

# Crescer 2015/2

Java

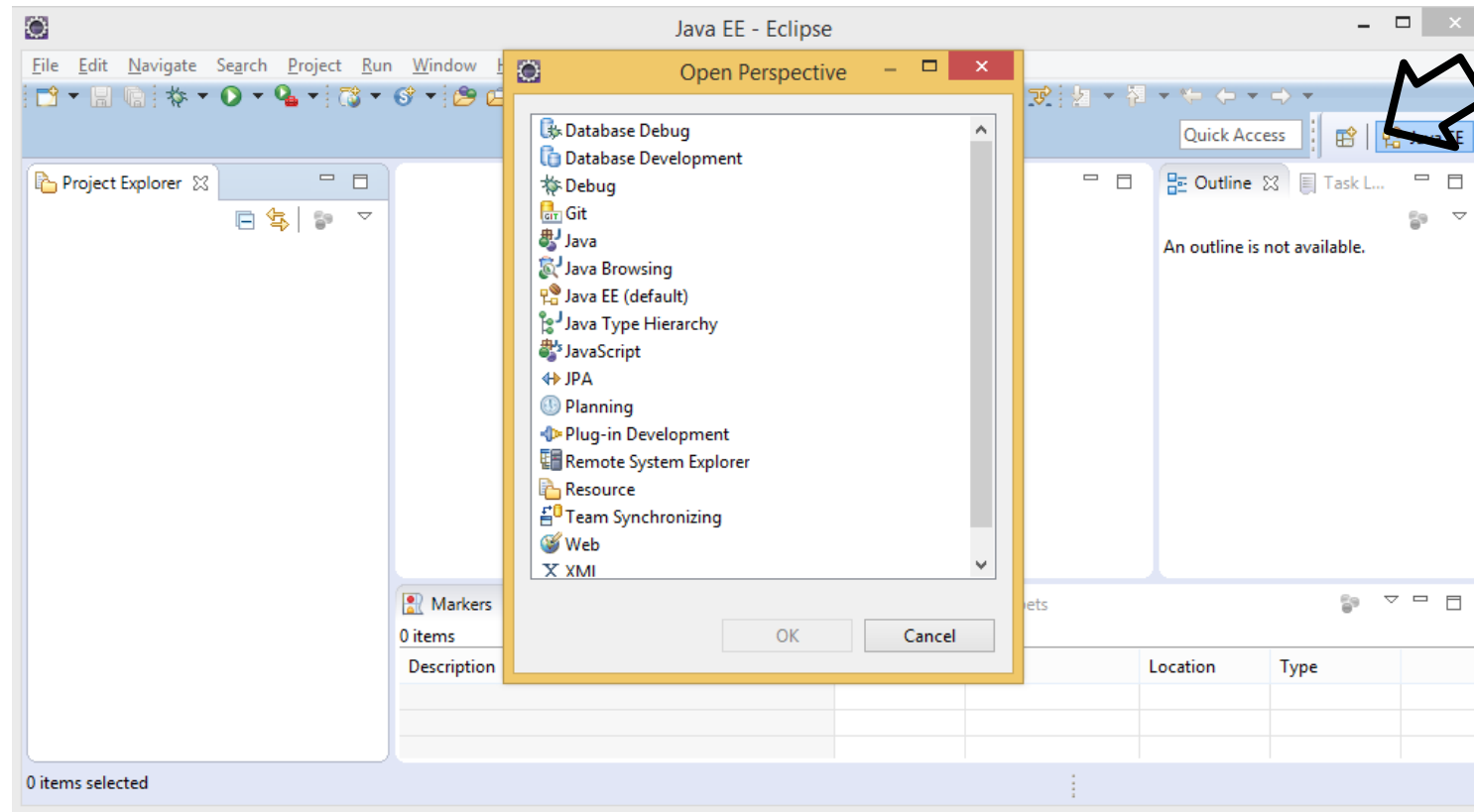
# Quem sou...

- ▶ Ingo Jost
- ▶ Desenvolvedor na CWI (11/2007)
- ▶ Contato:
  - ▶ [ingo@cwi.com.br](mailto:ingo@cwi.com.br)
  - ▶ Skype: [ingo\\_nh@hotmail.com](https://www.hotmail.com/ingo_nh)
  - ▶ Slack

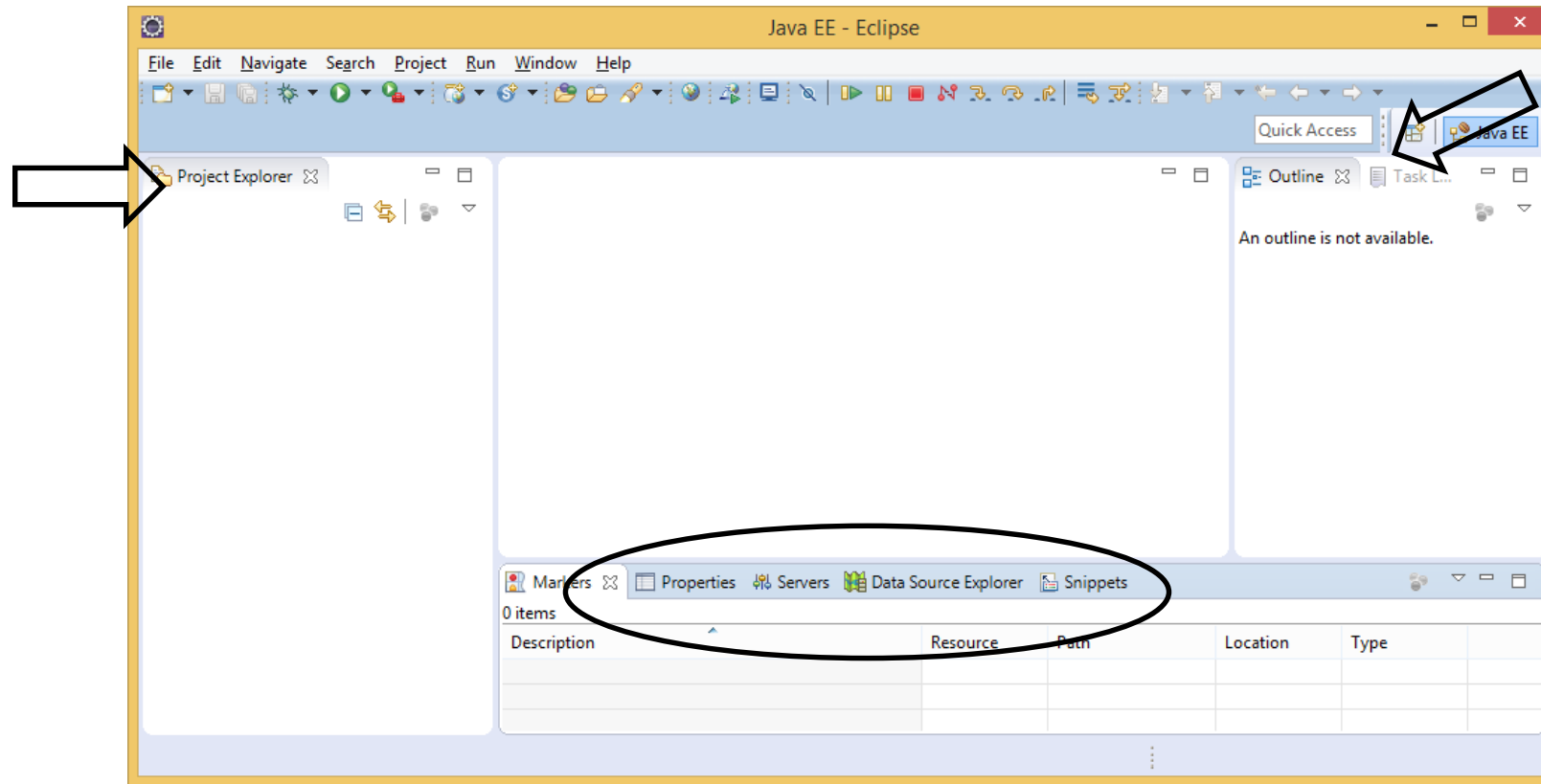
# Eclipse

- ▶ <http://www.eclipse.org/>
- ▶ *Integrated Development Environment (IDE)*
  - ▶ Desenvolvida em Java
  - ▶ Plugins
- ▶ Workspace
- ▶ *Perspective*
  - ▶ Ambientes dentro do Eclipse (*Database, Team Synchronizing, GIT, Java, Java EE...*)
- ▶ *Views*
  - ▶ Cada perspectiva possui *Views* default

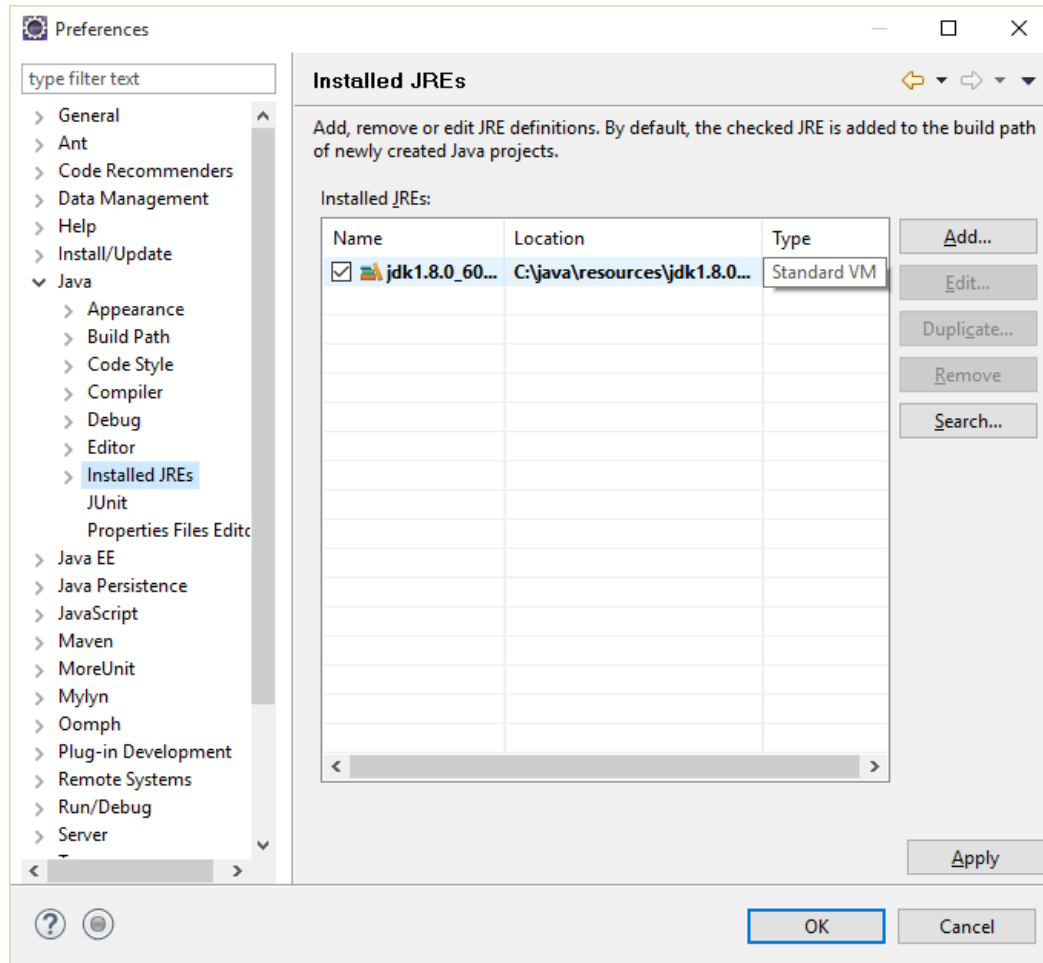
# Eclipse - *Perspective*



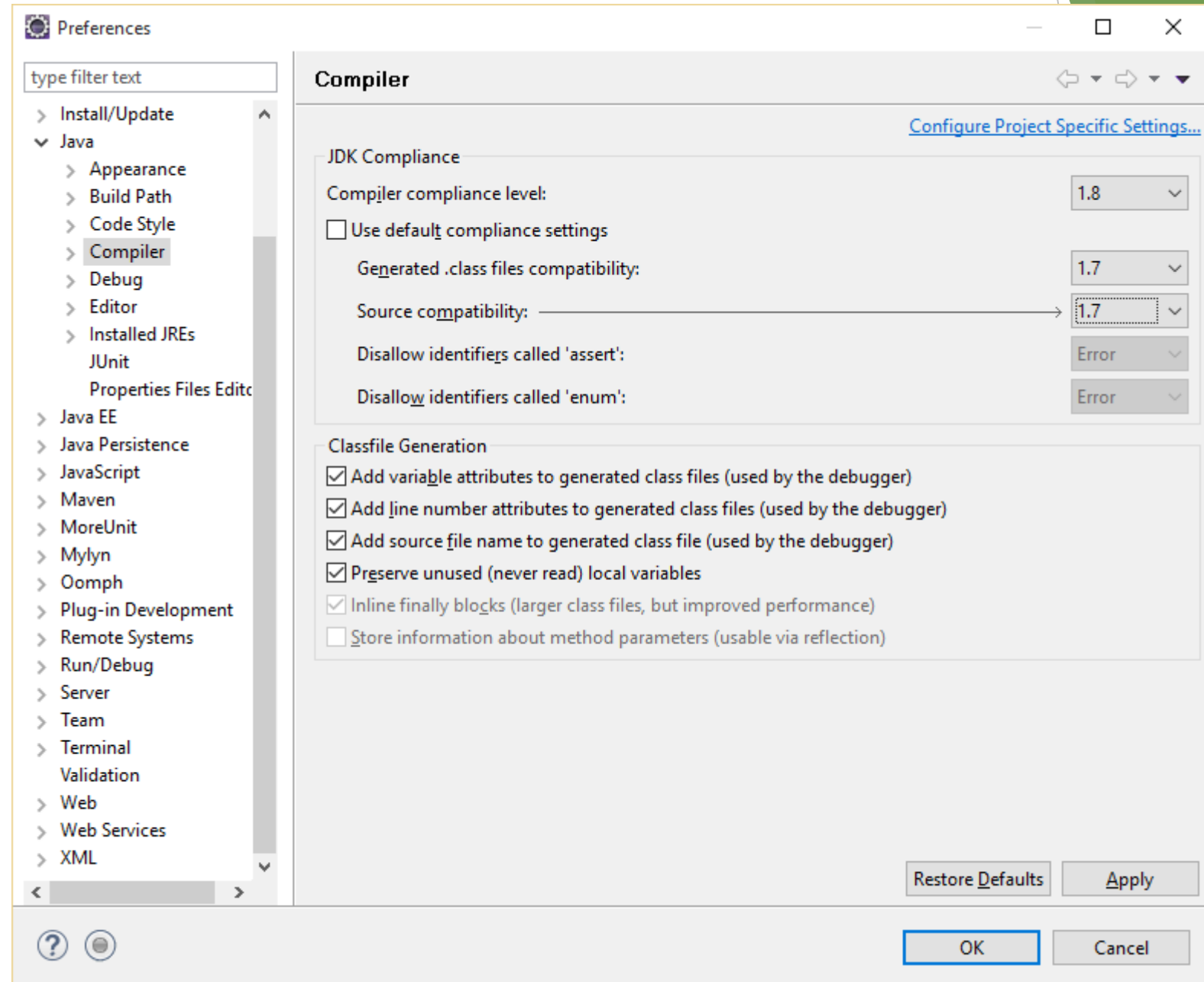
# Eclipse - Views



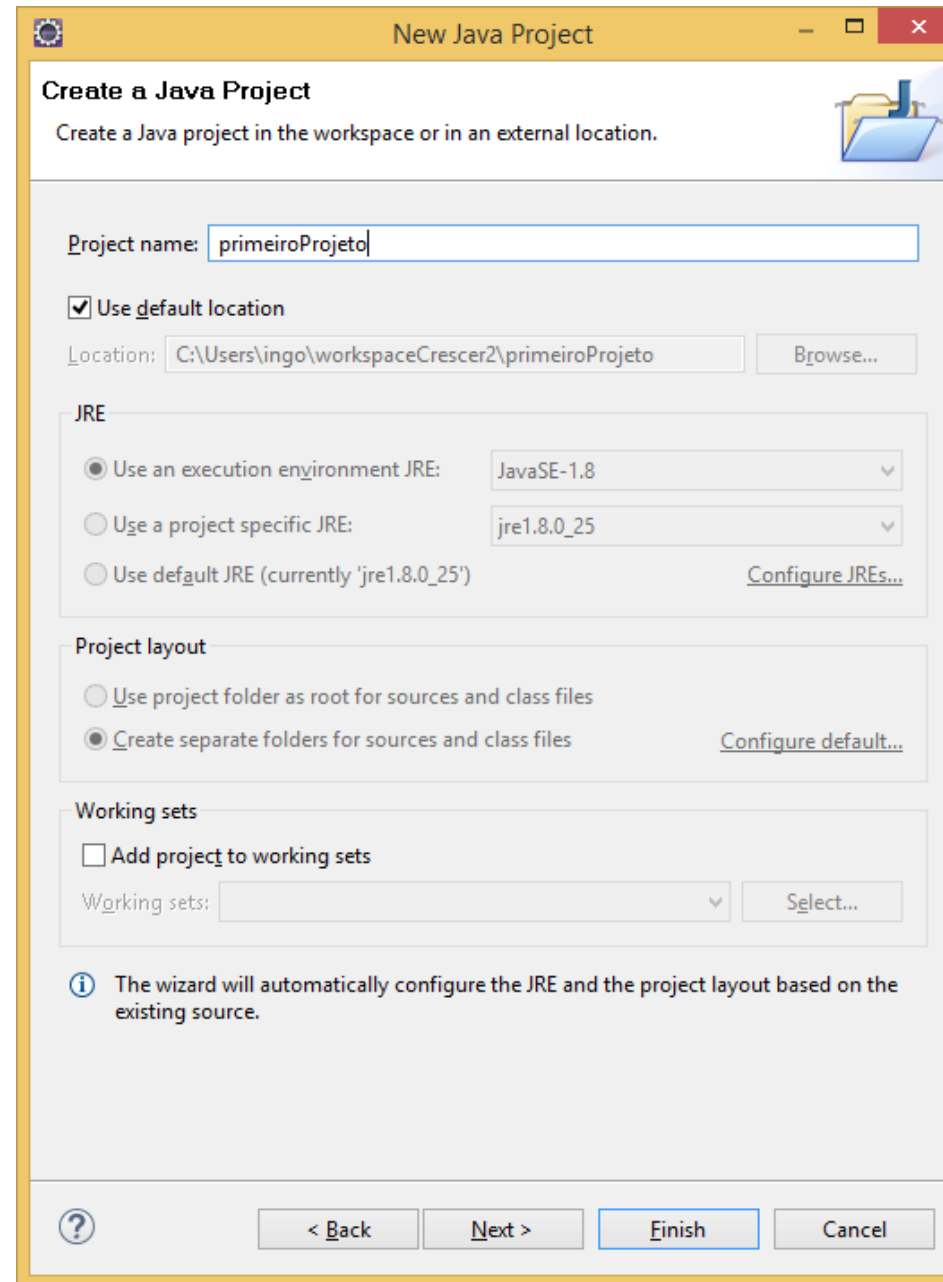
# Configurar JRE



# Compiler



# Eclipse - Projetos



The screenshot shows the 'New Java Project' wizard in Eclipse. The window title is 'New Java Project'. The main heading is 'Create a Java Project' with a subtext 'Create a Java project in the workspace or in an external location.' and a folder icon. The 'Project name' field contains 'primeiroProjeto'. The 'Use default location' checkbox is checked, and the 'Location' field shows 'C:\Users\ingo\workspaceCrescer2\primeiroProjeto' with a 'Browse...' button. The 'JRE' section has three radio buttons: 'Use an execution environment JRE:' (selected) with a dropdown showing 'JavaSE-1.8', 'Use a project specific JRE:' with a dropdown showing 'jre1.8.0\_25', and 'Use default JRE (currently 'jre1.8.0\_25')' with a 'Configure JREs...' link. The 'Project layout' section has two radio buttons: 'Use project folder as root for sources and class files' and 'Create separate folders for sources and class files' (selected) with a 'Configure default...' link. The 'Working sets' section has an unchecked 'Add project to working sets' checkbox, a 'Working sets:' dropdown, and a 'Select...' button. An information icon and text state: 'The wizard will automatically configure the JRE and the project layout based on the existing source.' The bottom bar contains a help icon, '< Back', 'Next >', 'Finish' (highlighted), and 'Cancel' buttons.

**Create a Java Project**  
Create a Java project in the workspace or in an external location.

Project name:

☒ Use default location

Location:  [Browse...](#)

**JRE**

☒ Use an execution environment JRE:  [Configure JREs...](#)

☐ Use a project specific JRE:  [Configure JREs...](#)

☐ Use default JRE (currently 'jre1.8.0\_25')

**Project layout**

☐ Use project folder as root for sources and class files

☒ Create separate folders for sources and class files [Configure default...](#)

**Working sets**

☐ Add project to working sets

Working sets:  [Select...](#)

**i** The wizard will automatically configure the JRE and the project layout based on the existing source.

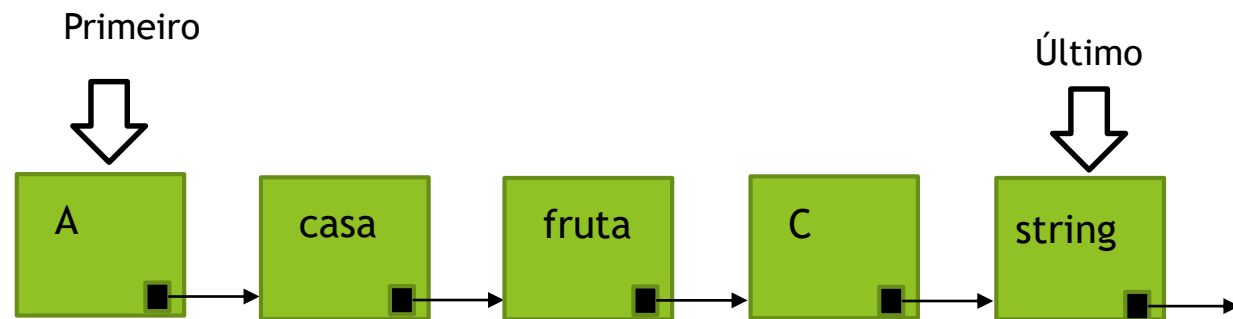
[?](#) [< Back](#) [Next >](#) [Finish](#) [Cancel](#)



# Exemplo - Lista Encadeada

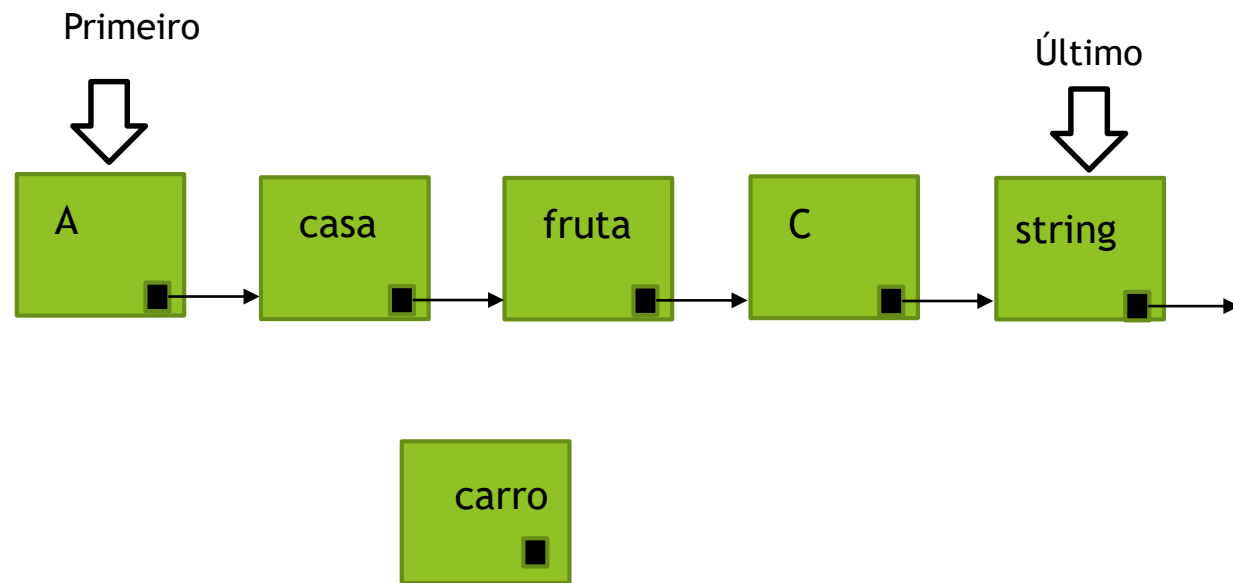
- ▶ Inner classes
- ▶ Tratamento Exceções
- ▶ ...
- ▶ LP e Estrutura de Dados

# Exemplo - Lista Encadeada



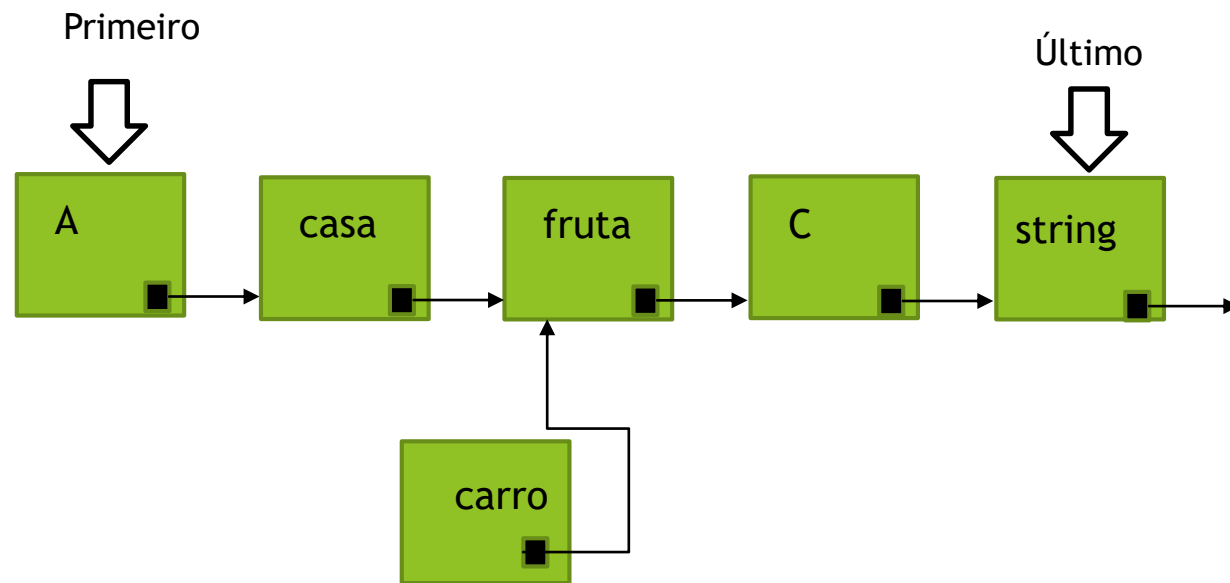
# Lista Encadeada - inclusão

Inclusão na terceira posição



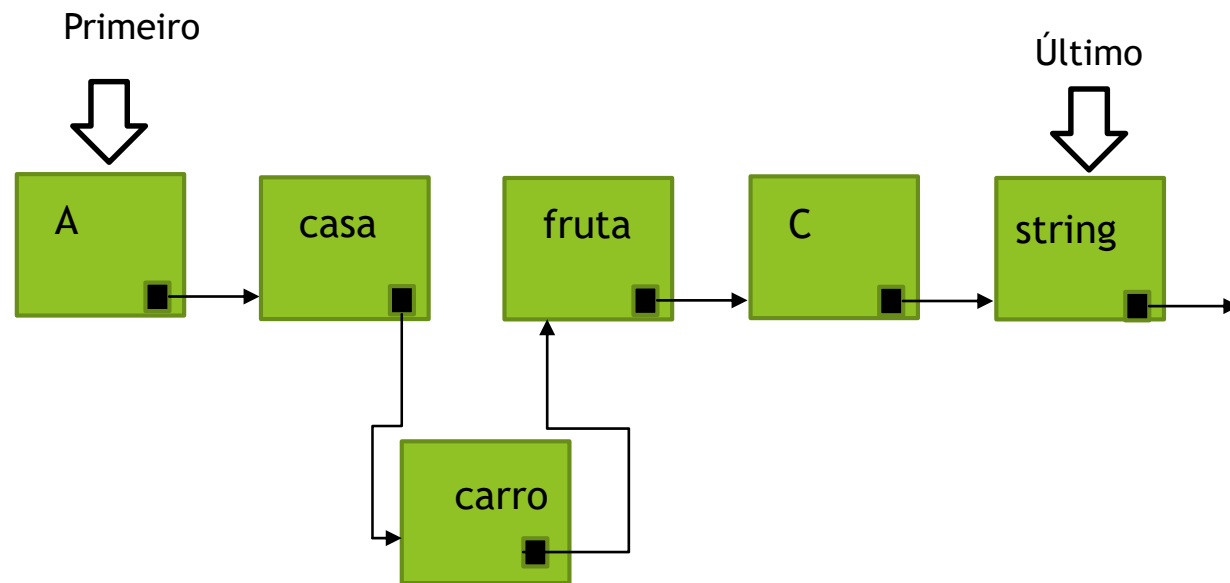
# Lista Encadeada - inclusão

Inclusão na terceira posição



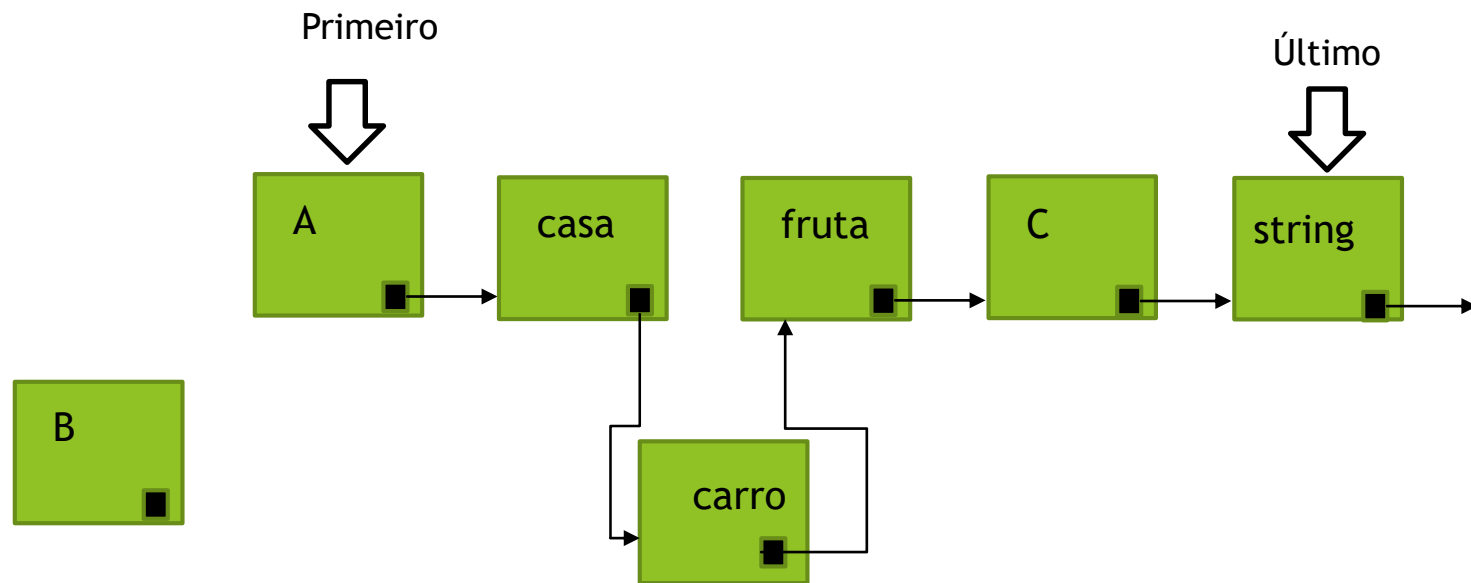
# Lista Encadeada - inclusão

Inclusão na terceira posição



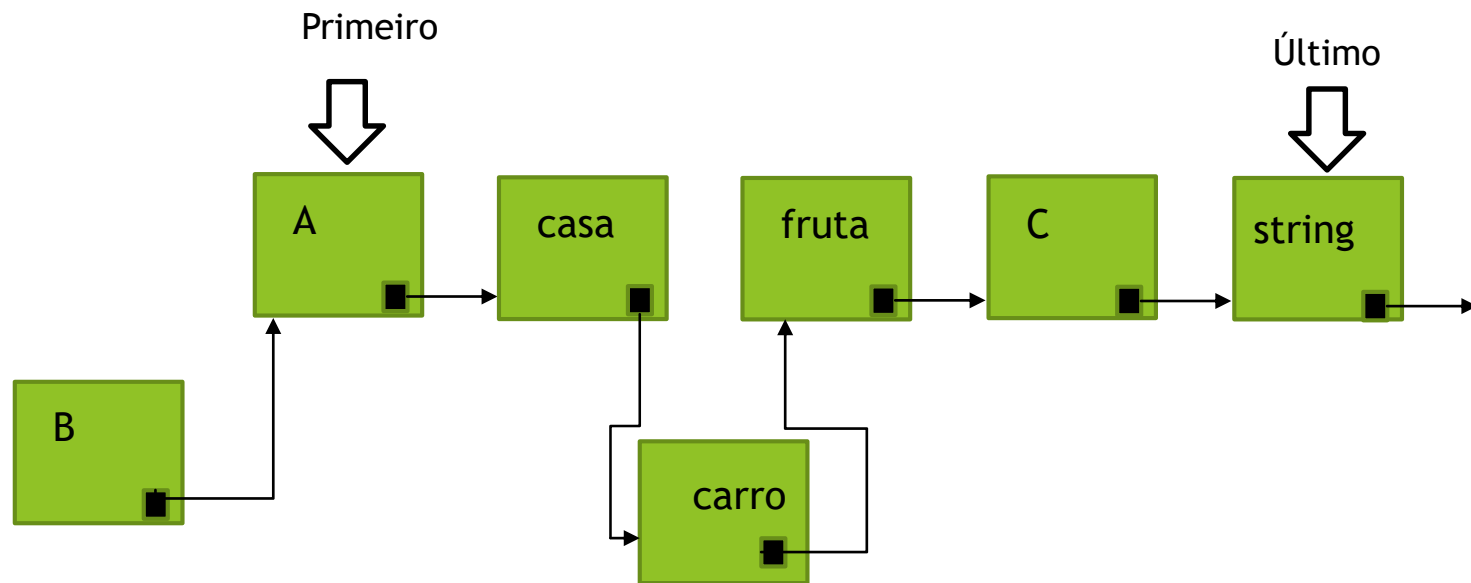
# Lista Encadeada - inclusão

Inclusão na primeira posição



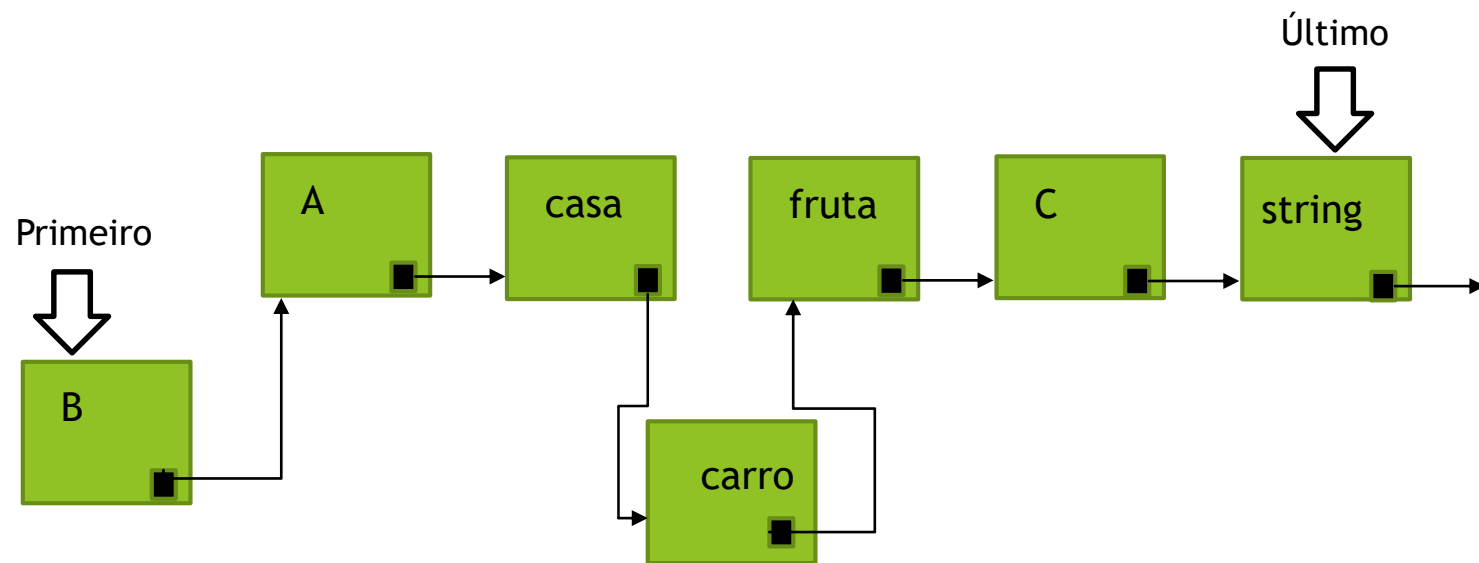
# Lista Encadeada - inclusão

Inclusão na primeira posição



# Lista Encadeada - inclusão

Inclusão na primeira posição



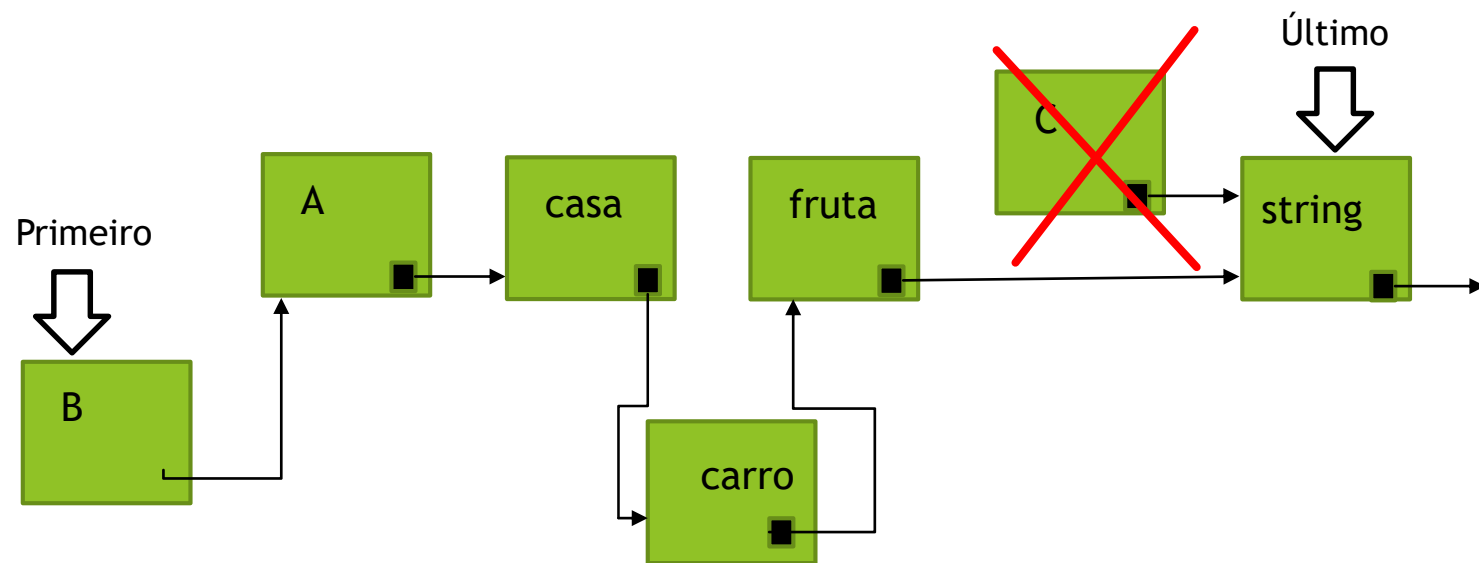


# Exercícios

- ▶ Criar classes Nodo e ListaEncadeada
  - ▶ getFirst e getLast
- ▶ Implementar método para incluir no início
  - ▶ Teste unitário
- ▶ Implementar método lista
- ▶ Implementar método incluir no final

# Lista Encadeada - Remoção

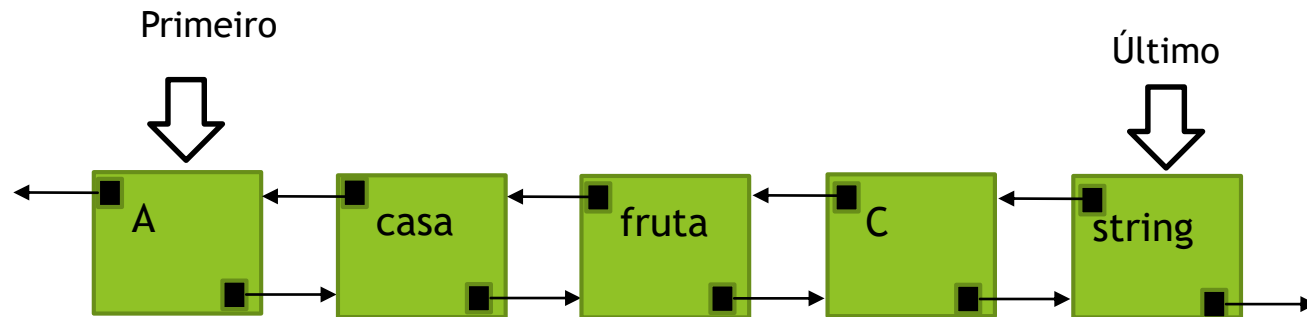
Remoção na sexta posição



# Exercícios

- ▶ Métodos
  - ▶ Remove primeiro
  - ▶ Insere no meio
  - ▶ Remove no meio

# Exemplo - Lista Duplamente Encadeada



# Exercício

- ▶ Implementar
  - ▶ Criar classe ListaDuplamenteEncadeada
  - ▶ Inclusão de item no início da classe

# Java - *inner classes*

- ▶ Classes em um arquivo .java
- ▶ *Inner class* (classe aninhada)
  - ▶ Exemplo - lista encadeada

# Classes anônimas

- ▶ Herança de classes
- ▶ Implementação de interface
- ▶ Exemplo - `JButton.addActionListener`

# Exceções

- ▶ Tratando exceções
  - ▶ Gerando exceção na Lista criada
  - ▶ Lançando exceção
- ▶ Criando as próprias exceções
- ▶ Checked e unchecked exceptions



# Lendo dados do teclado

- ▶ Classes

- ▶ System
- ▶ Console
- ▶ Scanner

```
InputStreamReader isr = new InputStreamReader(System.in);  
BufferedReader br = new BufferedReader(isr);  
br.readLine();
```

# Exercícios

- ▶ Implemente os métodos já construídos na ListaEncadeada na ListaDuplamenteEncadeada
- ▶ Modifique as implementações para que as estruturas aceitem qualquer tipo, não apenas strings (Generics)
- ▶ Herança, Interface ...?
- ▶ Crie uma aplicação com um menu de opções de inserção, remoção e lista para as duas estruturas, para que o usuário informe a opção desejada
- ▶ Crie exceções caso julgue necessário