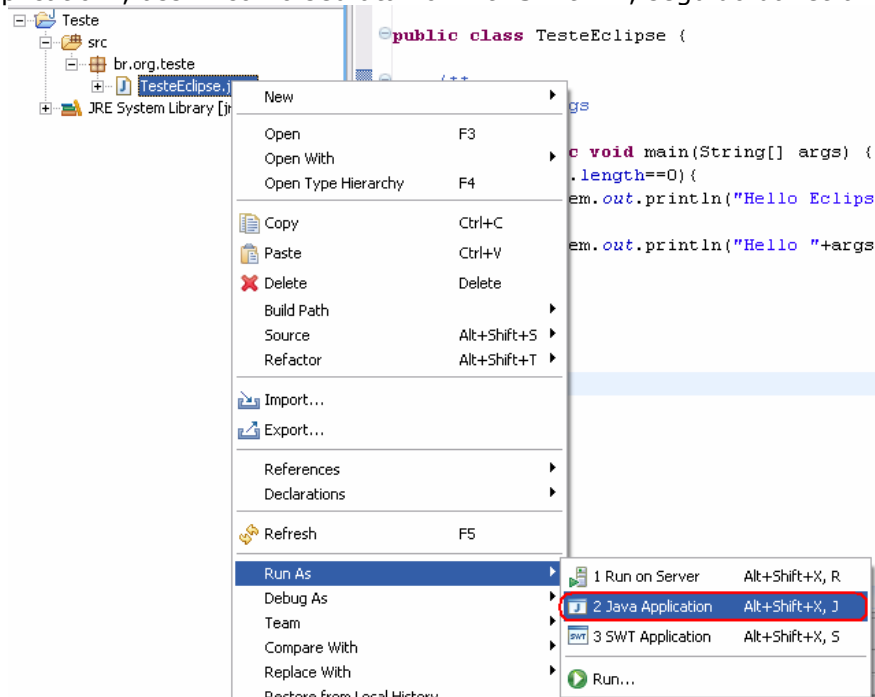
E a classe deve ficar como esta:

```
package br.org.teste;

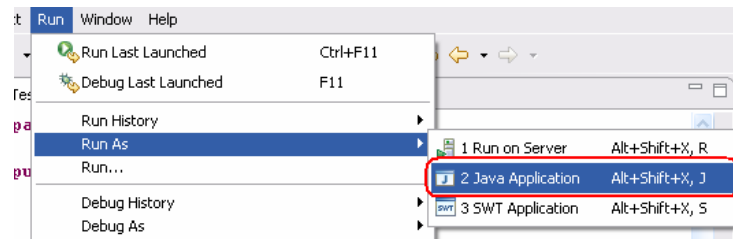
public class TesteEclipse {

    /**
     * @param args
     */
    public static void main(String[] args) {
        if (args.length==0){
            System.out.println("Hello Eclipse!");
        } else {
            System.out.println("Hello " + args[0] + " " + args[1]);
        }
    }
}
```

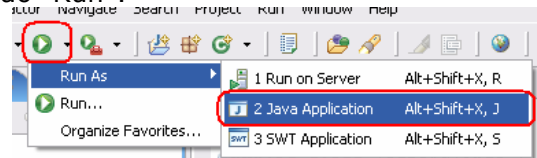
Clicando com o botão direito sobre a classe, selecionando "Run As", pode-se ver "Java Application", assim como seu atalho "Alt+Shift+X", seguido da letra "J".



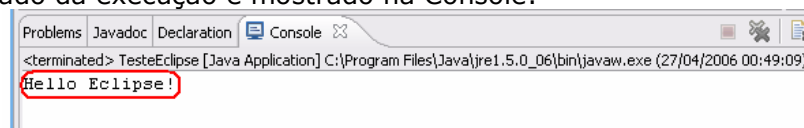
Este menu também pode ser acessado em "Run", "Run As" e "Java Application".



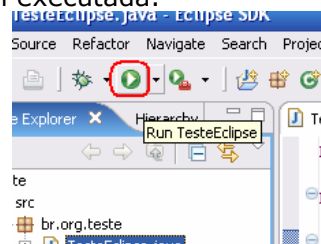
Assim como o botão "Run":



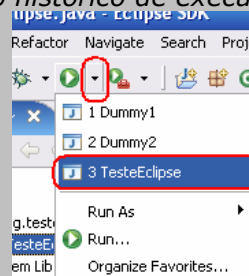
O resultado da execução é mostrado na Console:



Para executar novamente a aplicação, basta pressionar o botão "Run", que executa a última aplicação que foi executada.

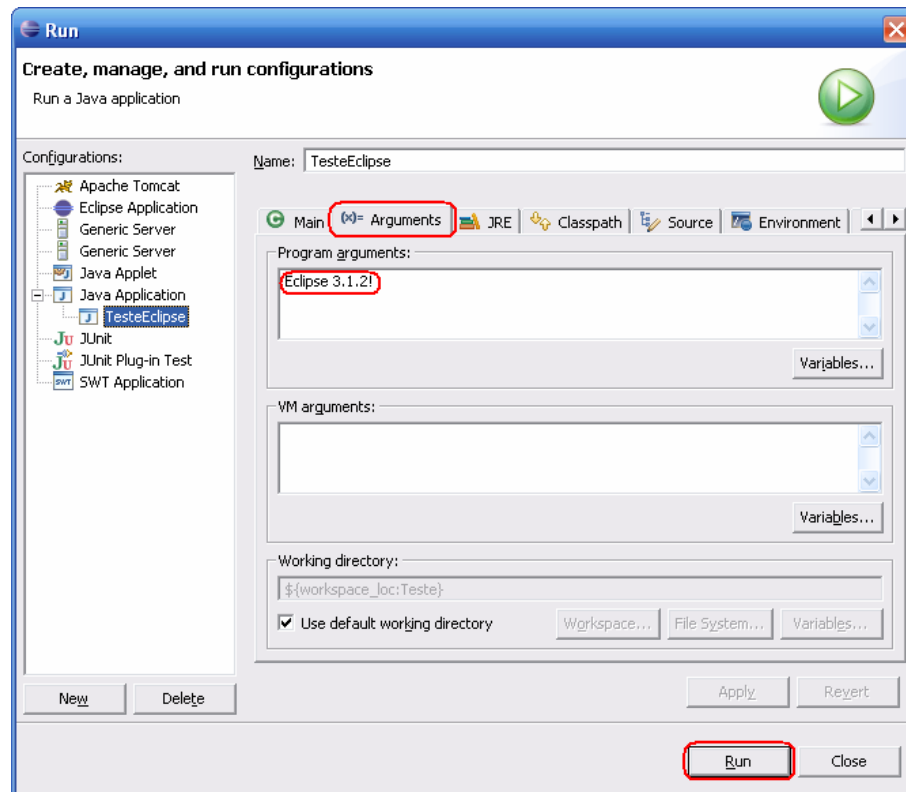


Nota: Para executar uma aplicação que foi executada anteriormente à última, basta selecioná-la no histórico de execuções, no botão "Run":



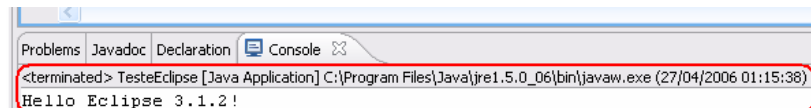
Clicando com o botão direito sobre a classe e selecionando "Run As", também podemos selecionar a opção "Run..." para mais opções na execução da aplicação.

Analogamente, uma tela como a que se apresentará também pode ser alcançada através do menu "Run" ou do botão "Run", selecionando-se a opção equivalente.



Na aba "Arguments", por exemplo, argumentos podem ser passados para a aplicação que será executada, digitando a linha de argumentos na caixa de texto logo abaixo de "Program arguments:".

Clicando no botão "Run" desta janela, a aplicação será executada os argumentos serão passados, como se fossem em linha de comando. O resultado será:

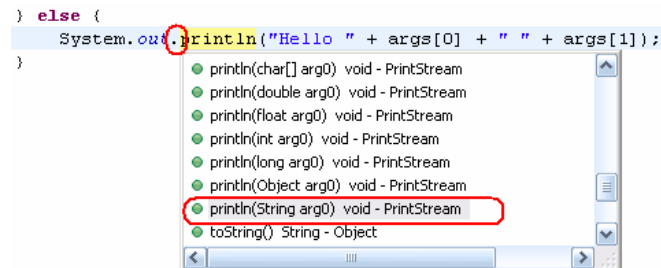


Nota: Caso a aplicação leia dados da entrada padrão (teclado), os dados poderão ser digitados pelo desenvolvedor diretamente na console.

Javadoc no code completion

O code completion do Eclipse é uma funcionalidade bastante útil. Ao pressionar "." (ponto) depois de um nome de classe ou referência para um objeto, uma janela aparece para que seja selecionado um atributo ou método da classe ou objeto referenciado antes do "." (ponto).

Esta janela também pode ser invocada através do atalho "ctrl + barra de espaço", permitindo a seleção até mesmo de um atributo ou método da mesma classe que está sendo codificada.



Selecionando-se um método ou atributo, o Eclipse automaticamente o escreve na posição do cursor.

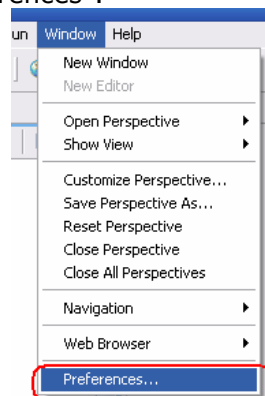
Mas o método que se deve usar para obter uma funcionalidade desejada é uma informação contida no Javadoc.

Se o código fonte da classe está disponível para o Eclipse, uma descrição do que estiver selecionado na janela de completion será mostrada, desde que o código fonte contenha o Javadoc para o que estiver selecionado.

Por padrão, o Eclipse necessita apenas da JRE para ser utilizado, então é a esta que ele se associa na sua primeira execução.

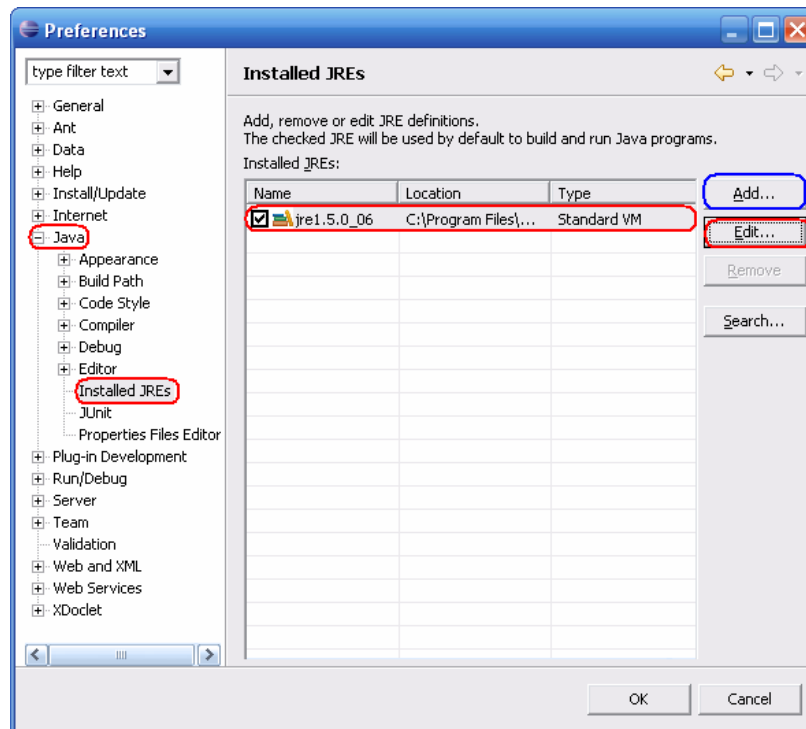
A JRE não traz o código fonte da API do Java e, portanto, o code completion não disponibilizará o Javadoc.

Para associar o Eclipse à JDK (supondo que esta está instalada), deve-se ir ao menu "Window" e selecionar "Preferences".



Na tela de preferências, no nó "Java" a funcionalidade desejada fica em "Installed JREs".

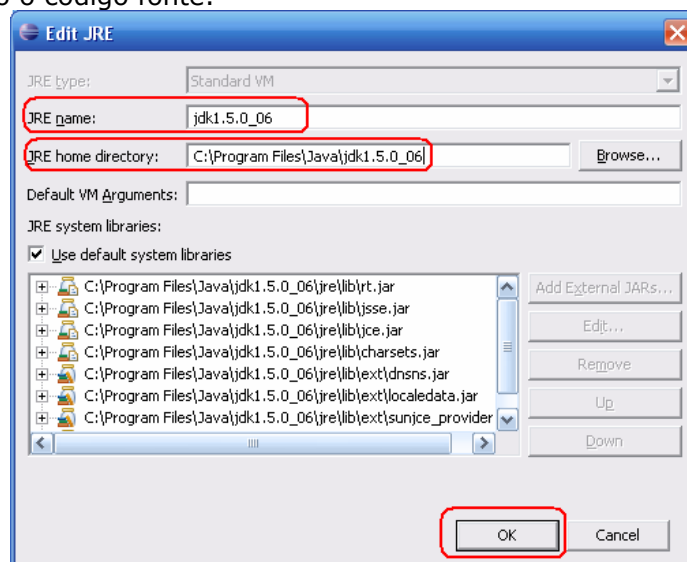
Em seguida, após selecionar a JRE encontrada pelo Eclipse deve-se clicar em "Edit...".



Nota: Em "Add..." pode-se adicionar outras JRE's, de forma que os projetos poderão utilizá-las alternativamente.

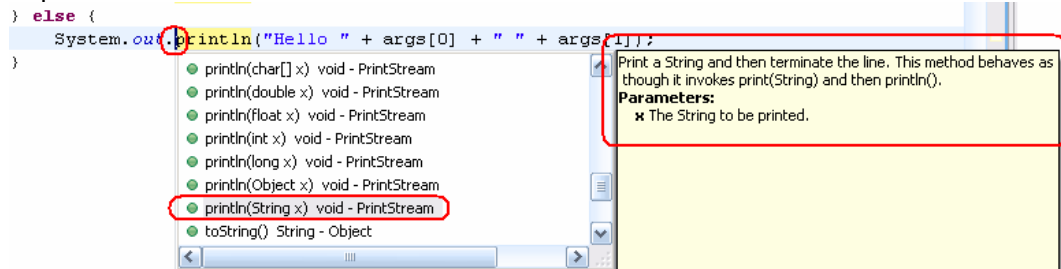
Na janela que segue, é possível alterar o nome da JRE para JDK no campo identificado por "JRE name:".

Alterando o caminho da JRE no campo identificado por "JRE home directory" para o caminho da JDK instalada e clicando em "OK", o Eclipse passará a usar a nova biblioteca, contendo o código fonte.



Após esta modificação e um curto período de espera para que o Eclipse recompile os projetos associados ao compilador "default" do workspace, o Javadoc ficará disponível para estes projetos.

Agora, ao pressionar "ctrl + barra de espaço" ou "." (ponto) e selecionar um método ou atributo, uma explicação este ele estará disponível, ao lado da janela de code completion.

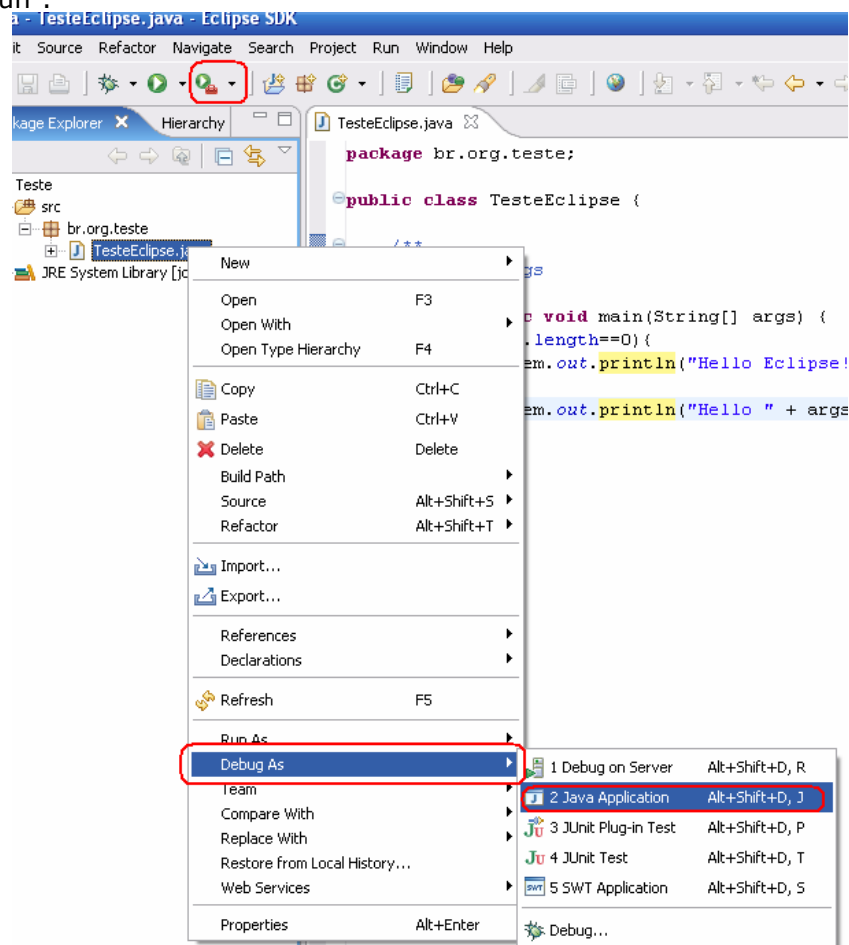


Debug

O Eclipse, como outras IDE's tem um debugger que auxilia o desenvolvedor a encontrar erros que podem acontecer em tempo de execução.

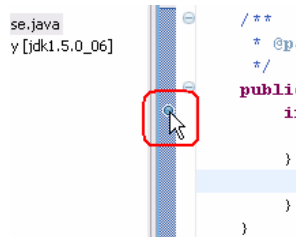
Para executar uma aplicação em modo debug, o processo é exatamente igual ao de executar uma aplicação normalmente, exceto que a opção selecionada deve ser "Debug As" ao invés de "Run As".

Esta opção está acessível em um botão "Debug" ao lado do botão "Run", no menu de contexto da classe e no menu "Run" e o comportamento é exatamente o mesmo de "Run".



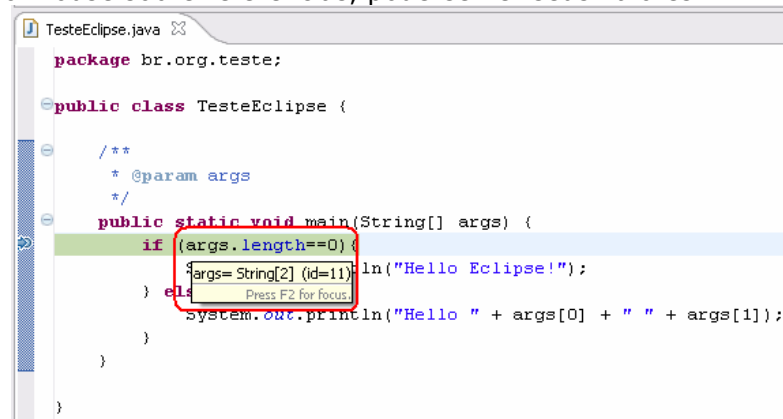
Porém, executar em modo debug apenas não traz nenhuma vantagem se não for definido um ponto de parada da execução ("breakpoint"), a partir do qual a máquina virtual passará a executar cada linha de código ao comando do desenvolvedor, que poderá então analisar o comportamento e o conteúdo das variáveis no código.

Para definir o breakpoint, um clique duplo do mouse deve ser dado na barra lateral à esquerda do código, na linha em que se deseja a parada.

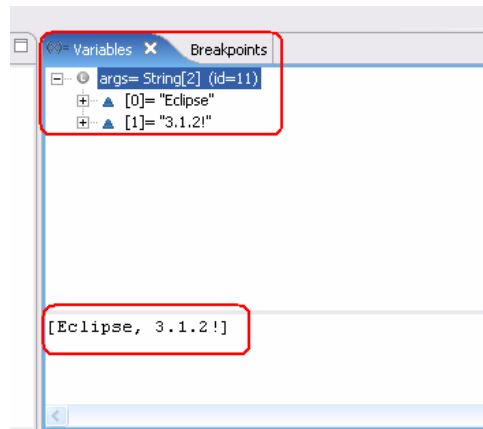


Agora, executando a aplicação em modo debug, na perspectiva de debug pode-se controlar a execução da aplicação.

Parando o mouse sobre referências, pode-se ver seus valores.



Na aba "Variables" pode-se acessar o estado interno de objetos e ver a lista de valores de variáveis, além de uma representação textual do objeto ou variável selecionado.



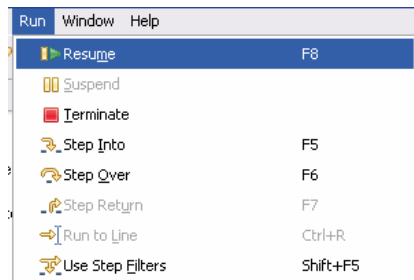
A execução pode prosseguir de formas diferentes:

A primeira, pressionando "F8" a execução segue até o fim, sem interrupções, a não ser que encontre outro breakpoint.

A tecla "F5" faz com que o ponto de execução a ser examinado seja a primeira linha do próximo método a ser avaliado na linha atual da execução. Ou seja, "descemos" ou "entramos" um nível na execução da aplicação, mudando de escopo.

A tecla "F6" passa para a próxima linha da execução da aplicação.

Estas e outras teclas de atalho estão acessíveis no menu "Run".



Nota: Durante a execução normal, um programa pode lançar uma exceção. Se o método "printStackTrace" da exceção for chamado ou se esta não for capturada, o Stack Trace será impresso na console do Eclipse.

```
package br.org.teste;

public class TesteEclipse {

    /**
     * @param args
     */
    public static void main(String[] args) {
        if (args.length==0) {
            System.out.println("Hello Eclipse!");
        } else {
            System.out.println("Hello " + args[0] + " " + args[1]);
            throw new RuntimeException("exemplo");
        }
    }
}
```

Problems | Javadoc | Declaration | Console

<terminated> TesteEclipse [Java Application] C:\Program Files\Java\jdk1.5.0_06\bin\javaw.exe (27/04/2006 20:09:00)
Hello Eclipse 3.1.2!
Exception in thread "main" java.lang.RuntimeException: exemplo
at br.org.teste.TesteEclipse.main(TesteEclipse.java:13)

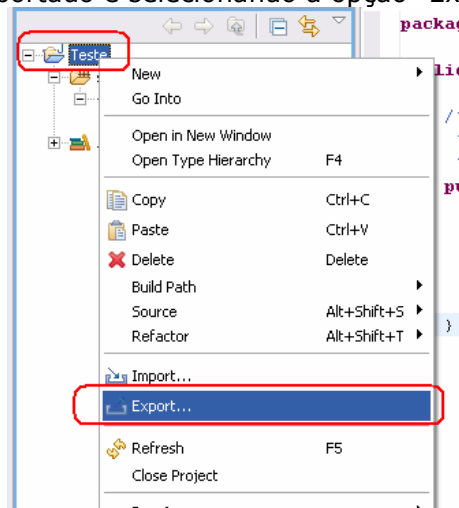
Um clique duplo em uma linha do Stack Trace fará com que o eclipse selecione a linha correspondente no código fonte da classe em questão, porém isto só ocorrerá se o código estiver disponível para o projeto.

Nota: lançar uma exceção Runtime é uma prática de programação que deve ser evitada e está aqui apenas para efeito ilustrativo.

Export

Muitas vezes é necessário exportar um ou mais arquivos do projeto ou o projeto inteiro, para enviar a outra pessoa ou colocar em outro workspace do Eclipse.

Esta funcionalidade pode ser acessada pelo menu de contexto do arquivo, diretório ou projeto a ser exportado e selecionando a opção "Export...".



Na tela seguinte, selecionando "Archive file" e clicando em "Next" pode-se criar um arquivo compactado (zip ou tar) contendo o conteúdo que está sendo exportado.

Nesta tela, há várias possibilidades de tipos de arquivos para exportação:

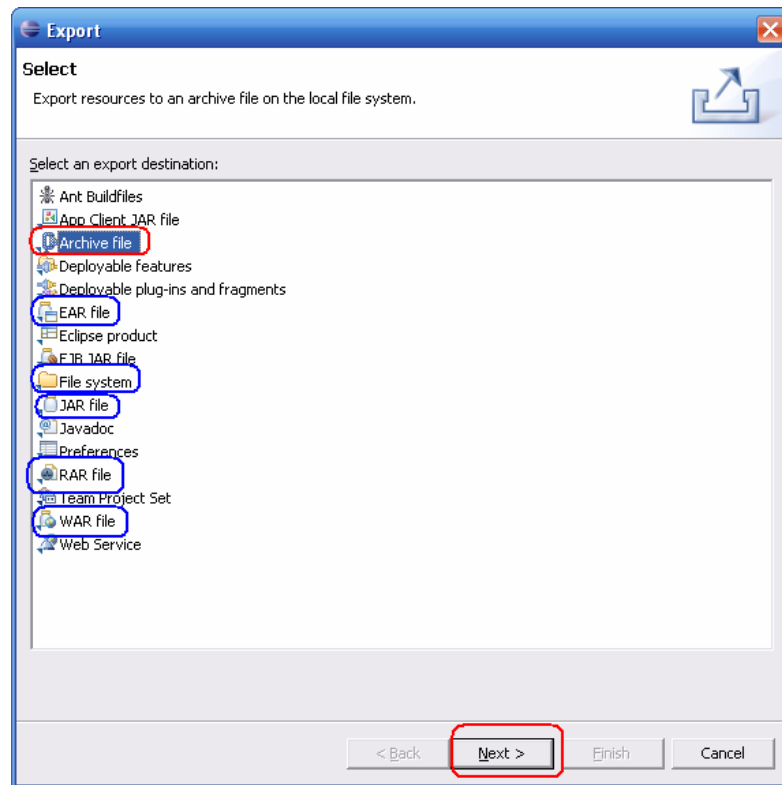
EAR file: O Enterprise Web Archive pode ser exportado através de um wizard, desde que se tenha um Eclipse com WTP integrado.

File system: Apenas uma cópia do diretório, projeto ou arquivo selecionado.

JAR file: Um Java Archive, criado a partir de um wizard.

RAR file: Um arquivo compactado RAR.

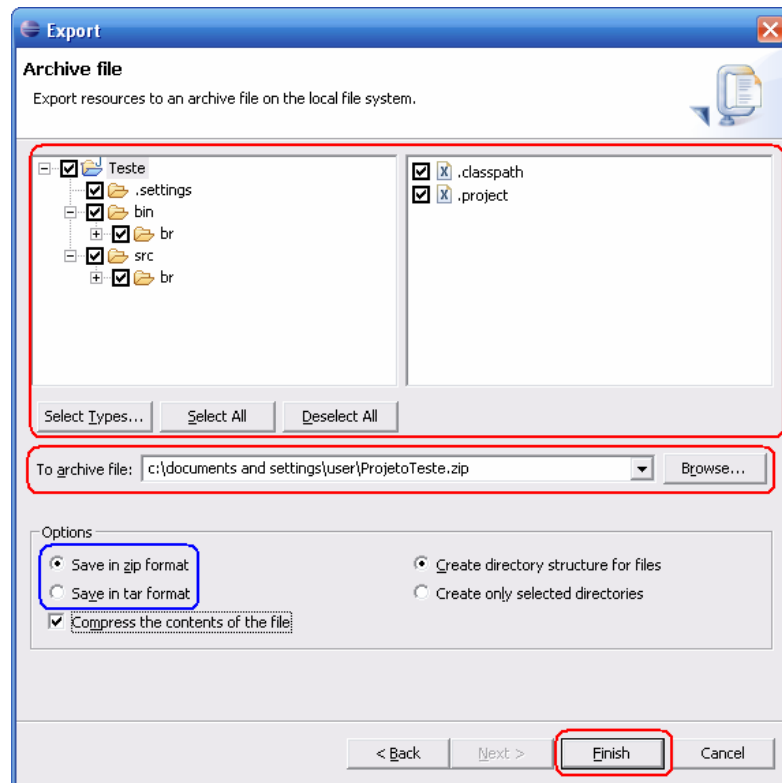
WAR file: O Web Archive pode ser exportado através de um wizard, desde que se tenha um Eclipse com WTP integrado.



Na tela seguinte é possível refinar a seleção, adicionando ou retirando arquivos da estrutura de diretórios a ser exportada.

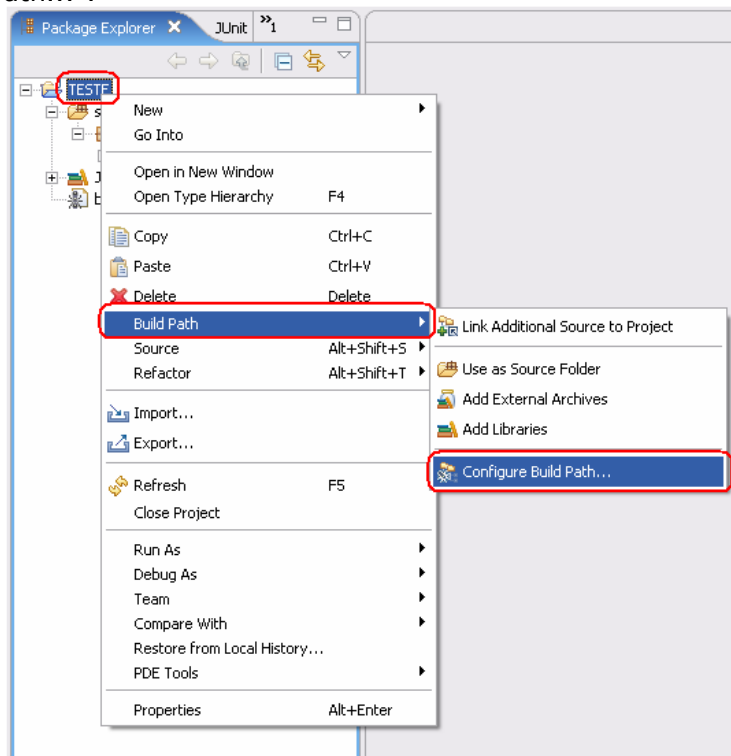
É necessário também preencher o nome do arquivo que será criado e clicar em "Finish" para completar o processo.

Em azul há a possibilidade de escolher o formato do arquivo de exportação.



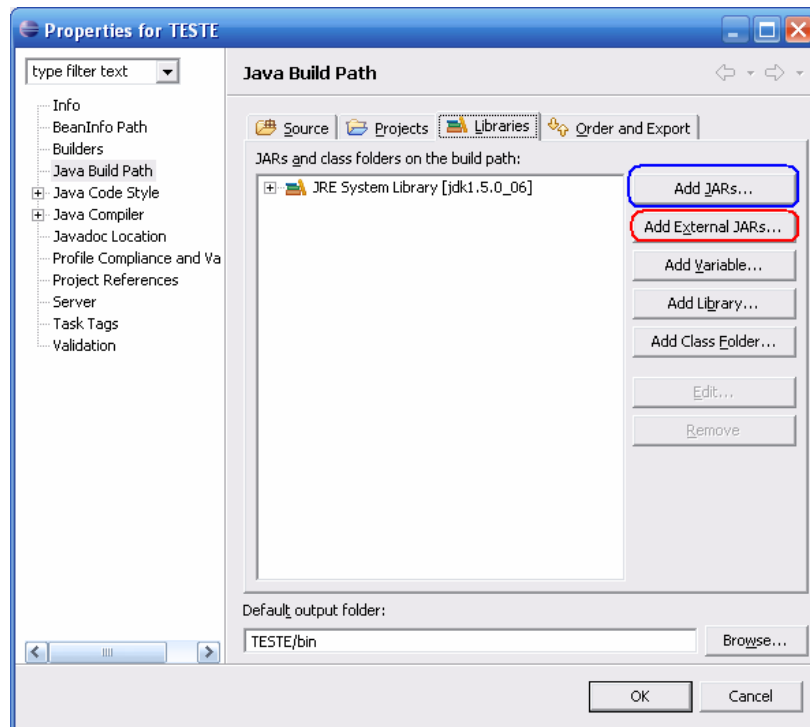
Importando biblioteca

No menu de contexto do projeto é possível adicionar uma biblioteca para utilizá-la no desenvolvimento da aplicação, selecionando "Build Path" e em seguida "Configure Build Path..."



O botão "Add External JARs..." permite que se escolha uma biblioteca JAR em qualquer diretório.

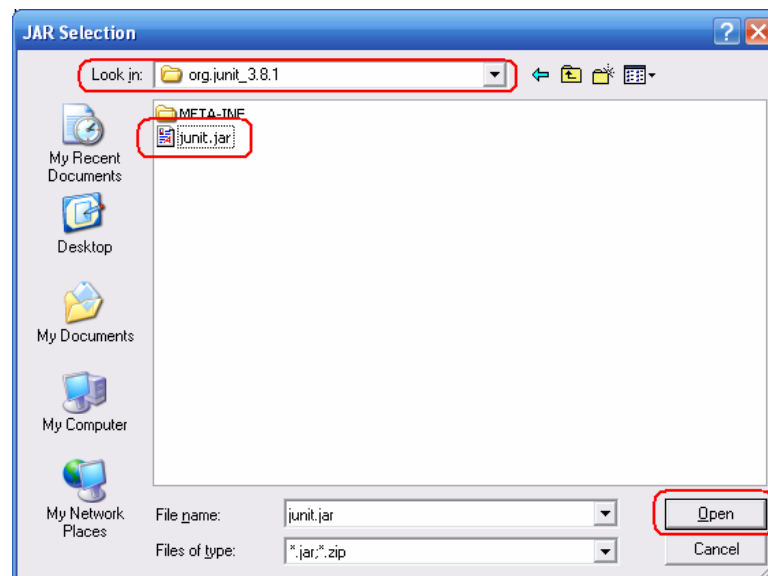
Caso a biblioteca esteja dentro da estrutura de diretórios do projeto, também é possível escolher a opção "Add JARs..."



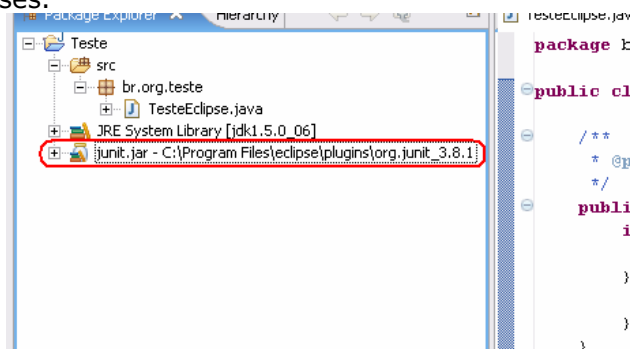
Nota: Ao adicionar uma biblioteca externa, o caminho guardado no projeto é o caminho absoluto para o arquivo JAR. Desta forma, caso o projeto seja copiado para outro computador, pode deixar de funcionar caso a biblioteca não esteja disponível exatamente no mesmo lugar. Assim, é sempre recomendável adicionar uma biblioteca que esteja presente dentro da estrutura de diretórios do projeto, pois desta forma o caminho armazenado ficará relativo à raiz do projeto e este se manterá funcionando ainda que o projeto esteja sendo executado em outro computador.

Na tela seguinte, deve ser selecionada a biblioteca que se deseja instalar. A título de exemplo, está sendo selecionada a biblioteca "JUnit", presente no diretório "org.junit_3.8.1" dentro do diretório "plugins" no diretório de instalação do Eclipse.

Por não estar dentro do projeto, o botão "Add external JARs..." é o que permitirá a adição desta biblioteca.



Clicando em “Open” a biblioteca será adicionada ao projeto, podendo ser utilizada por suas classes.



Criando e executando teste JUnit

Para executar o JUnit, primeiro precisa-se de uma classe com um método a ser testado. À classe "TesteEclipse" foi adicionado o seguinte método:

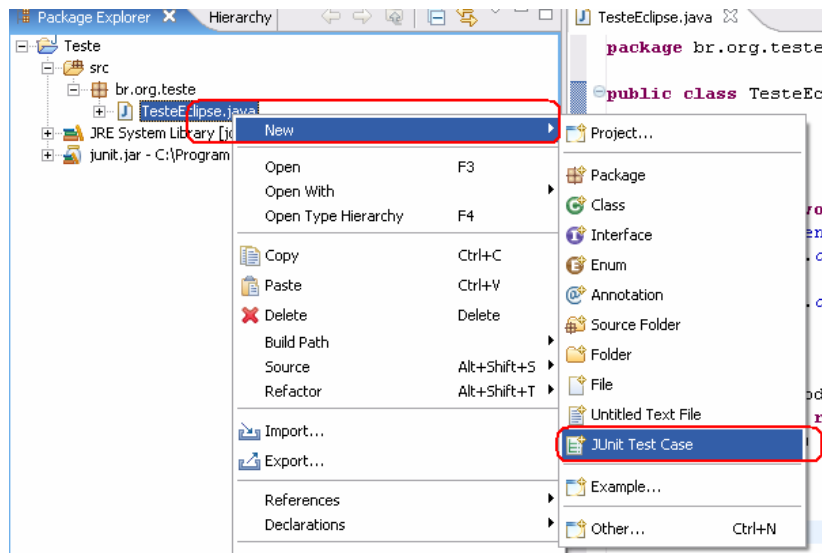
```
public int metodoParaTestar(int x) {  
    if (x < 0) return 1;  
    if (x == 0) throw new RuntimeException();  
    return 0;  
}
```

A classe então ficou:

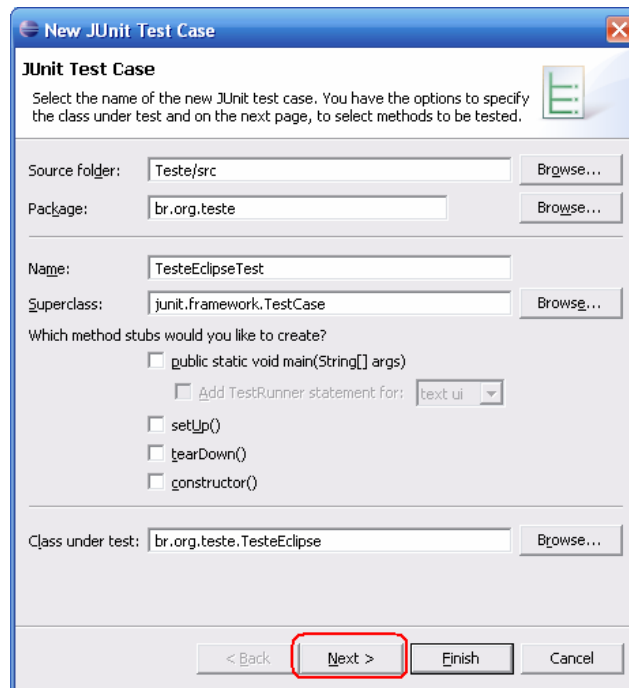
```
package br.org.teste;  
  
public class TesteEclipse {  
  
    /**  
     * @param args  
     */  
    public static void main(String[] args) {  
        if (args.length==0){  
            System.out.println("Hello Eclipse!");  
        } else {  
            System.out.println("Hello " + args[0] + " " + args[1]);  
        }  
    }  
  
    public int metodoParaTestar(int x) {  
        if (x < 0) return 1;  
        if (x == 0) throw new RuntimeException();  
        return 0;  
    }  
}
```

Nota: lançar uma exceção Runtime é uma prática de programação que deve ser evitada e está aqui apenas para efeito ilustrativo.

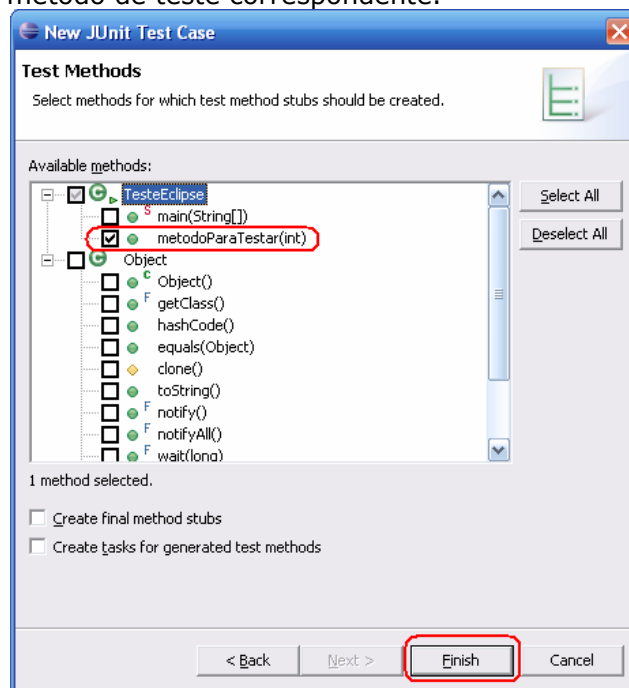
No menu de contexto dela, pode-se escolher "JUnit test case" no menu "New".



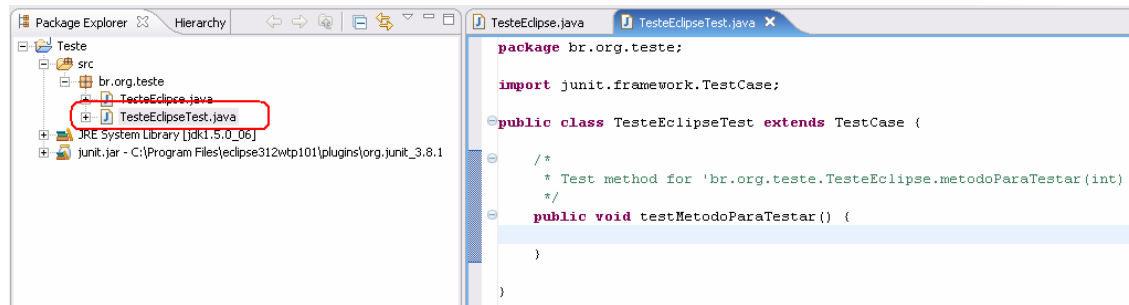
Na tela seguinte há várias opções para a classe de teste que será gerada.



Clicando em "Next >" pode-se selecionar o método a ser testado de forma que o Eclipse criará o método de teste correspondente.



Selecionando o método e clicando em Finish, a classe de teste terá sido criada.



No corpo do método "testMetodoParaTestar()" pode-se inserir um código de teste.

Para fins de exemplo, será inserido o seguinte código:

```
TesteEclipse testeEclipse = new TesteEclipse();
assertEquals("Este teste passa", testeEclipse.metodoParaTestar(-1), 1);
assertEquals("Este teste dá failure", testeEclipse.metodoParaTestar(2), 5);
assertEquals("Este teste dá error", testeEclipse.metodoParaTestar(0), 5);
```

E a classe ficará assim:

```
package br.org.teste;

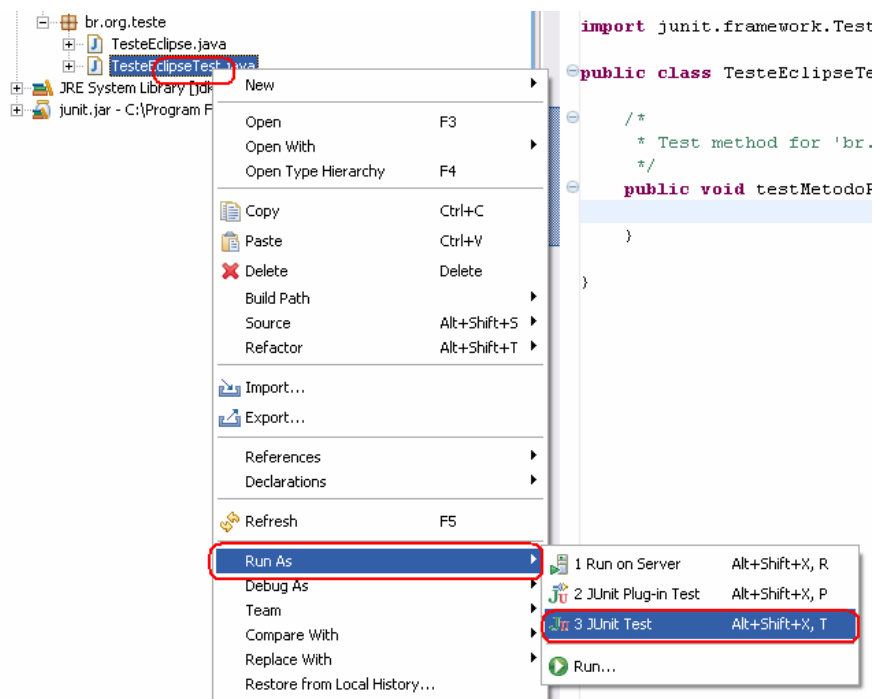
import junit.framework.TestCase;

public class TesteEclipseTest extends TestCase {

    /*
     * Test method for 'br.org.teste.TesteEclipse.metodoParaTestar(int)'
     */
    public void testMetodoParaTestar() {
        TesteEclipse testeEclipse = new TesteEclipse();
        assertEquals("Este teste passa", testeEclipse.metodoParaTestar(-1), 1);
        assertEquals("Este teste dá failure", testeEclipse.metodoParaTestar(2), 5);
        assertEquals("Este teste dá error", testeEclipse.metodoParaTestar(0), 5);
    }
}
```

Este teste passará no primeiro assert, no segundo dará uma failure e no terceiro um error.

No menu de contexto da classe de teste, da mesma forma que se executa uma aplicação, o teste pode ser executado.



O resultado será mostrado na aba JUnit.

De cima para baixo e da esquerda para a direita, o primeiro botão circulado executa novamente o último teste e o da direita executa novamente o último teste executando primeiro os que tiveram resultado "failure".

Em "Runs" é apresentado o número de métodos de teste executados e o total de métodos de teste.

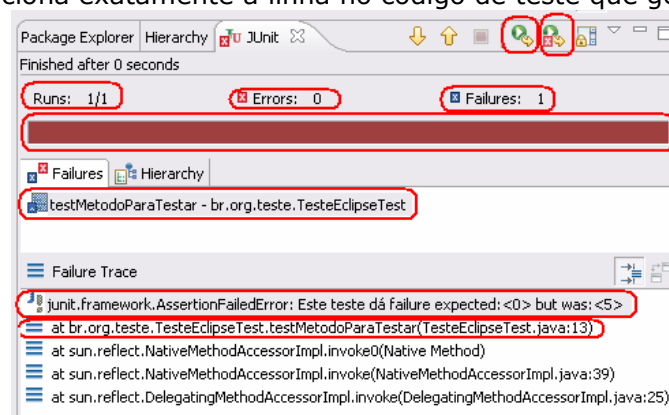
Em "Errors" a quantidade de erros obtidos e em "Failures" a quantidade de falhas obtidas.

A barra vermelha é uma barra de progresso e ao mesmo tempo de sucesso, pois se mantém verde enquanto nenhum teste der erro ou falhar.

Em "Failures" estão relatadas as falhas. Um clique duplo seleciona a classe e o método de teste que gerou a falha.

Em "Failure Trace" há informações detalhadas sobre a falha selecionada em "Failures" (acima). A primeira linha retorna informações a respeito da falha.

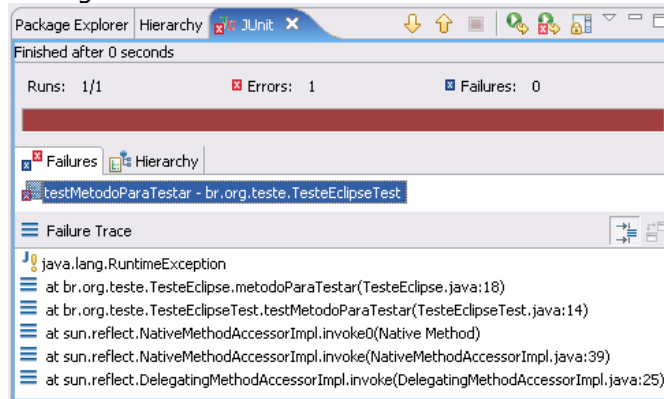
Na segunda linha é mostrado o ponto em que ocorreu a falha e um clique duplo sobre ela seleciona exatamente a linha no código de teste que gerou a falha.



Dando um clique duplo nesta linha e comentando esta linha de código, pode-se executar novamente o teste e o resultado não mais relatará esta falha.

Nota: Obviamente esta não é uma solução. Se o teste falha, o código deve ser corrigido. No entanto o objetivo deste documento não é explicar como se trata erros e testes. Comentar esta linha não resolve o problema, mas permite que o restante desta funcionalidade do Eclipse possa ser mostrado mais rapidamente.

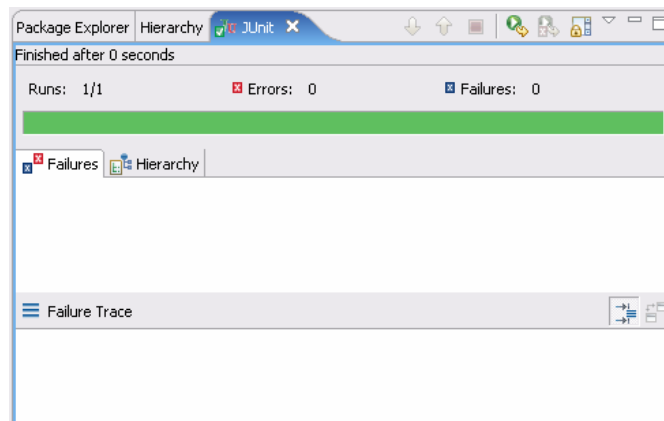
O novo resultado agora será o relato de um erro.



Um erro é o resultado de um erro de programação e não um teste que falhou. Sendo assim, uma exceção é relatada.

No Eclipse, um clique duplo em uma determinada linha da pilha de execução trazida pela exceção leva à linha que a originou, no código fonte de uma determinada classe (caso esteja disponível).

Sendo assim, um clique duplo sobre a terceira linha em "Failure Trace" leva à linha de código que, quando comentada, fará o teste passar com sucesso. Quando é executado com sucesso, a barra de progresso é apresentada na cor verde, desta forma:



Nota: Obviamente esta não é uma solução. Se o teste dá um erro, o código deve ser corrigido. No entanto o objetivo deste documento não é explicar como se trata erros e testes. Comentar esta linha não resolve o problema, mas permite que o restante desta funcionalidade do eclipse possa ser mostrado mais rapidamente.

Nota: Informações sobre o JUnit, incluindo tutorial para seu aprendizado pode ser obtido em <http://www.junit.org/>.

Executando um Ant build

Um Ant build é um arquivo XML que pode ser executado através do framework de automação Ant, desenvolvido e mantido pela Fundação Apache.

O arquivo deve se chamar build.xml para que o Eclipse o reconheça como um script de Ant e forneça toda a funcionalidade de code completion e validação para este XML.

Para este exemplo, um script ant foi escrito:

```
<?xml version="1.0"?>
<project name="Teste" default="Nada" basedir=".">
  <target name="Nada" description="Este será o target padrão, mas
nada será feito nele além de imprimir uma mensagem.">
    <echo message="Mensagem impressa ao executar o target
Nada (padrão)."/>
  </target>

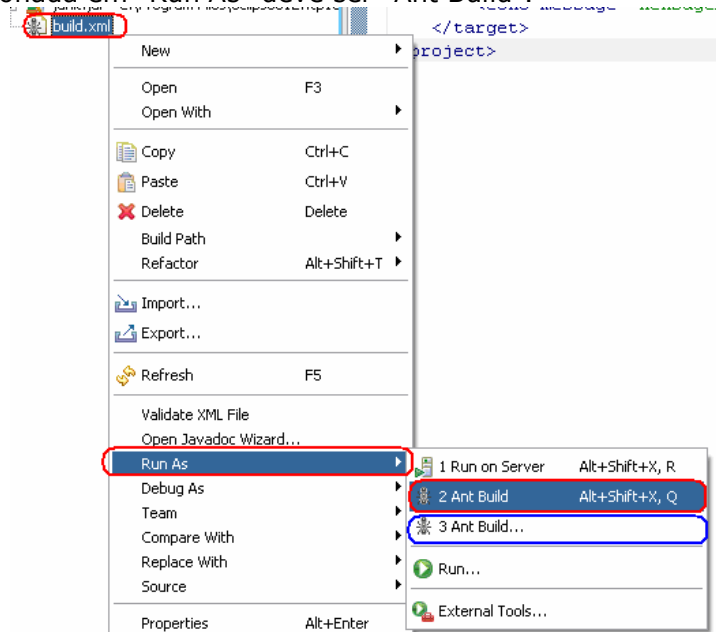
  <target name="Nada2" description="Este NÃO será o target padrão,
mas também não fará nada além de imprimir uma mensagem.">
    <echo message="Mensagem impressa ao executar o target
Nada2."/>
  </target>
</project>
```

Ficando assim:

```
<?xml version="1.0"?>
<project name="Teste" default="Nada" basedir=".">
  <target name="Nada" description="Este será o target padrão, mas nada será feito nele além de imprimir uma mensagem.">
    <echo message="Mensagem impressa ao executar o target Nada (padrão)."/>
  </target>

  <target name="Nada2" description="Este NÃO será o target padrão, mas também não fará nada além de imprimir uma mensagem.">
    <echo message="Mensagem impressa ao executar o target Nada2."/>
  </target>
</project>
```

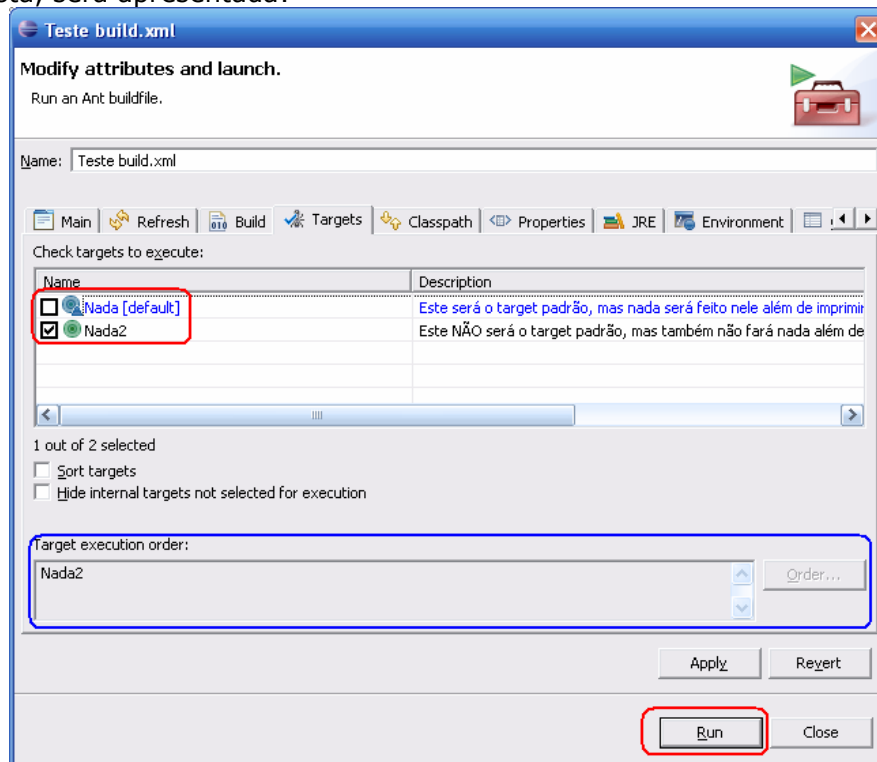
O procedimento para execução é o mesmo para uma aplicação Java, exceto que a opção selecionada em "Run As" deve ser "Ant Build".



E o resultado será mostrado na console do Eclipse:

```
Problems Javadoc Declaration Console
<terminated> Teste build.xml [Ant Build] C:\Program Files\Java\jdk1.5.0_06\bin\javaw.exe (27/04/2006 20:33:47)
Buildfile: C:\Documents and Settings\User\workspace\Teste\build.xml
Nada:
[echo] Mensagem impressa ao executar o target Nada (padrão).
BUILD SUCCESSFUL
Total time: 1 second
```

Caso se deseje selecionar um target diferente do padrão descrito no script Ant, a opção abaixo, “Ant Build...” (marcada em azul), deve ser selecionada e uma tela como esta, será apresentada:



Nesta tela podem ser selecionados os targets que se quiser executar (em “Check targets to execute:”) e a ordem de execução (em “Target execution order:”). Clicando em “Run”, a seleção de targets será executada na ordem estabelecida.

Nota: Nesta tela, outras opções de configuração para execução deste script ant podem ser configuradas, como variáveis de ambiente, classpath e outras.

O resultado desta execução será:

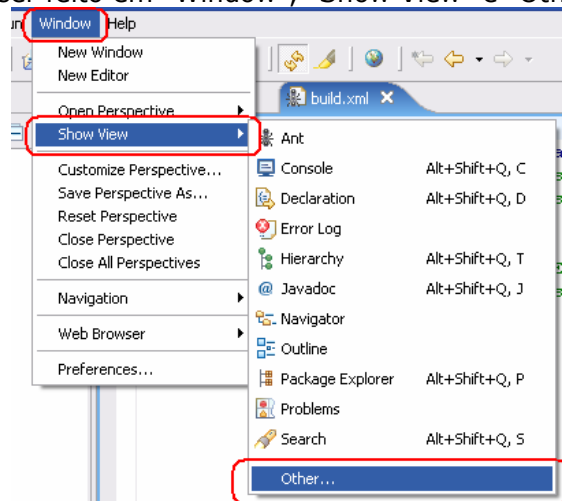
```
Problems Javadoc Declaration Console
<terminated> Teste build.xml [Ant Build] C:\Program Files\Java\jdk1.5.0_06\bin\javaw.exe (27/04/2006 20:48:15)
Buildfile: C:\Documents and Settings\User\workspace\Teste\build.xml
Nada2:
[echo] Mensagem impressa ao executar o target Nada2.
BUILD SUCCESSFUL
Total time: 157 milliseconds
```

Nota: Mais informações sobre o Ant como manual de utilização, atualizações e API no site do Ant: <http://ant.apache.org/>.

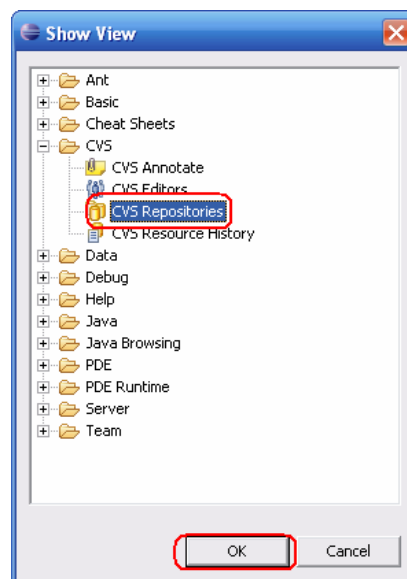
Utilizando o CVS

O plugin para utilização do CVS no Eclipse fornece toda a funcionalidade de um cliente de CVS convencional agregado às vantagens de estar integrado a IDE de desenvolvimento.

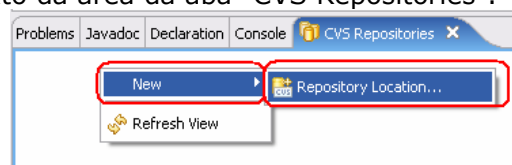
Para começar a utilização do CVS, primeiro deve ser configurado um repositório, o que pode ser feito em "Window", "Show View" e "Other...".



A opção a ser selecionada na janela que será aberta é "CVS Repositories" e em seguida, "OK".



Para criar um repositório, a opção a ser acessada é "Repository Location" em "New" no menu de contexto da área da aba "CVS Repositories".



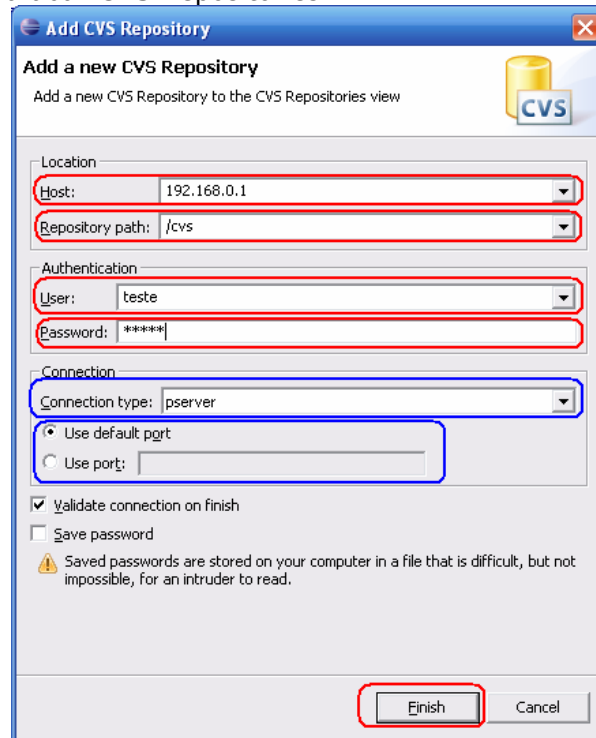
Nesta opção, uma tela para configurar um repositório aparecerá.

Nela, os campos "Host", "Repository path", "User" e "Password" devem ser preenchidos.

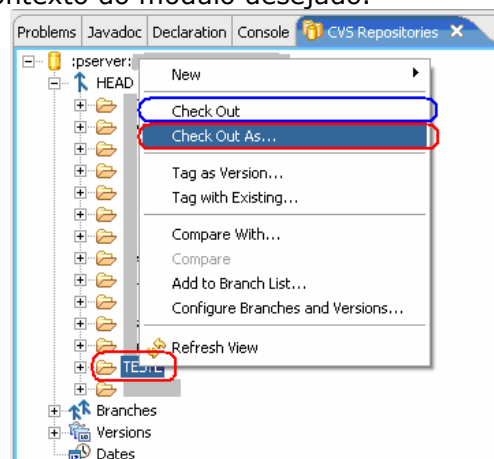
Também pode ser selecionado o protocolo utilizado e a porta do servidor.

Em "Finish", o Eclipse tentará conectar no repositório usando o nome e a senha fornecidos, para "validar" as informações.

Se estiver tudo correto e o servidor funcionando, logo após a validação, será criado um repositório na aba "CVS Repositories".



Para fazer o check out de um módulo, a opção "Check Out As..." deve ser selecionada no menu de contexto do módulo desejado.

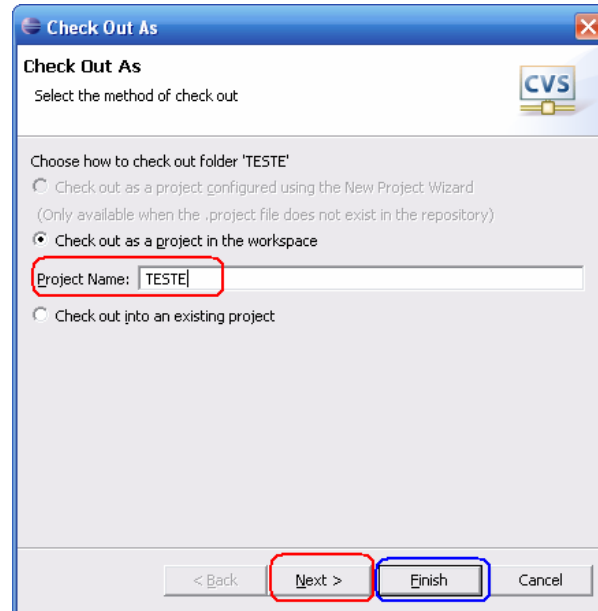


Nota: A opção "Check Out" automaticamente cria um projeto com o nome do módulo.

Em "Project Name" pode ser escolhido o nome para um projeto que será criado a partir do módulo do repositório.

Clicando em "Finish", o projeto será criado no workspace padrão com o conteúdo da última versão do módulo.

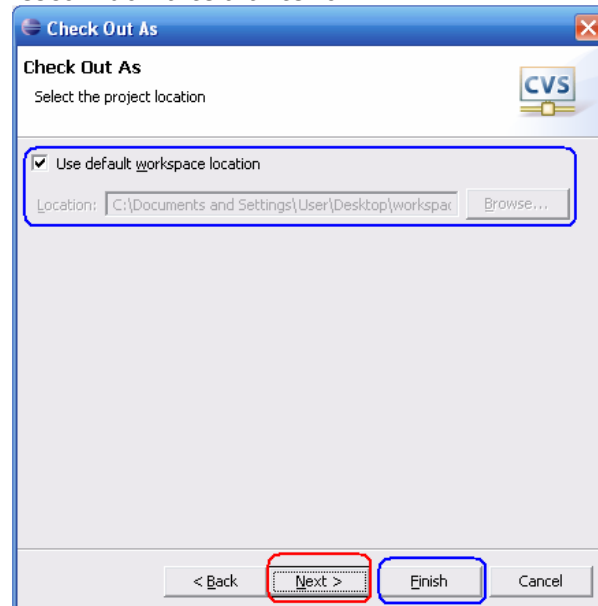
Em "Next" há a possibilidade de selecionar outro workspace para o projeto que será criado.



Na tela seguinte, desmarcando a opção "Use default workspace location" pode ser selecionado outro workspace onde o projeto será criado.

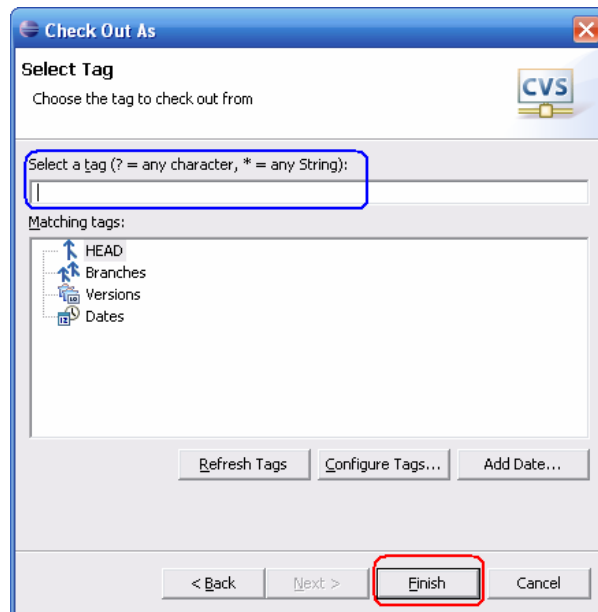
Em "Finish" o projeto será criado no workspace selecionado com a última versão do módulo.

"Next" permitirá selecionar uma determinada tag ou versão do módulo a partir da qual o projeto deverá ser criado com o nome escolhido na primeira tela do deste wizard e no lugar escolhido na tela anterior.



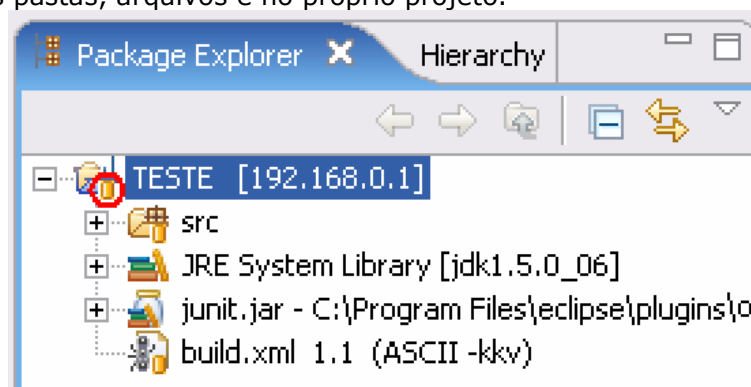
Na última tela, pode ser selecionada a tag, versão e/ou branch preenchendo os campo "Select a tag" e/ou navegando nas opções do campo "Matching tags:".

Clicando em "Finish" o projeto será criado.



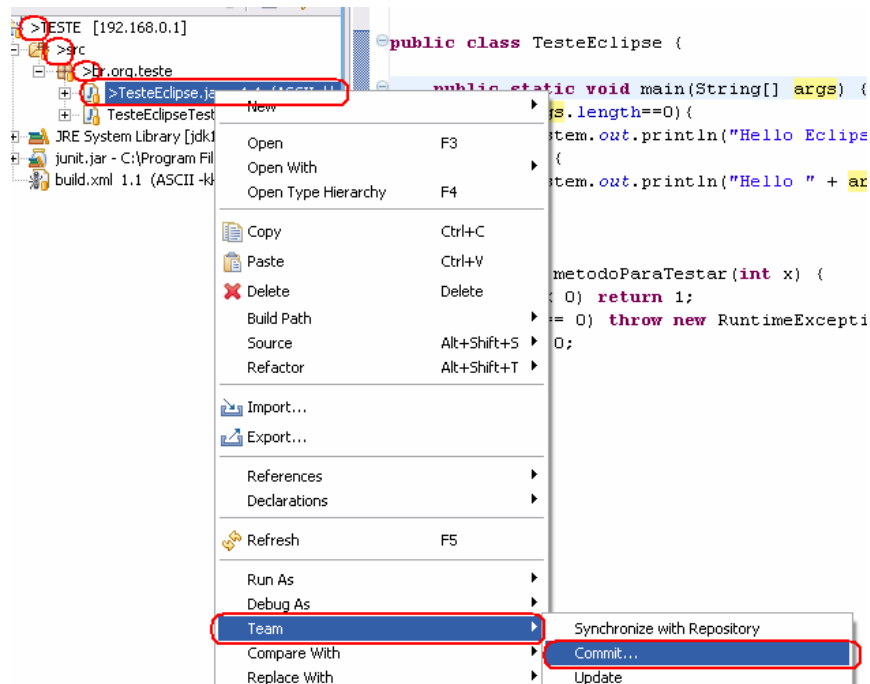
O projeto compartilhado através do plugin de CVS do Eclipse tem a vantagem de estar dentro da IDE e portanto a IDE pode fornecer informações que um cliente de CVS não poderia.

Quando o projeto está compartilhado, o símbolo marcado em vermelho aparece nas suas pastas, arquivos e no próprio projeto.

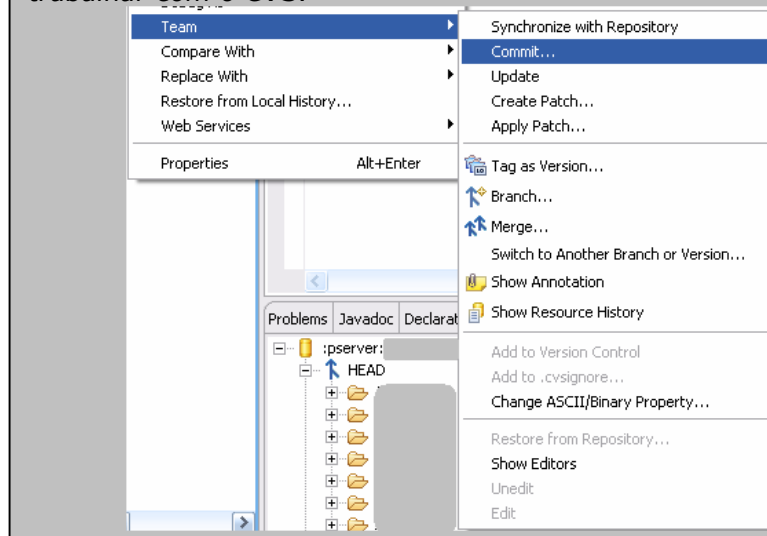


Ao se modificar algum arquivo no projeto, o Eclipse automaticamente marca este arquivo com um sinal de maior (">") antes do nome. Como uma modificação no arquivo é também uma modificação no diretório onde ele está, o Eclipse coloca o mesmo sinal identificador de modificação antes do nome do diretório e assim faz com toda a hierarquia acima, até o projeto. Desta forma, facilmente se encontra as modificações em um projeto.

Para fazer um commit de uma modificação, no menu de contexto do elemento a ser enviado para o repositório (no caso deste exemplo, um arquivo), existe a opção "Team" (onde ficam as opções do CVS) e "Commit...".

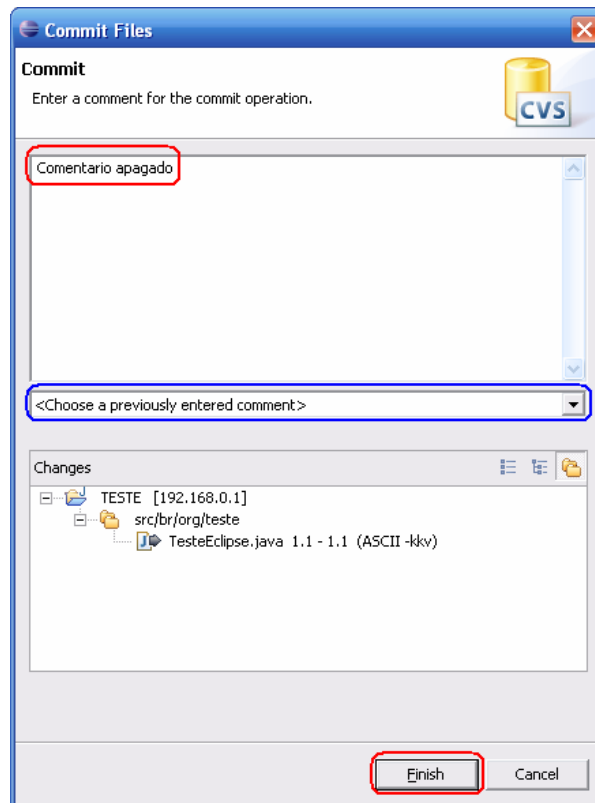


Nota: No menu "Team" são encontradas várias opções para trabalhar com o CVS.



Na tela de commit um comentário relacionado com a mudança que está sendo introduzida no repositório deve ser digitado ou um comentário que já tenha sido escrito anteriormente deve ser selecionado.

"Finish" efetuará o commit.



Para o update, a opção no menu "Team" do menu de contexto do elemento a ser atualizado é "Update".

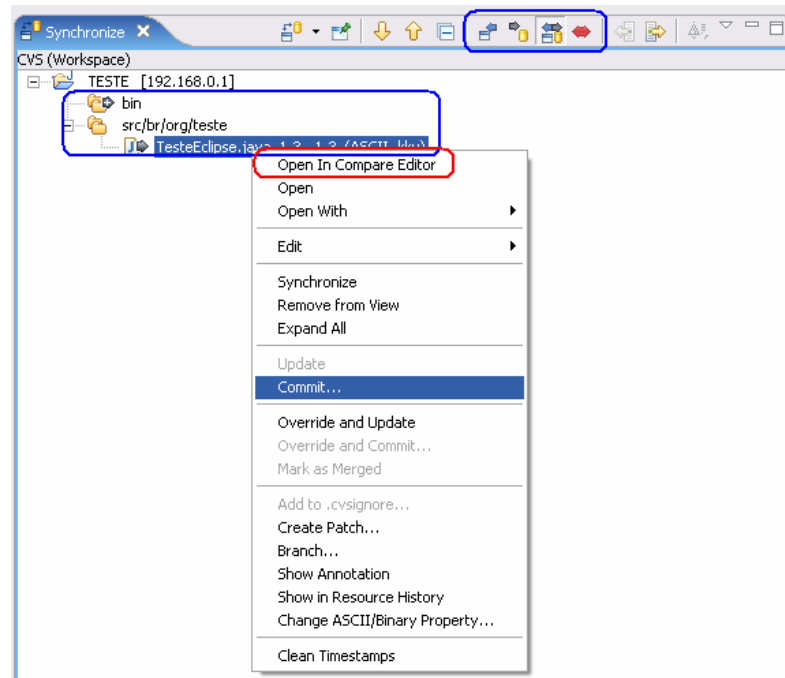


Selecionando esta opção o update será efetuado.

O Eclipse também fornece uma opção para sincronizar um projeto, diretório etc. com o repositório, na opção "Synchronize with Repository" no menu "Team".

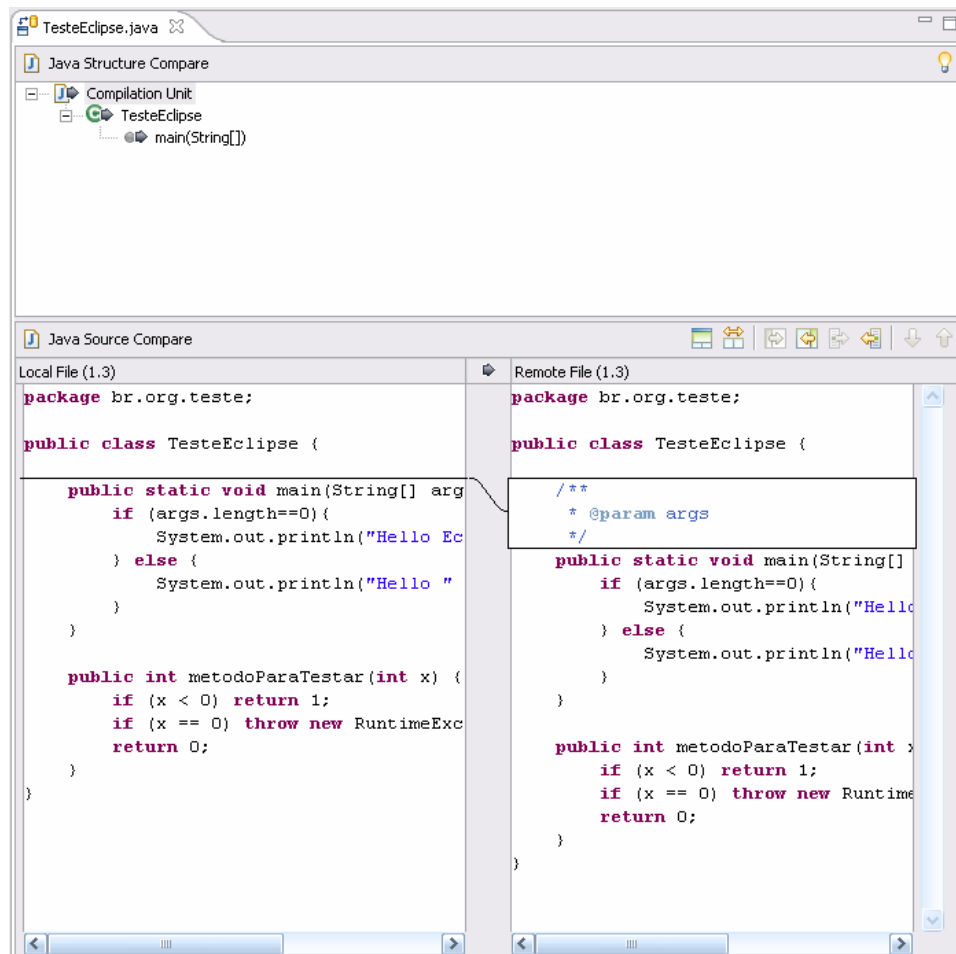


A perspectiva de comparação do Eclipse mostrará então uma árvore com as diferenças entre o que foi selecionado e seu equivalente no repositório. No caso deste exemplo, todo o projeto está sendo sincronizado (clitando com o botão direito do mouse no projeto e selecionando "Team" e "Synchronize with Repository").



No topo da imagem, os quatro botões marcados com azul são filtros, que da esquerda para a direita mostram apenas o que existe de diferente no repositório, apenas o que existe de diferente no projeto, o que existe de diferente nos dois e modificações conflitantes.

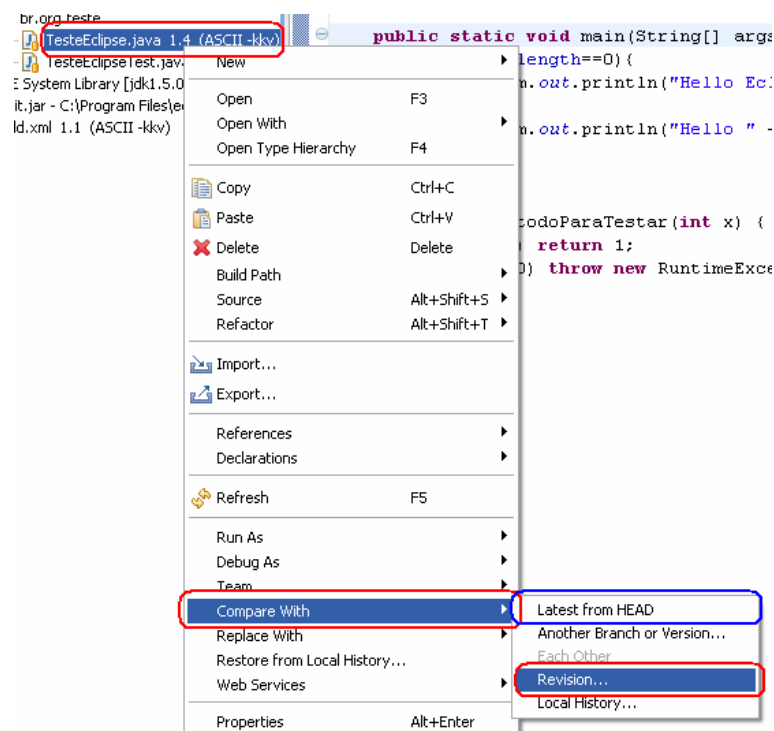
A opção "Open In Compare Editor" no menu de contexto, abre uma tela de comparação.



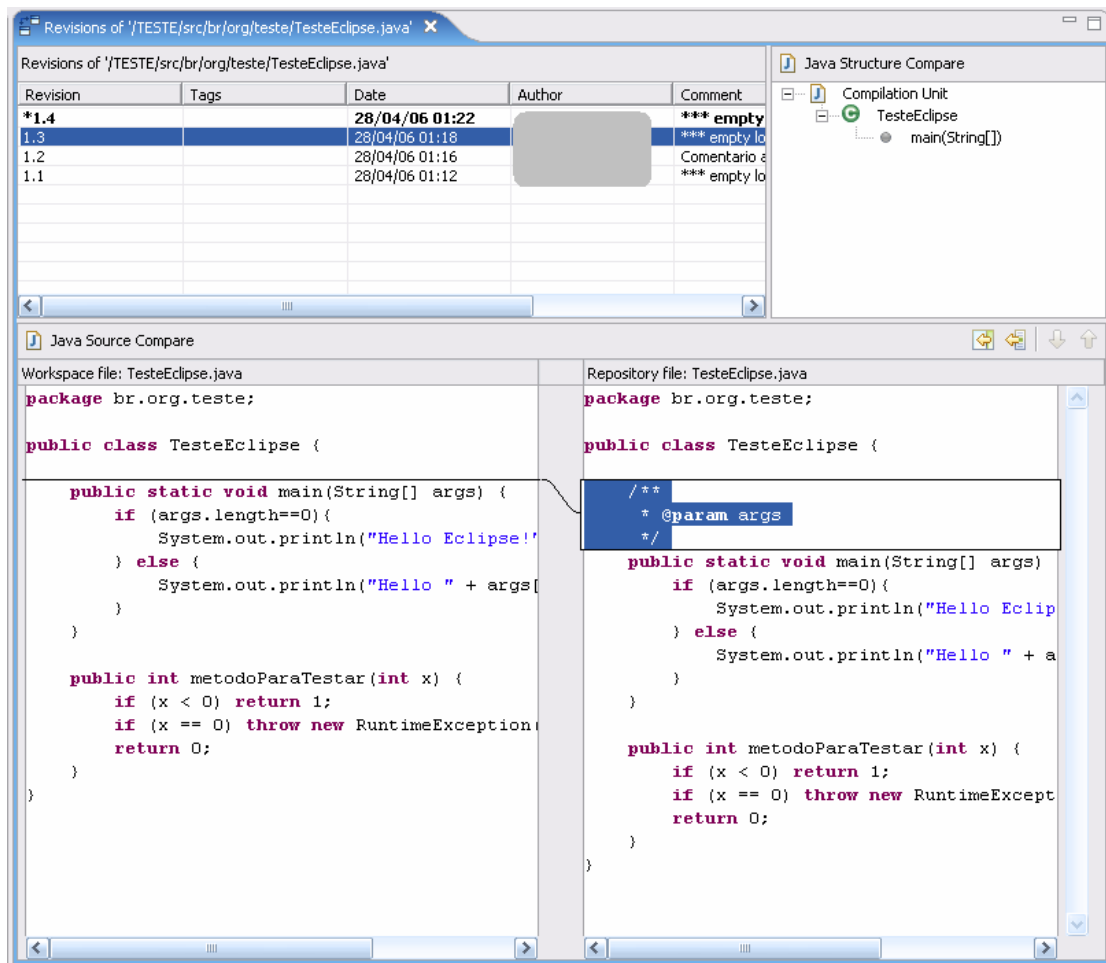
Nesta tela, as alterações podem ser copiadas de uma versão para outra das classes, pacotes, diretórios, etc.

Comparar diferentes versões é uma tarefa bastante comum e não é necessário fazer o sincronismo para obter esta funcionalidade.

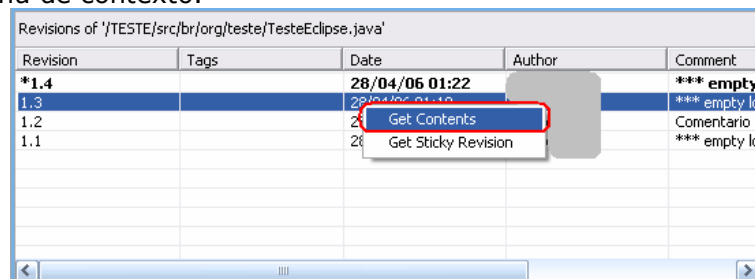
No menu de contexto, na opção "Compare" também está disponível a comparação através das opções "Latest from Head" que compara com a última versão no módulo ou "Revision..." que permite seleccionar a versão com a qual se quer fazer a comparação.



Em "Revision..." uma tela com as informações de cada revisão do que foi selecionado é apresentada e um duplo clique sobre uma determinada revisão faz a comparação da revisão com a versão atual no projeto.



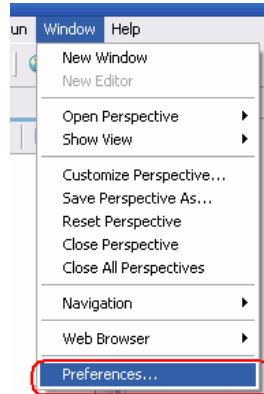
Também é possível pegar o conteúdo de uma versão anterior do que foi selecionado clicando na versão desejada com o botão direito e selecionando "Get Contents" no menu de contexto.



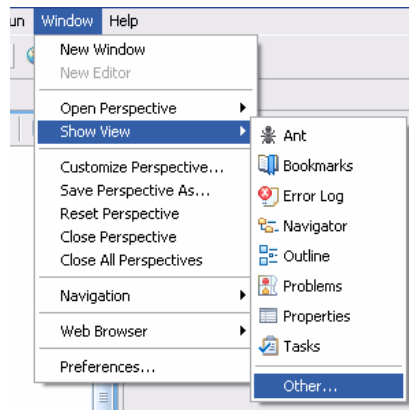
Nota: Há na internet plugins para integrar o Eclipse com outros servidores de controle de versão além do CVS, como o Visual Source Safe (VSS) da Microsoft, o Subversion (SVN), o StarTeam da Borland, etc.

Dicas de configurações personalizadas e outras funcionalidades

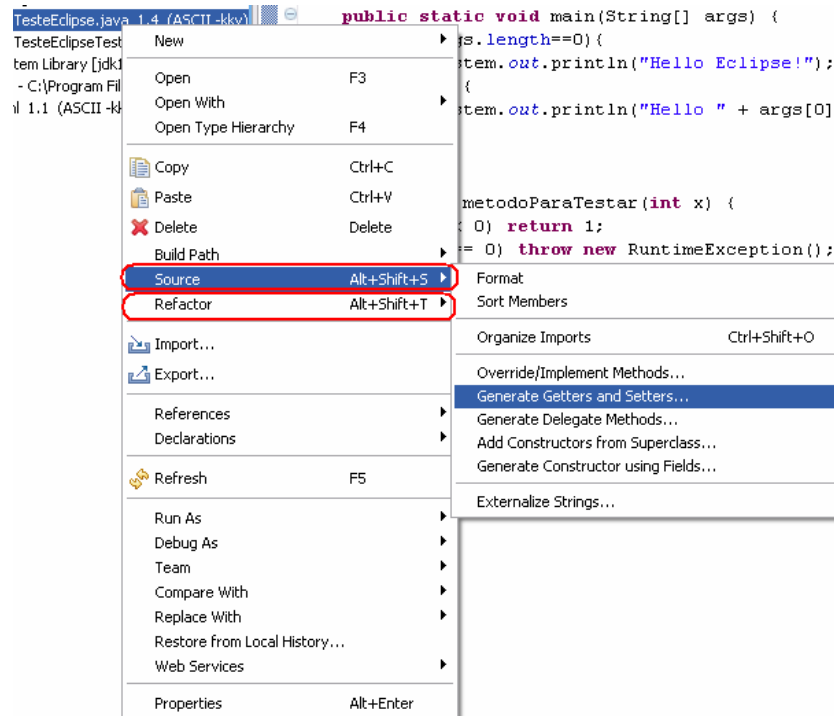
Para configurações personalizadas no Eclipse, deve ser utilizada a janela "Preferences", do menu "Window".



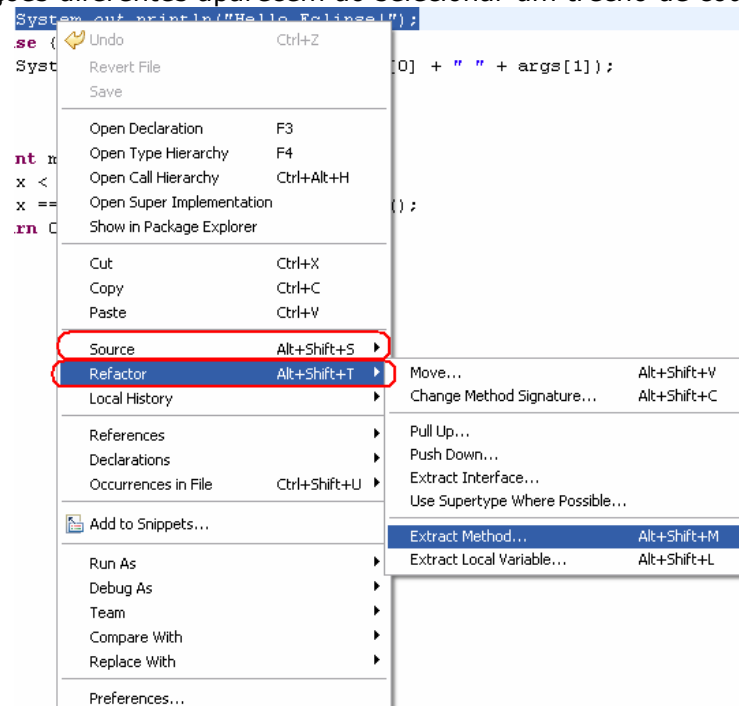
Outras funcionalidades também podem ser encontradas no menu "Show View", também em "Window".



Para automatizar tarefas como renomear classes mantendo a integridade do sistema, renomear métodos, atributos e variáveis, criar automaticamente métodos acessores para atributos ou criar um método a partir de um trecho de código selecionado, vale explorar os menus "source" e "refactor" no menu de contexto das classes, projetos e pacotes no "Package Explorer".

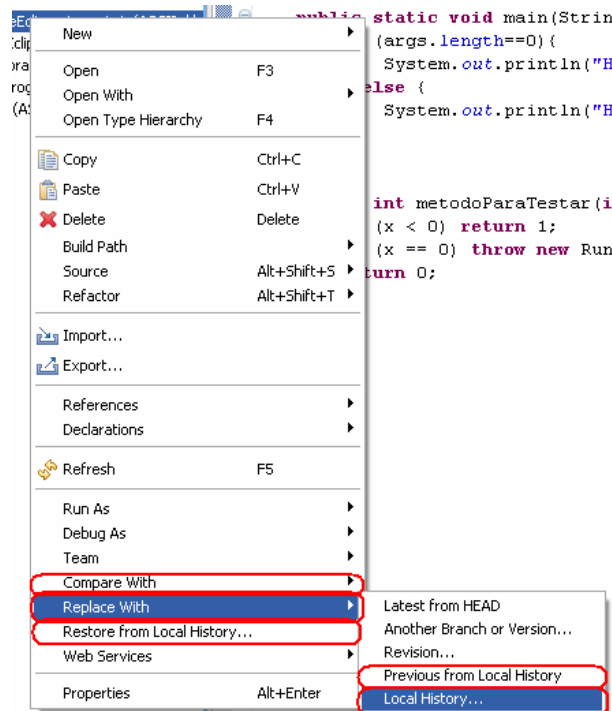


Estes menus também estão disponíveis no menu de contexto do código fonte das classes e opções diferentes aparecem ao selecionar um trecho de código.

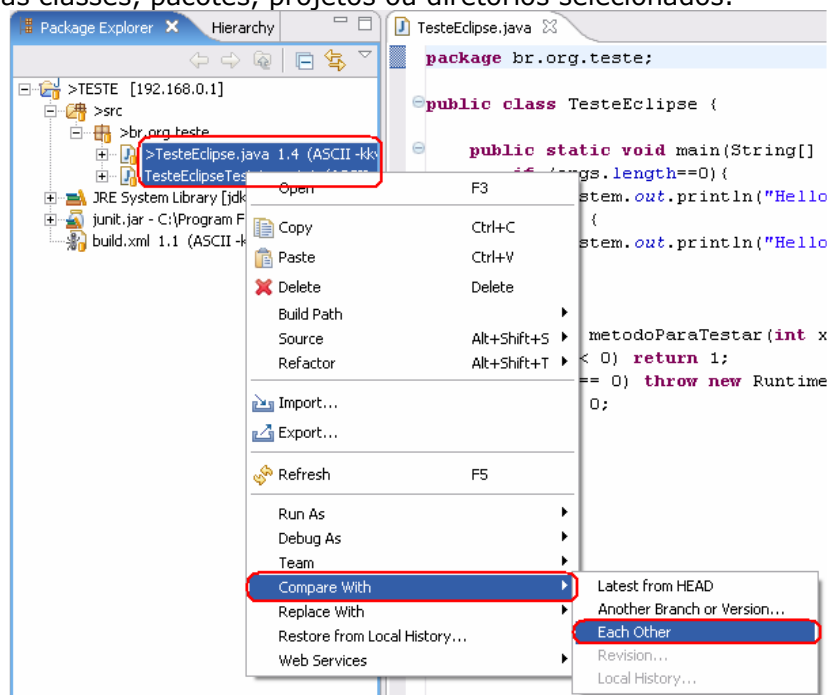


O Eclipse mantém também histórico das classes que estão sendo modificadas, criando uma entrada no histórico para cada salvamento feito pelo desenvolvedor.

Este histórico pode ser acessado através das opções "Compare With", "Replace With" e "Restore from Local History", presentes nos menus de contextos de diretórios, classes, pacotes, métodos e atributos.



O "Compare With" também traz a opção "Each Other" quando no package explorer há duas classes, pacotes, projetos ou diretórios selecionados.



Para selecionar os dois elementos a serem comparados (que podem até ser de projetos diferentes), basta clicar na segunda segurando a tecla "Ctrl".