

Instituto Federal do Paraná – IFPR  
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas  
Orientação a objetos II

# Plataforma de desenvolvimento Eclipse

Prof. Felipe Scheidt

2016

# O que é uma IDE?

- Eclipse é uma IDE
  - *integrated development environment*
  - ou Ambiente de desenvolvimento integrado
- É um programa que possui um conjunto de funcionalidades embutidas, cuja finalidade é prover um modo mais fácil e produtivo de desenvolver software.

# Plugins

- Possui diversos *plugins* que adicionam funcionalidades ao Eclipse, dentre elas:
  - Php eclipse
  - JavaScript
  - Java
  - C/C++
  - Web
  - Modelagem



A screenshot of the Eclipse IDE plugin marketplace showing a list of available plugins. Each entry includes an icon, the plugin name, its size, and the number of times it has been downloaded. A 'Details' link is provided for each plugin.

Plugin Name	Size	Downloaded Times	Details
Eclipse IDE for Java EE Developers	212 MB	397,535	<a href="#">Details</a>
Eclipse Classic 3.7.2	174 MB	297,679	<a href="#">Details</a> <a href="#">Other Downloads</a>
Eclipse IDE for Java Developers	128 MB	116,411	<a href="#">Details</a>
Eclipse IDE for C/C++ Developers (includes Incubating components)	108 MB	61,067	<a href="#">Details</a>
Eclipse IDE for Java and Report Developers	243 MB	19,878	<a href="#">Details</a>
Eclipse IDE for JavaScript Web Developers	110 MB	18,662	<a href="#">Details</a>
Eclipse Modeling Tools	272 MB	9,712	<a href="#">Details</a>

# Ferramentas

- Possui diversas funcionalidades, por exemplo:
  - Um editor de texto;
  - Um compilador;
  - Um editor gráfico;
  - Um *debugger*
  - Um gerenciador de *plugins*

# Instalação

- Acessar o site:
- [www.eclipse.org](http://www.eclipse.org)
- “downloads”, selecionar a versão:
- Eclipse IDE for Java EE Developers, ~254 MB

Versão:



# Instalação

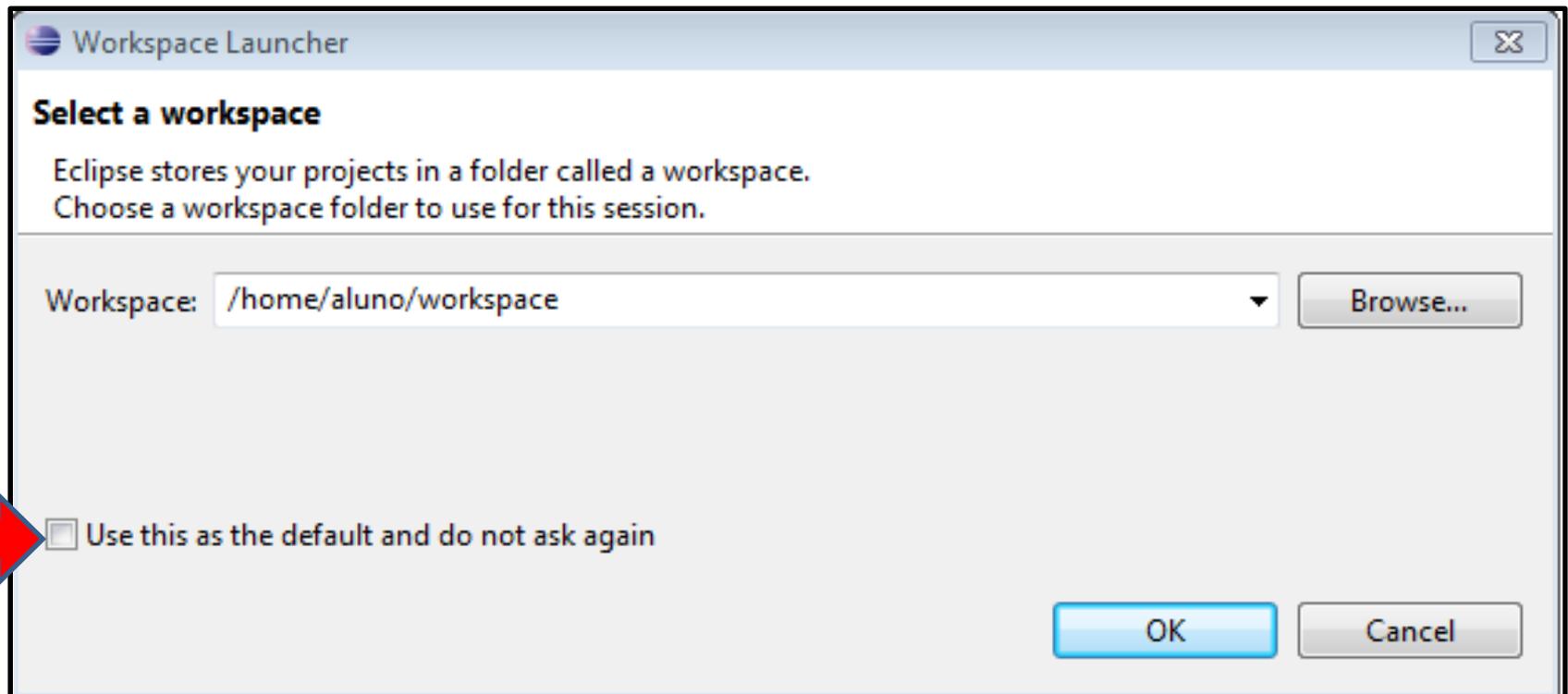
- Não é necessário instalar o eclipse, basta descompactar em uma pasta o arquivo baixado do site.
- Dentro da pasta eclipse, deve ser executado o seguinte arquivo para iniciar o eclipse:
  - eclipse.exe (windows)
  - eclipse (linux)

# Workspace

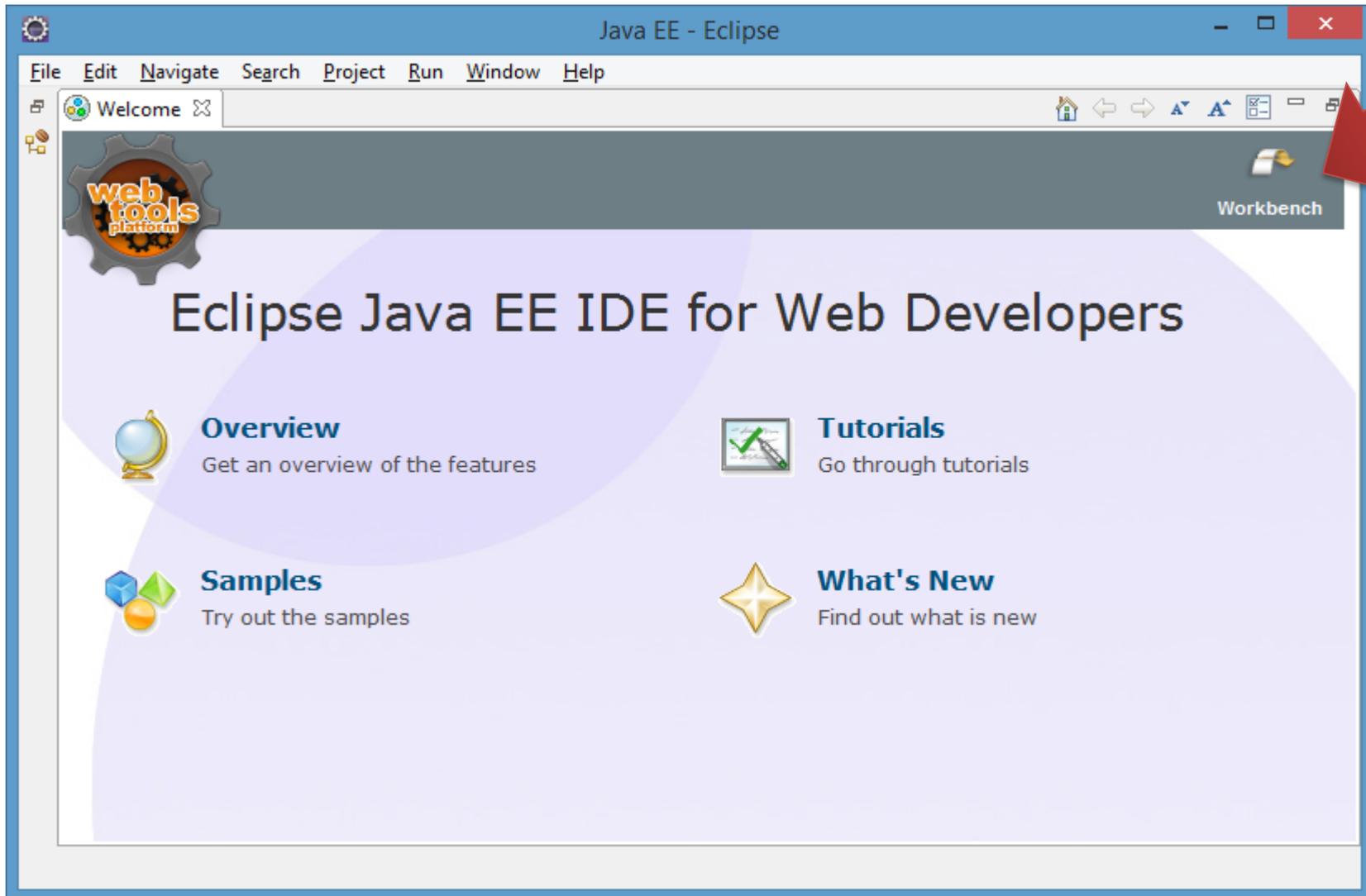
- O Eclipse trabalha com o conceito de Workspace.
- Basicamente é um diretório onde ficam armazenados os projetos java, juntamente com arquivos que salvam as preferências de configuração do Eclipse.
- Portanto é importante manter backup desta pasta.
- O Eclipse pode trabalhar com diversos Workspaces. Acessando o menu “File” -> “Switch workspace” é possível mudar entre um workspace e outro.
- Assim mantemos organizado nossos projetos.

# Criando o workspace

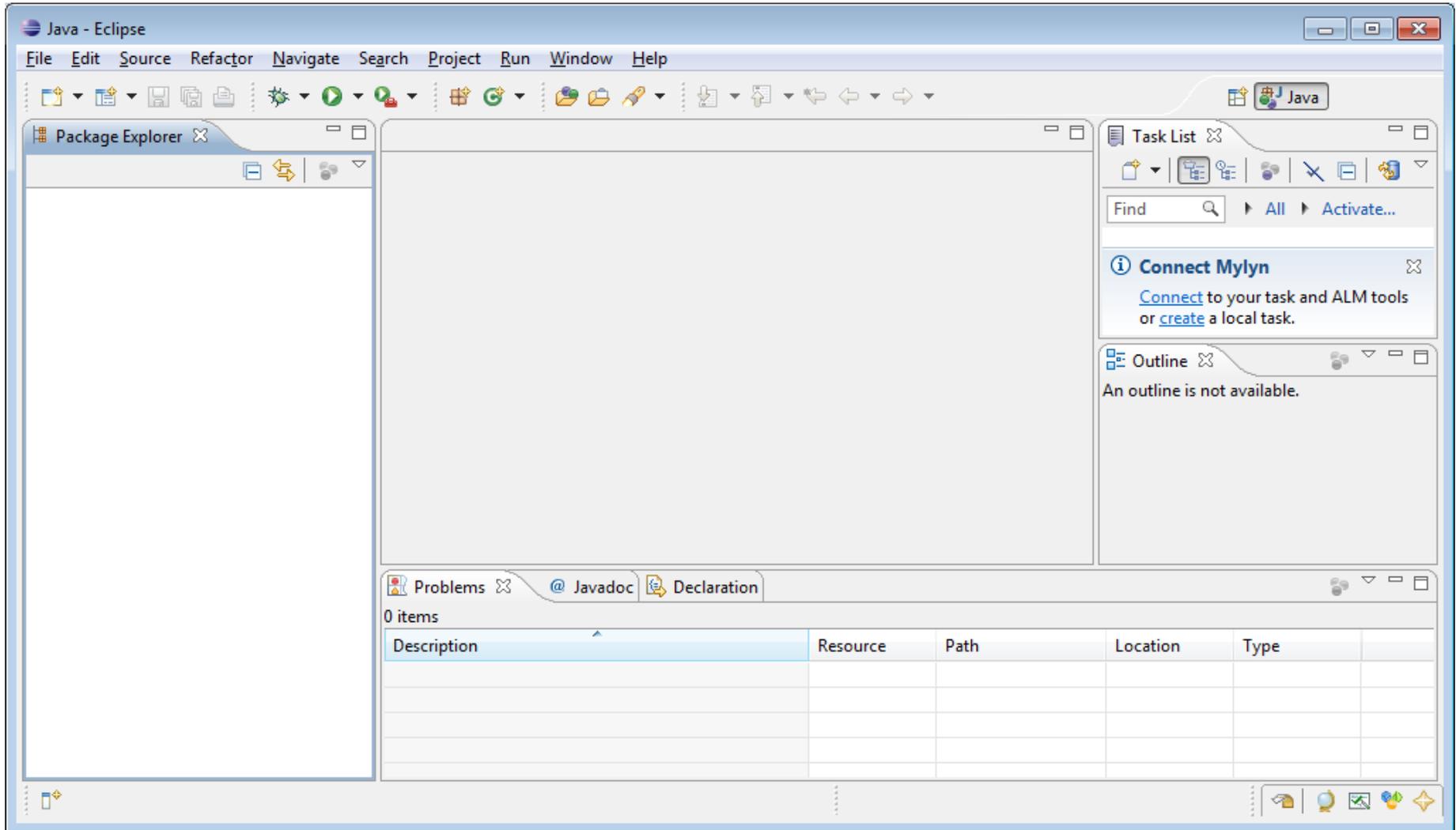
- Configure para o diretório:  
`/home/aluno/workspace`



# Tela inicial

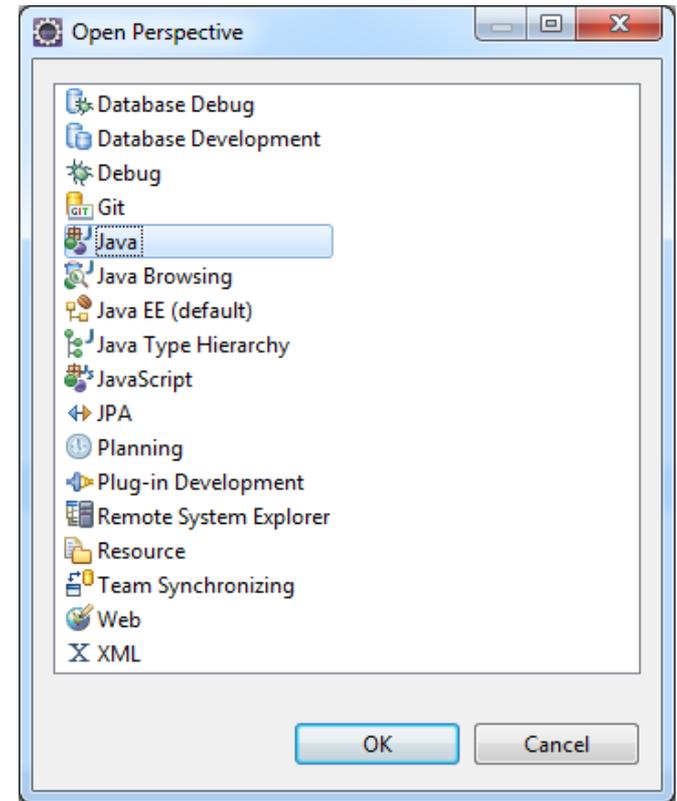


# Visão do eclipse



# Perspectivas

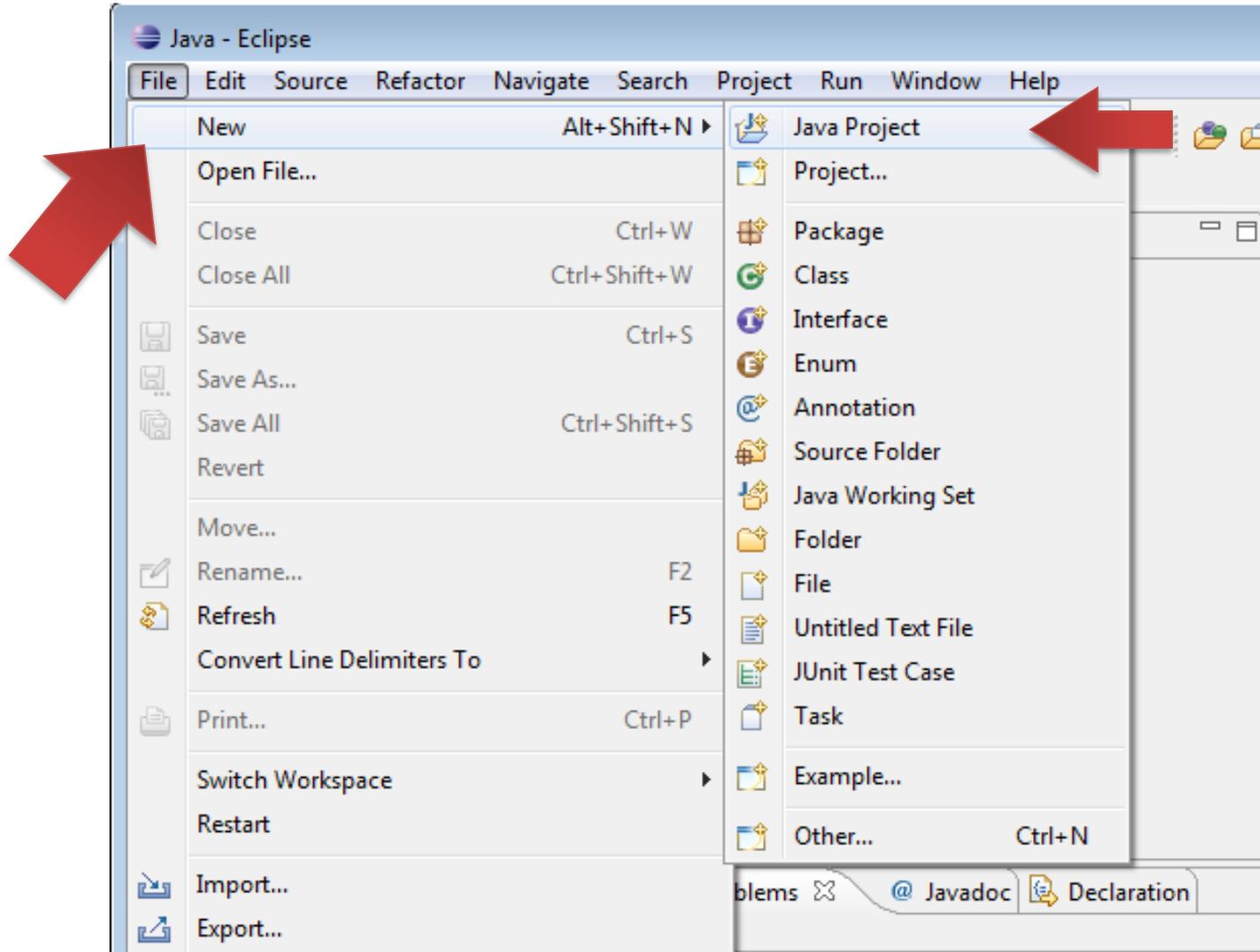
- O Eclipse possui diversas “perspectivas”, que configuram a aparência para apresentar as funcionalidades mais usadas naquela modalidade de desenvolvimento.
- Para as nossas aulas usaremos a perspectiva Java
- Para modificar a perspectiva acesse Window -> Perspective -> Open Perspective -> Other -> Java



# Criando um novo projeto

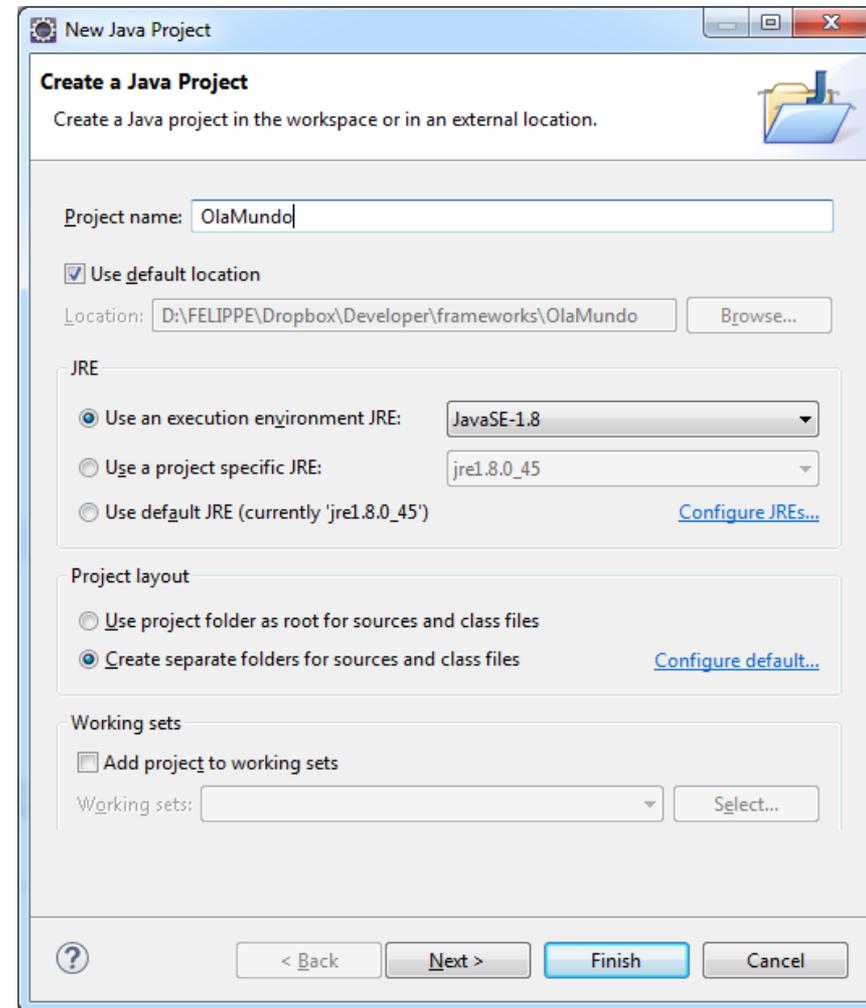
- Primeiramente clique em
  - File -> New -> Java Project 
- Um *wizard* de criação de projetos aparecerá

# Criando um novo projeto



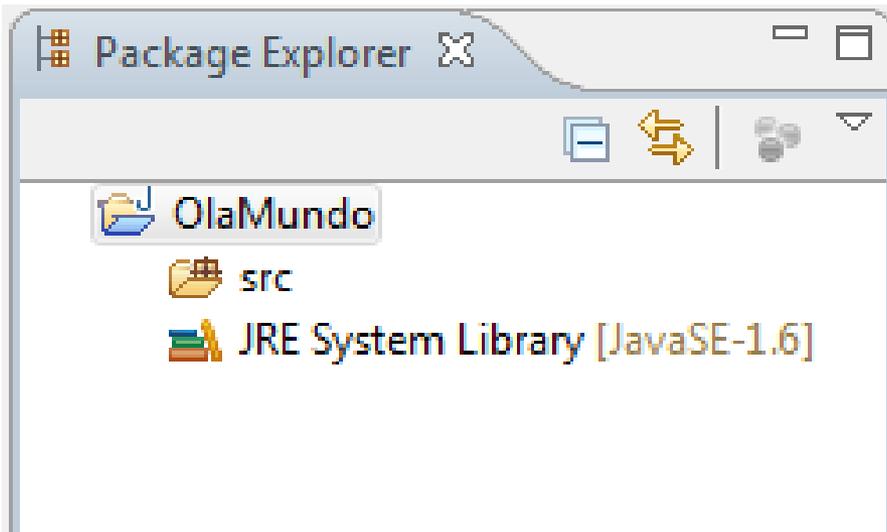
# Definindo o projeto

- Coloque um nome para o projeto:  
OlaMundo
- Escolher JRE 1.8
- Demais configurações, deixar padrão.
- Clique em **Finish**



# Estrutura do projeto

- No package Explorer vai aparecer o projeto recém criado.



O Package Explorer permite visualizar toda a estrutura de diretórios e arquivos contidos no projeto

# Criando uma classe

- Clicar botão direito na pasta src:

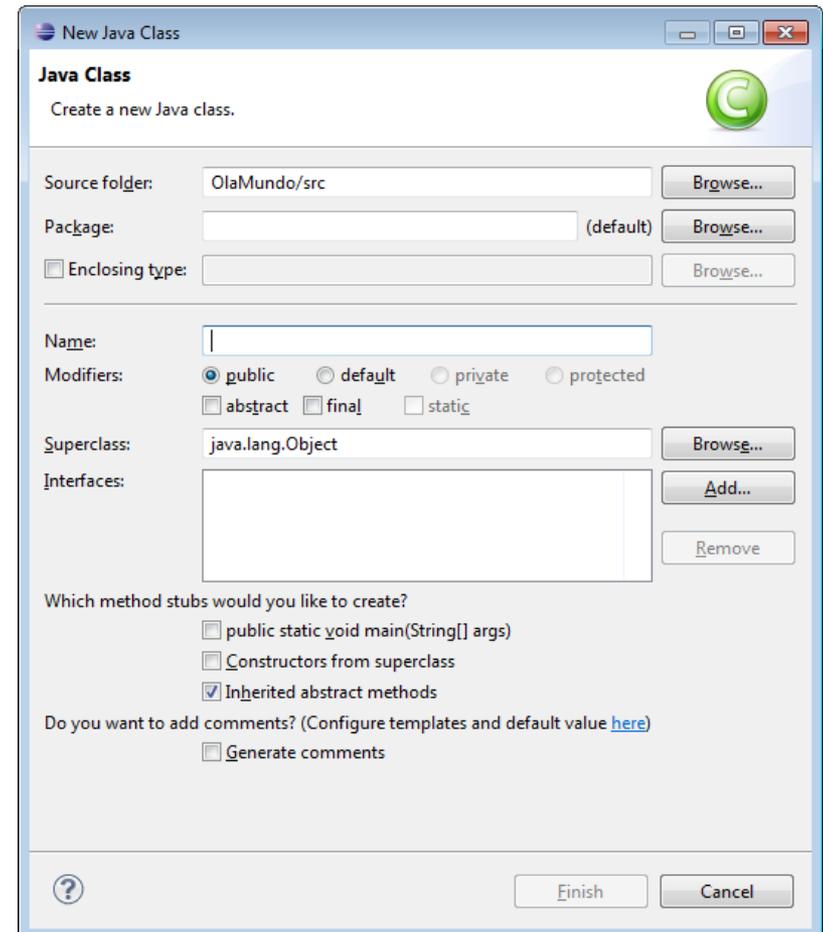
- New -> Class

- No campo Name digite:

Exemplo

- No campo Package:

br.ifpr.oo2



# Executando o código

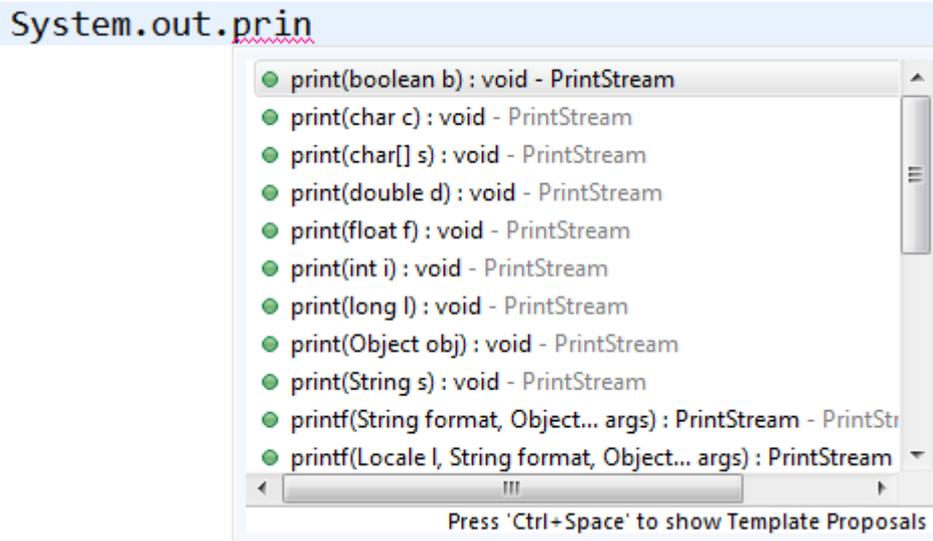
```
public class Exemplo {  
  
    public static void main(String[] args) {  
        System.out.println(">>>...");  
    }  
  
}
```

- Usar o atalho **CTRL + F11**
- ou clique no botão:



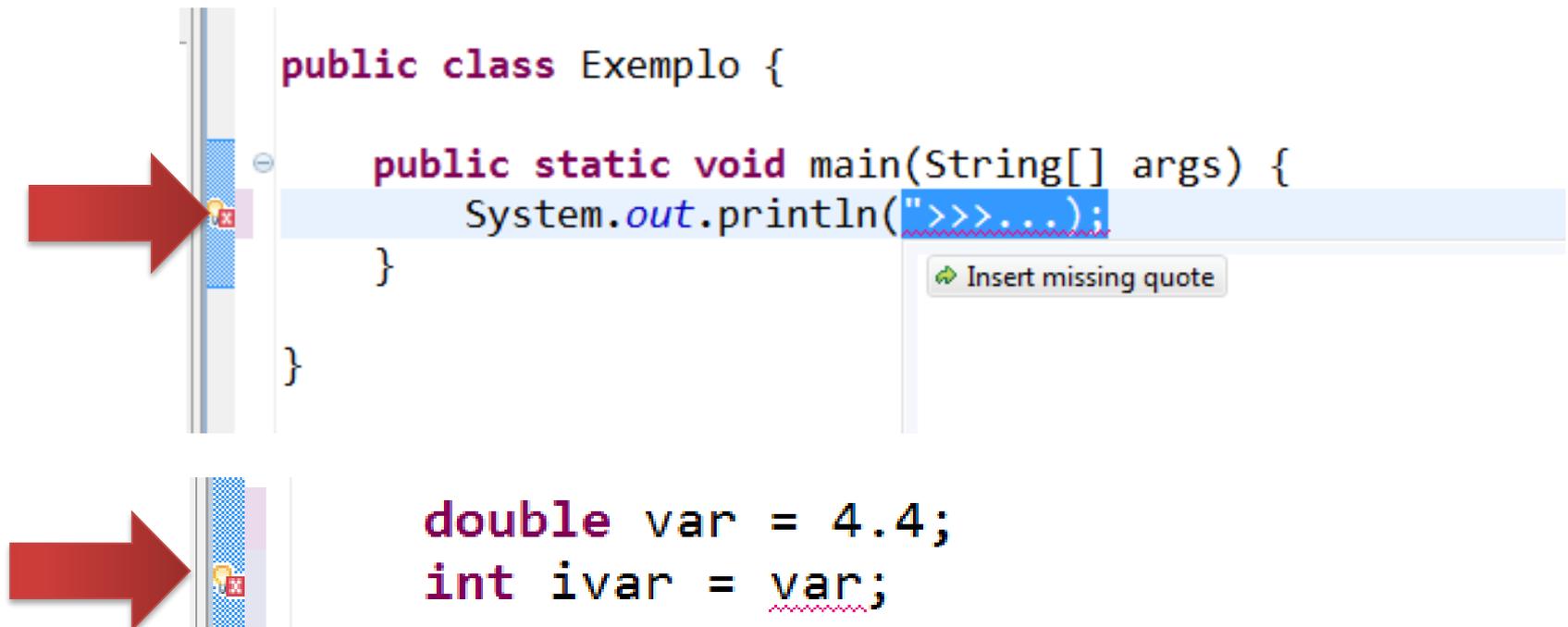
# Principais funcionalidades

- Detecção de erro
- Função autocompletar.



# Sugestão de correção

- Possui um assistente que sugere correções a serem feitas no código, quando encontrado um erro. Erros são sublinhados em vermelho.



The image shows two examples of code errors in an IDE. In the first example, a red arrow points to a blue error bar on the left side of the code editor. The code is a Java class named 'Exemplo' with a 'main' method. The line 'System.out.println(">>>...");' is highlighted in blue, and a tooltip suggestion 'Insert missing quote' is visible. In the second example, a red arrow points to a blue error bar on the left side of the code editor. The code is a Java snippet with 'double var = 4.4;' and 'int ivar = var;'. The variable 'var' in the second line is underlined in red, indicating an error.

```
public class Exemplo {  
    public static void main(String[] args) {  
        System.out.println(">>>...");  
    }  
}
```

Insert missing quote

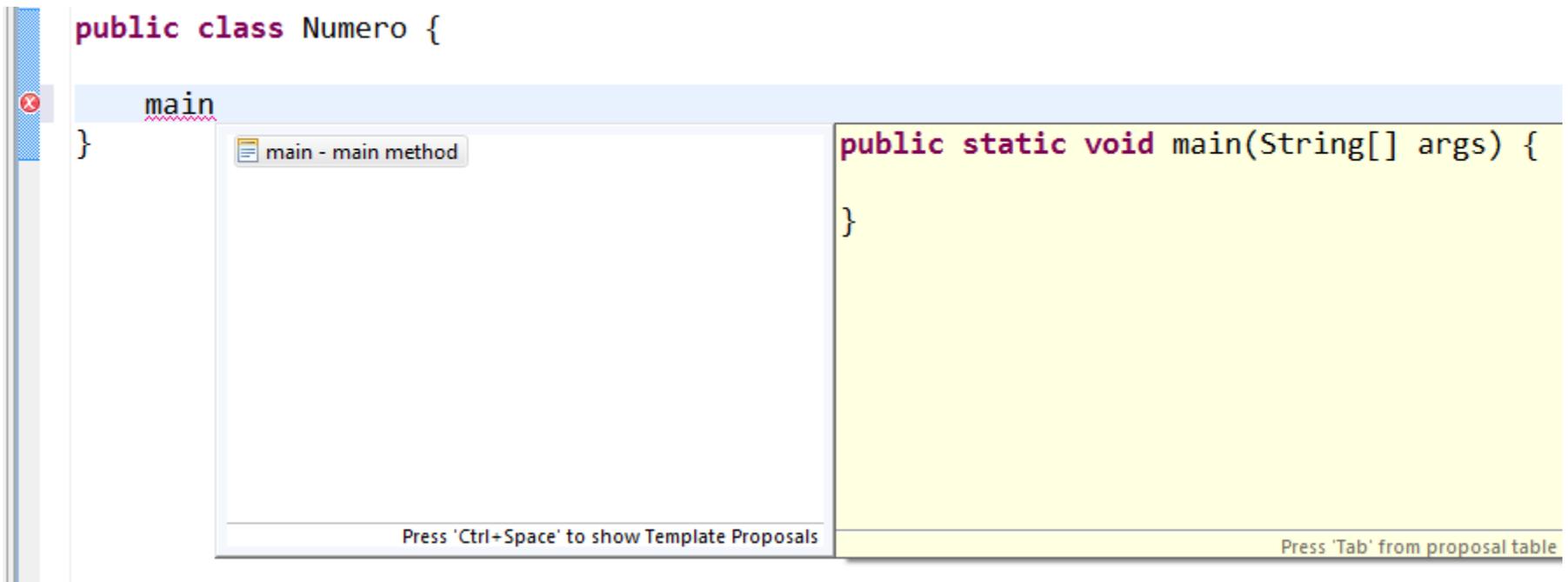
```
double var = 4.4;  
int ivar = var;
```

# Função ctrl+space

- É uma funcionalidade do eclipse que permite injetar código a partir de palavras chaves.
- Também usado para importar um pacote.
- Aprenda a usá-la para ganhar produtividade e diminuir erros de digitação.
- Qual a classe que permite ler um arquivo no java?
  - Buffer...

# Função ctrl+space

- Digite main e logo em seguida: **CTRL + SPACE**
- O eclipse completa a palavra com a definição do método main.  
Teste para sysout



The screenshot shows the Eclipse IDE interface. On the left, a snippet of Java code is visible: `public class Numero {` followed by a line where the word `main` is partially typed and underlined with a red squiggly line. Below this, a tooltip window titled "main - main method" is open, displaying the full signature of the `main` method: `public static void main(String[] args) {` followed by a closing brace `}`. At the bottom of the IDE, two instructions are visible: "Press 'Ctrl+Space' to show Template Proposals" and "Press 'Tab' from proposal table".

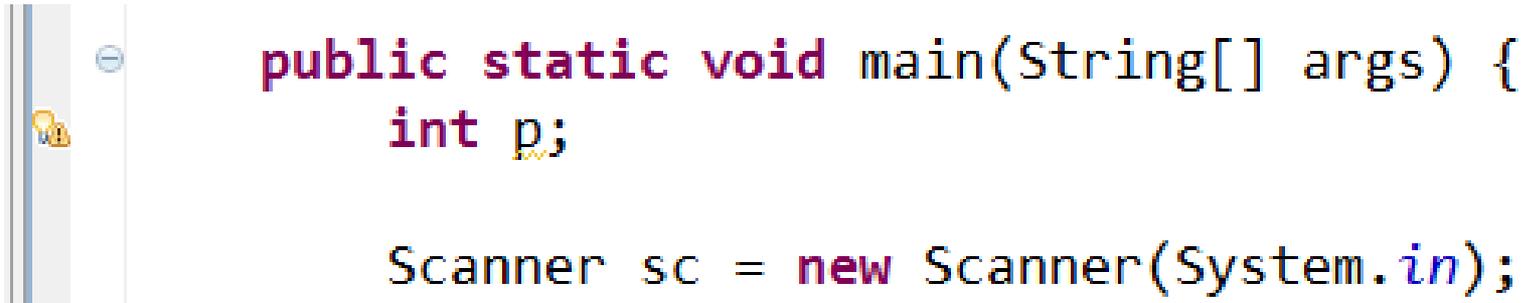
```
public class Numero {  
    main  
}  
public static void main(String[] args) {  
}
```

Press 'Ctrl+Space' to show Template Proposals

Press 'Tab' from proposal table

# Avisos (*warning*)

- Eclipse sublinha em amarelo, quando algo deve ser verificado no código, entretanto, não impede a compilação:

A screenshot of the Eclipse IDE's code editor. On the left side, there is a vertical toolbar with a warning icon (a yellow triangle with an exclamation mark) highlighted. The code editor shows the following Java code:

```
public static void main(String[] args) {  
    int p;  
  
    Scanner sc = new Scanner(System.in);
```

The variable 'p' in the second line is underlined in yellow, indicating a warning. The rest of the code is in its standard color scheme.

- Nesse exemplo, a variável “p” foi declarada, mas não é utilizada.

# Erros vs. Avisos

- Qual a diferença entre erros e avisos?
- No geral é a seguinte:
- **Avisos.** Algo no seu código fonte não está em conformidade com uma regra de boa programação, porém será compilado. Ex: declara variável mas não usa.
- **Erros.** Algo no seu código fonte apresenta uma incorretude e seu código não será compilado (grave).
- Porém esse comportamento pode ser alterado nas preferências do eclipse (não recomendado!!!)

# Teclas de atalho

**CTRL + /**

Comenta uma linha

**CTRL + D**

Deleta uma linha

**CTRL + S**

Salva arquivo

**CTRL + SHIF + F**

Indentação do código

**CTRL + SHIF + O**

Importa pacotes necessários

**CTRL + M**

Maximiza a janela de trabalho

# Teclas de atalho

**ALT + UP/DOW** Move uma linha para cima/baixo

**CTRL + SHIFT + H** Encontra rapidamente uma classe

**CTRL + SHIFT + R** Encontra rapidamente qualquer arquivo

**CTRL + 1** Quick fix

# Getters and Setters

- Permite gerar os métodos **getters** e **Setters** para os atributos de uma classe.
- Para gerar os métodos, clique com o botão direito na área do código, clicar em "source" -> "generate getters and setters" -> select All

# Atividade

- 1) Escreva um programa no Eclipse que lê 2 números inteiros do teclado e imprime o **maior** deles;
- 2) Altere este programa para que ele imprima também a **soma** e a **média** aritmética dos 2 números. Crie métodos para melhor organizar o seu código.
- 3) Faça um programa que possui uma classe chamada **Idade**. Essa classe possui um método – **getMeses(int idade)** – que calcula a idade em meses. Teste o funcionamento da mesma.

# Atividade

- 4) Implemente uma classe Computador, que possui: (1) uma placa mãe. Essa placa mãe possui: (a) uma placa de vídeo, (b) um HD e (c) um driver de DVD. Instancie as classes e relacione as mesmas com a classe Computador. Crie ao final um método start que inicializa do computador.