

Programação Orientada a Objetos

ITE - 003

profº Mauricio Conceição Mario

Conteúdo Programático

- Princípios de Programação Orientada a Objetos:
 - Abstração
- Classes e Objetos
- Encapsulamento
- Escopo de Classe e Escopo de Instância
- Construtores
- Definição de Pacotes
- Herança
- Classes Abstratas e Concretas
- Polimorfismo
- Interfaces
- Técnicas de programação com a linguagem Java

Critério de Avaliação

1º bimestre:

$$\text{Média 1} = 0,4 * \text{trabalho} + 0,6 * \text{prova 1}$$

2º bimestre:

$$\text{Média 2} = 0,4 * \text{trabalho} + 0,6 * \text{prova 2}$$

$$\text{Média final} = (\text{Média 1} + \text{Média 2}) / 2$$

prova 3: substitui a prova1 ou prova2

Referências Bibliográficas

- Java 7 - Ensino Didático
Sérgio Furgeri - Editora Érica
- Java - como Programar - 8ª edição
H. M. Deitel & P. J. Deitel / Ed. Bookman
- Programação Orientada a Objeto - Referências: Estruturas de Dados & Algoritmos em Java.
Goodrich, M. T. e Tamassia R.. Editora Bookman, Porto Alegre - 2013.
- Modelagem de Sistemas Orientados a Objetos – Ensino Didático
Sérgio Furgeri - Editora Érica
- Programação Orientada a Objetos em Java
Isaias Camilo Boratti – Editora Visual Books ISBN 85-7502-199-0.
Florianópolis, 2007

Programação Orientada a Objetos

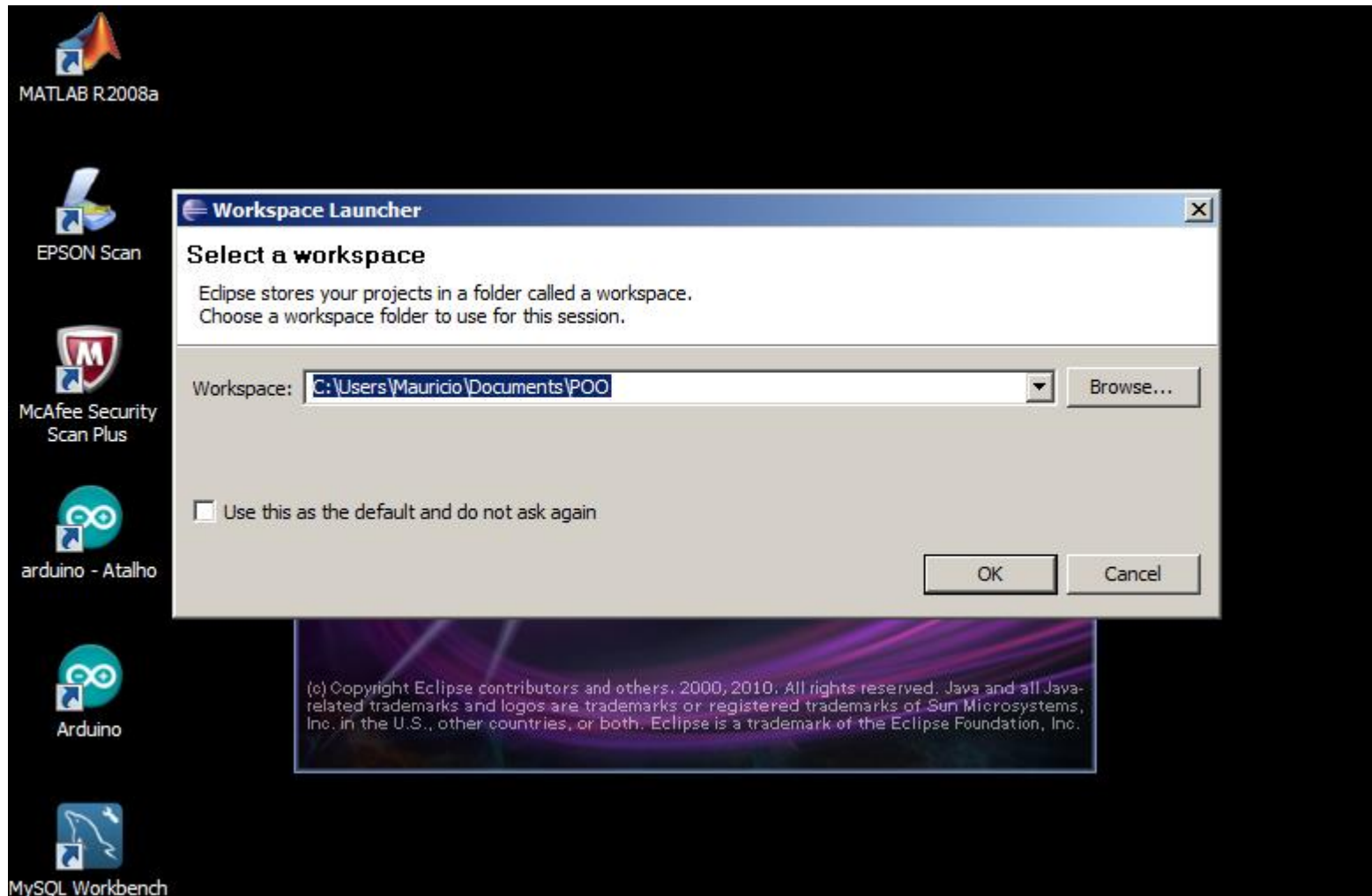
Abstração: genericamente abstração é o processo de análise de determinada situação do mundo real, através do qual se determinam os aspectos e fenômenos considerados essenciais àquela situação, excluindo os aspectos irrelevantes (Boratti, 2007).

Em programação orientada a objetos abstração é a decomposição de um sistema complexo em suas partes fundamentais, descrevendo-o em linguagem simples e precisa. A descrição das partes de um sistema consiste em atribuir-lhes nomes e descrever suas funcionalidades (Goodrich & Tamassia, 2013).

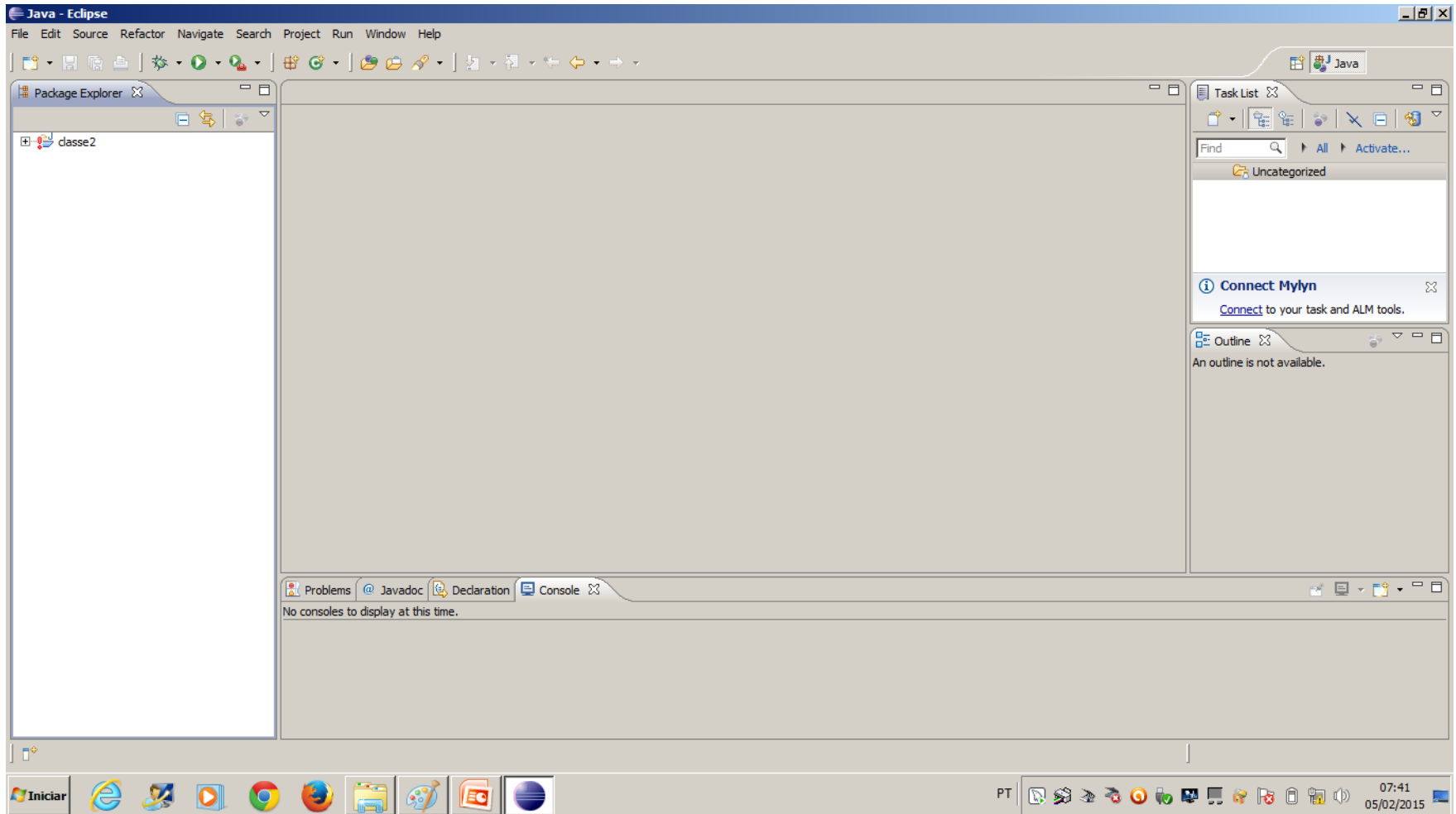
Projeto Java no Eclipse

- Java: www.sun.com (redireciona para página da *Oracle*)
- Fazer download de versão compatível com o sistema operacional →
- Windows ou Linux, 32 ou 64 bits; instalar
- Fazer download da IDE Eclipse: www.eclipse.org → instalar
- Criar pasta onde será desenvolvido o projeto Java;
- Abrir o Eclipse, redirecionando através do browse, para a pasta criada:

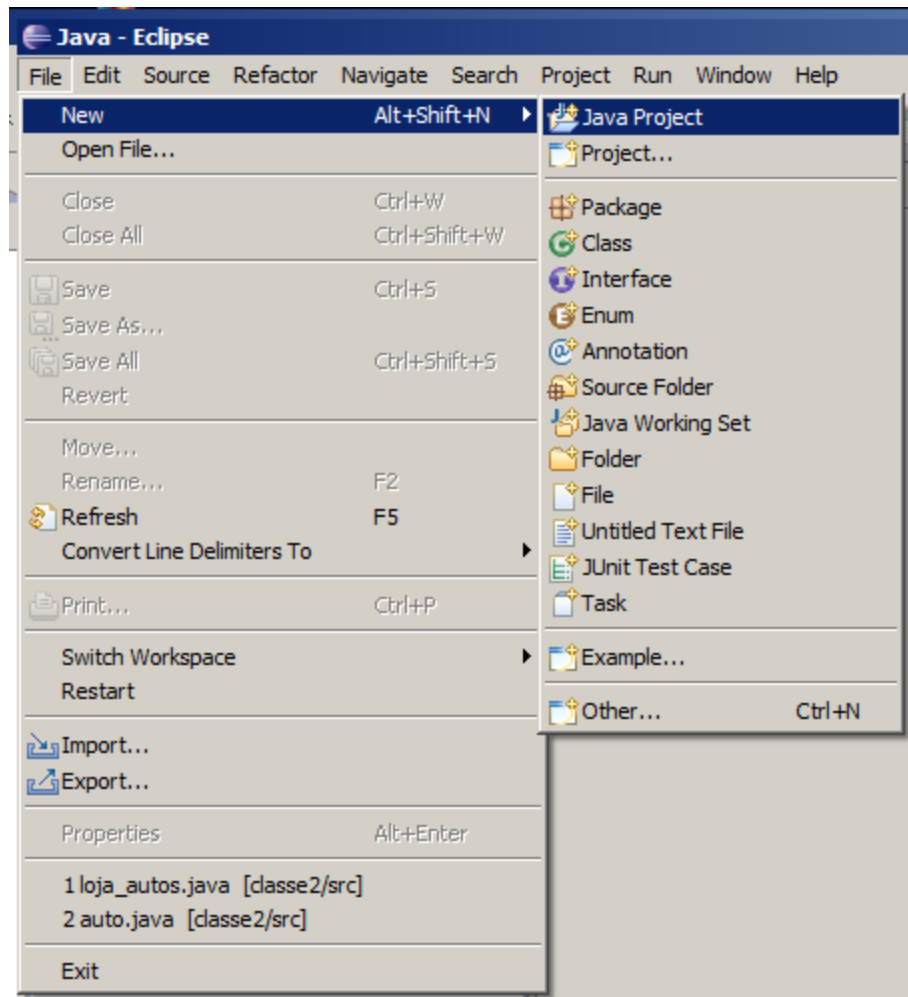
Projeto no Eclipse



Projeto no Eclipse



Projeto no Eclipse



Projeto no Eclipse

New Java Project

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:

☐ Use a project specific JRE:

☒ Use default JRE (currently 'jre1.8.0_20') [Configure JREs...](#)

Project layout

☐ Use project folder as root for sources and class files

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

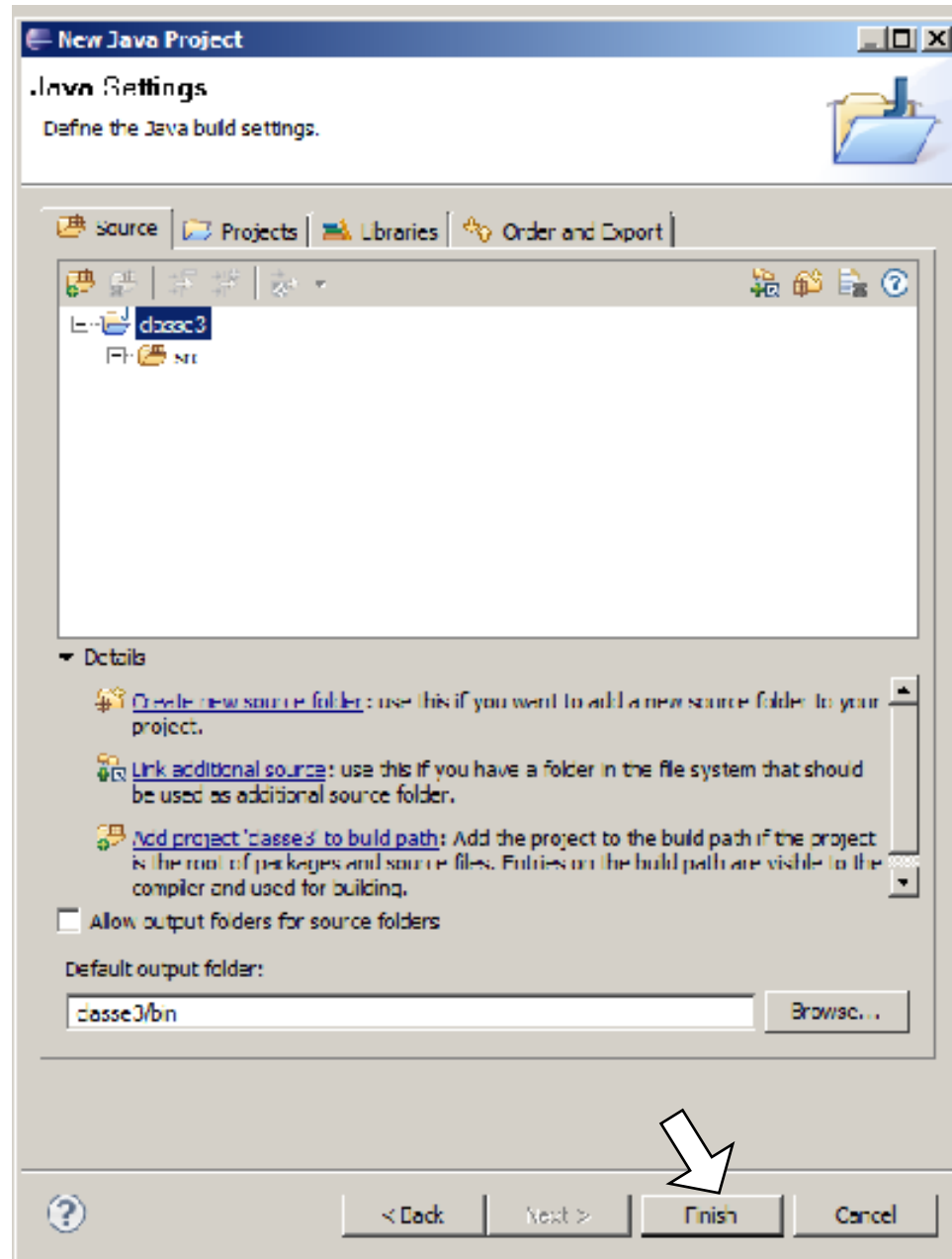
Working sets

☐ Add project to working sets

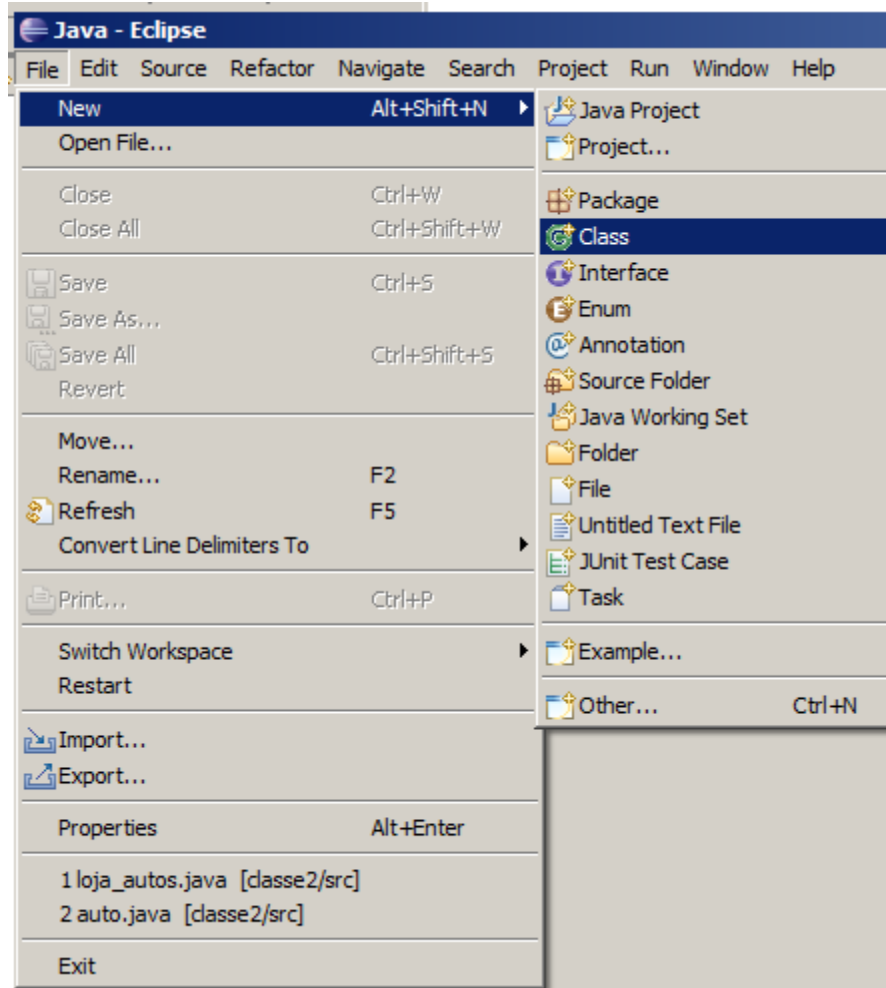
Working sets: [Select...](#)

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

Projeto no Eclipse



Projeto no Eclipse



Projeto no Eclipse

New Java Class

Java Class

⚠ The use of the default package is discouraged.

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)

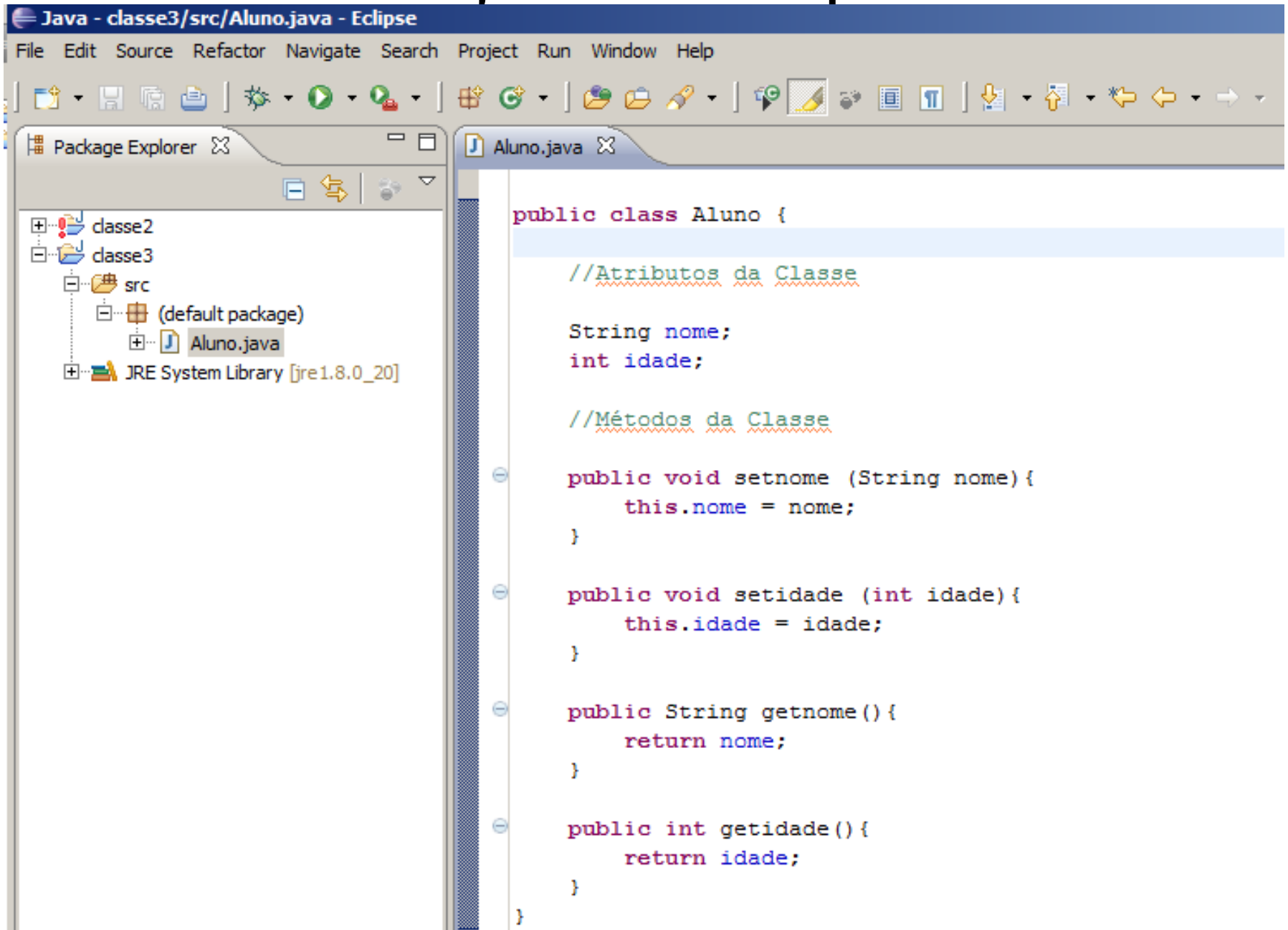
☐ Constructors from superclass

☒ Inherited abstract methods

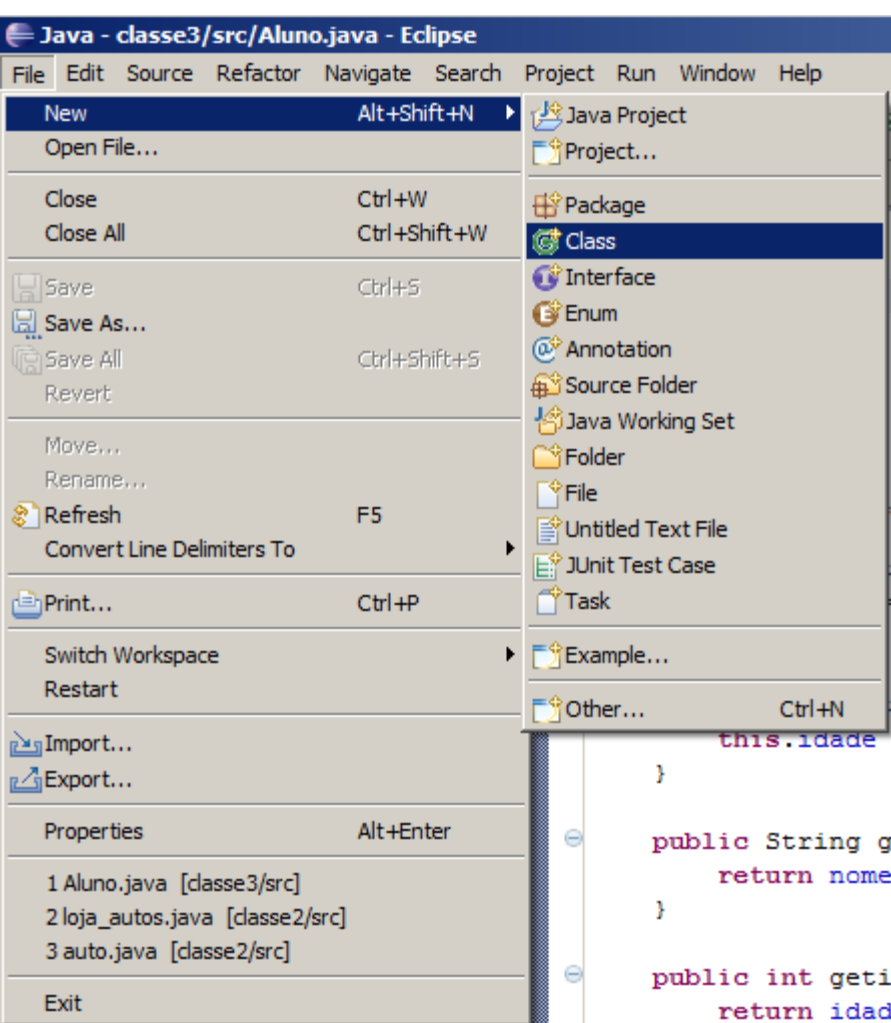
Do you want to add comments? (Configure templates and default value: [ncrc](#))

☐ Generate comments

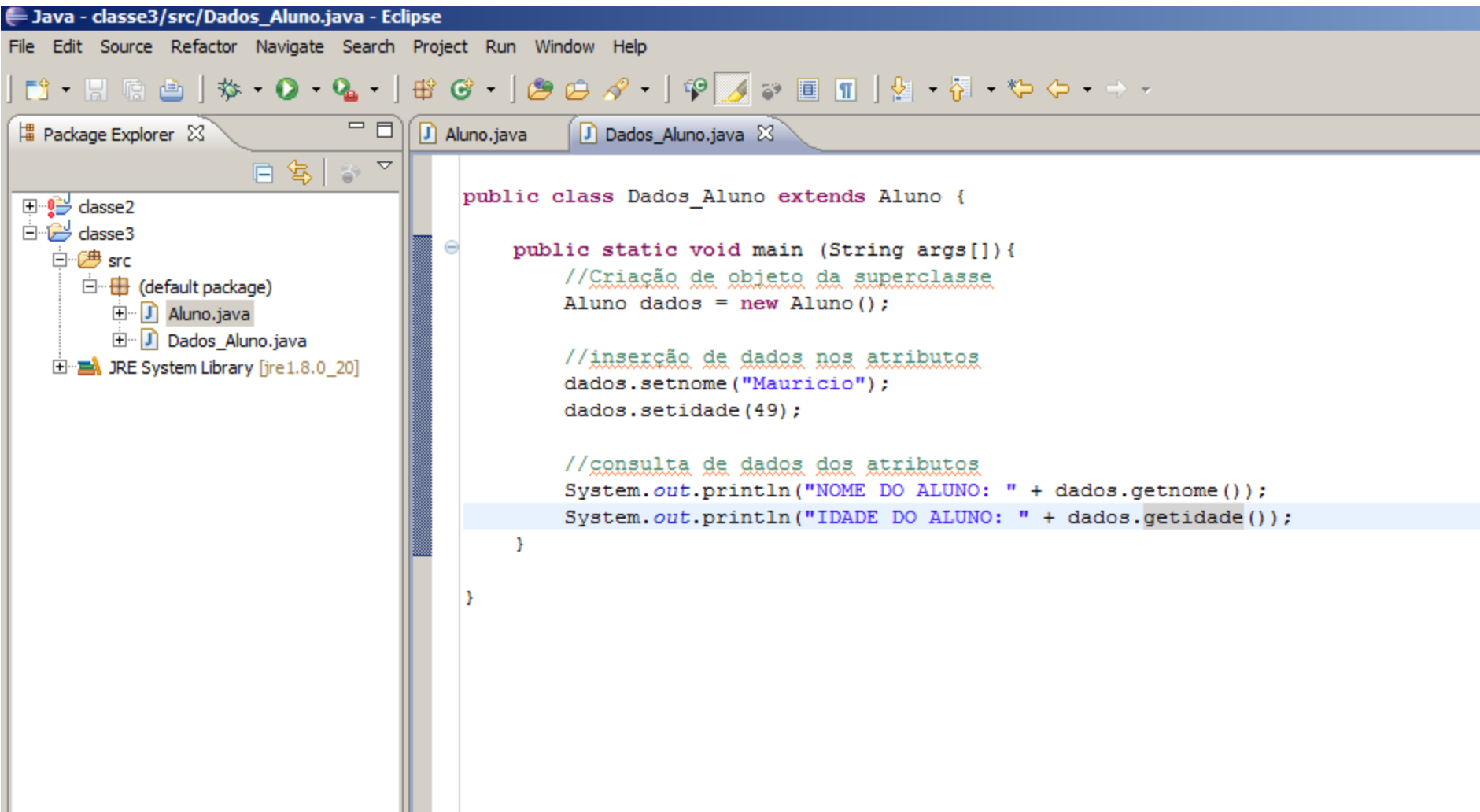
Projeto no Eclipse



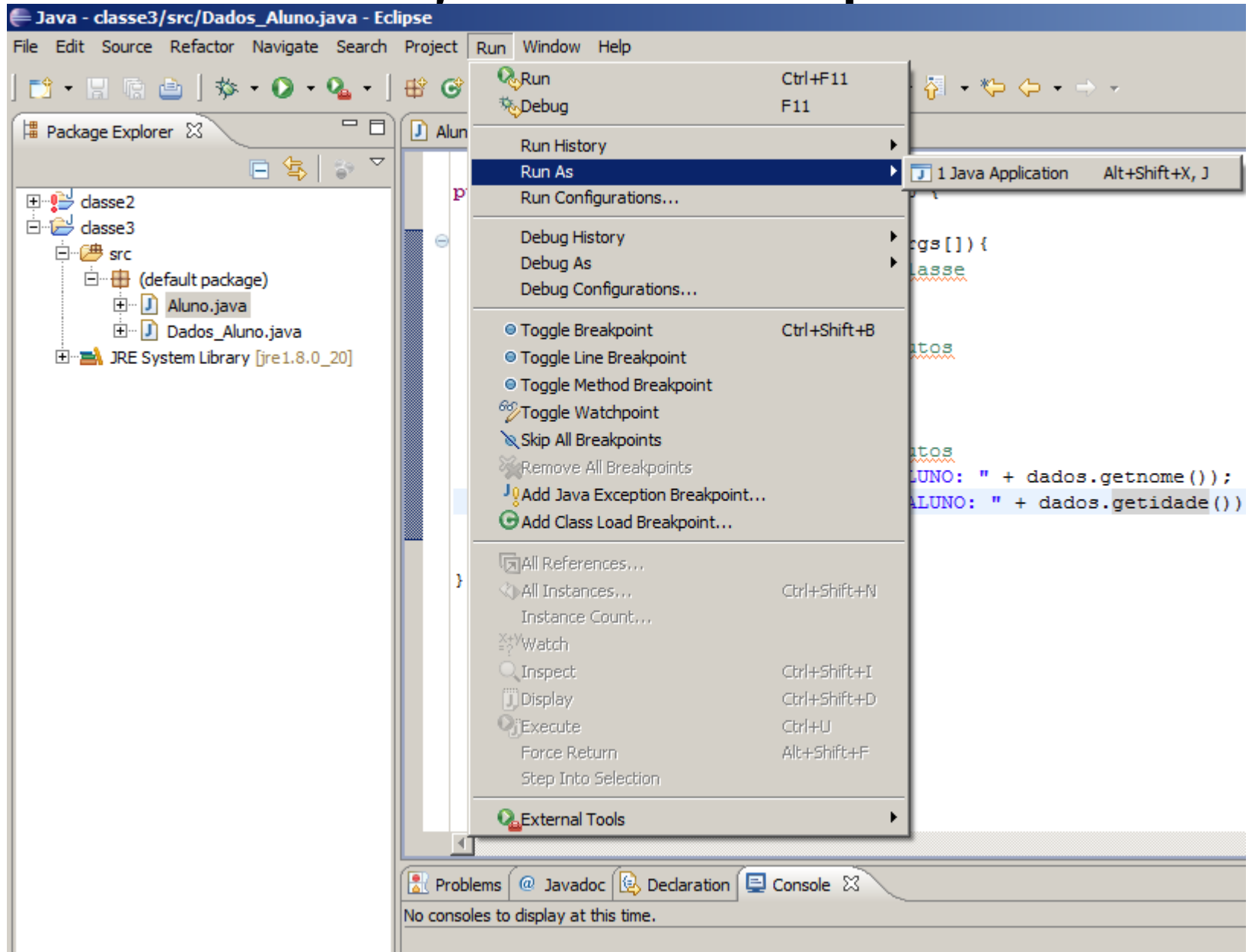
Projeto no Eclipse



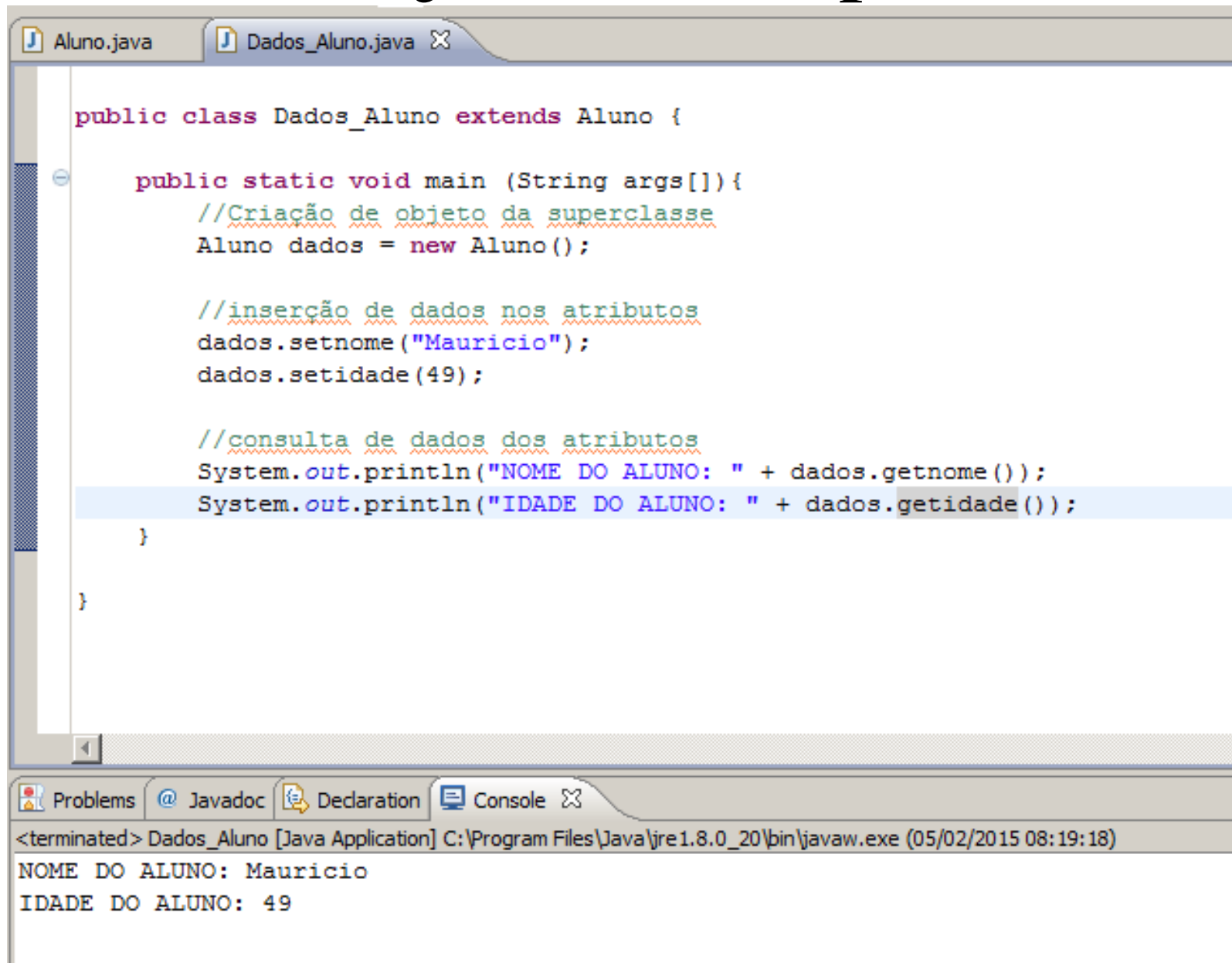
Projeto no Eclipse



Projeto no Eclipse



Projeto no Eclipse



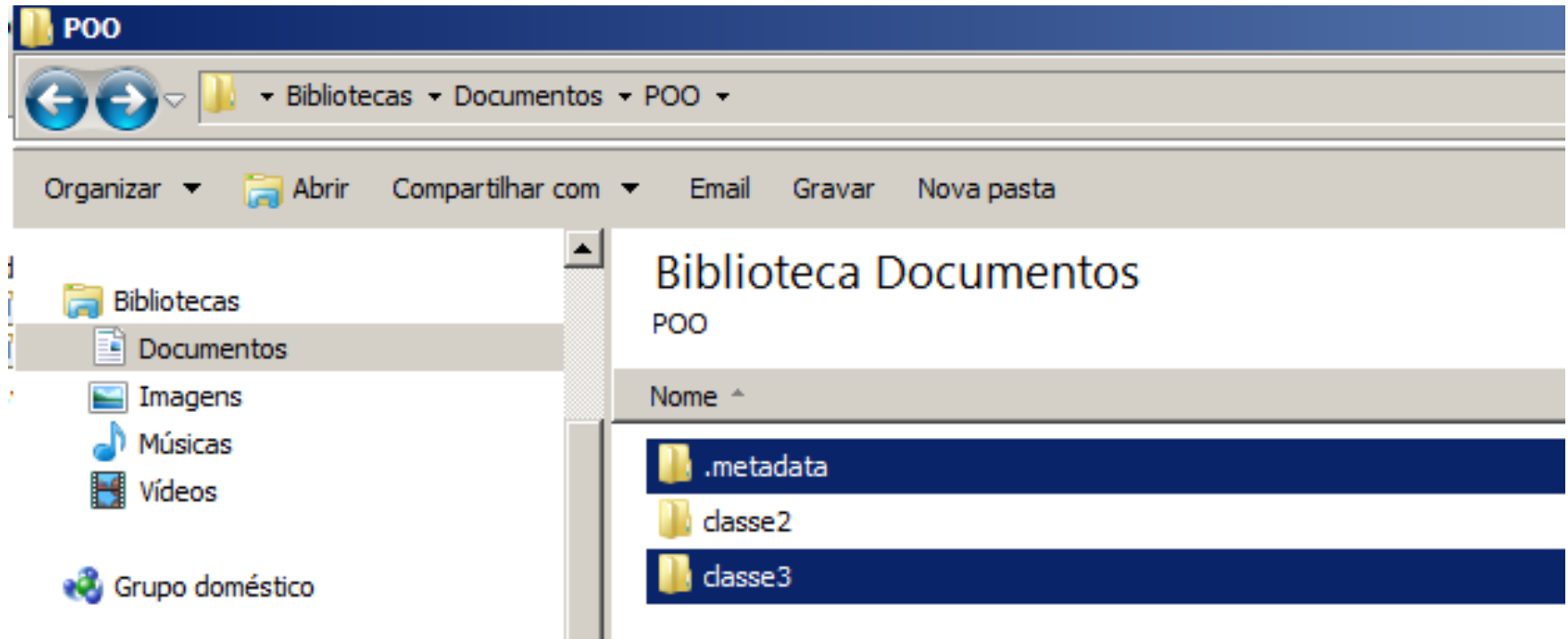
```
public class Dados_Aluno extends Aluno {  
  
    public static void main (String args[]) {  
        //Criação de objeto da superclasse  
        Aluno dados = new Aluno();  
  
        //inserção de dados nos atributos  
        dados.setnome("Mauricio");  
        dados.setidade(49);  
  
        //consulta de dados dos atributos  
        System.out.println("NOME DO ALUNO: " + dados.getnome());  
        System.out.println("IDADE DO ALUNO: " + dados.getidade());  
    }  
}
```

Problems Javadoc Declaration Console

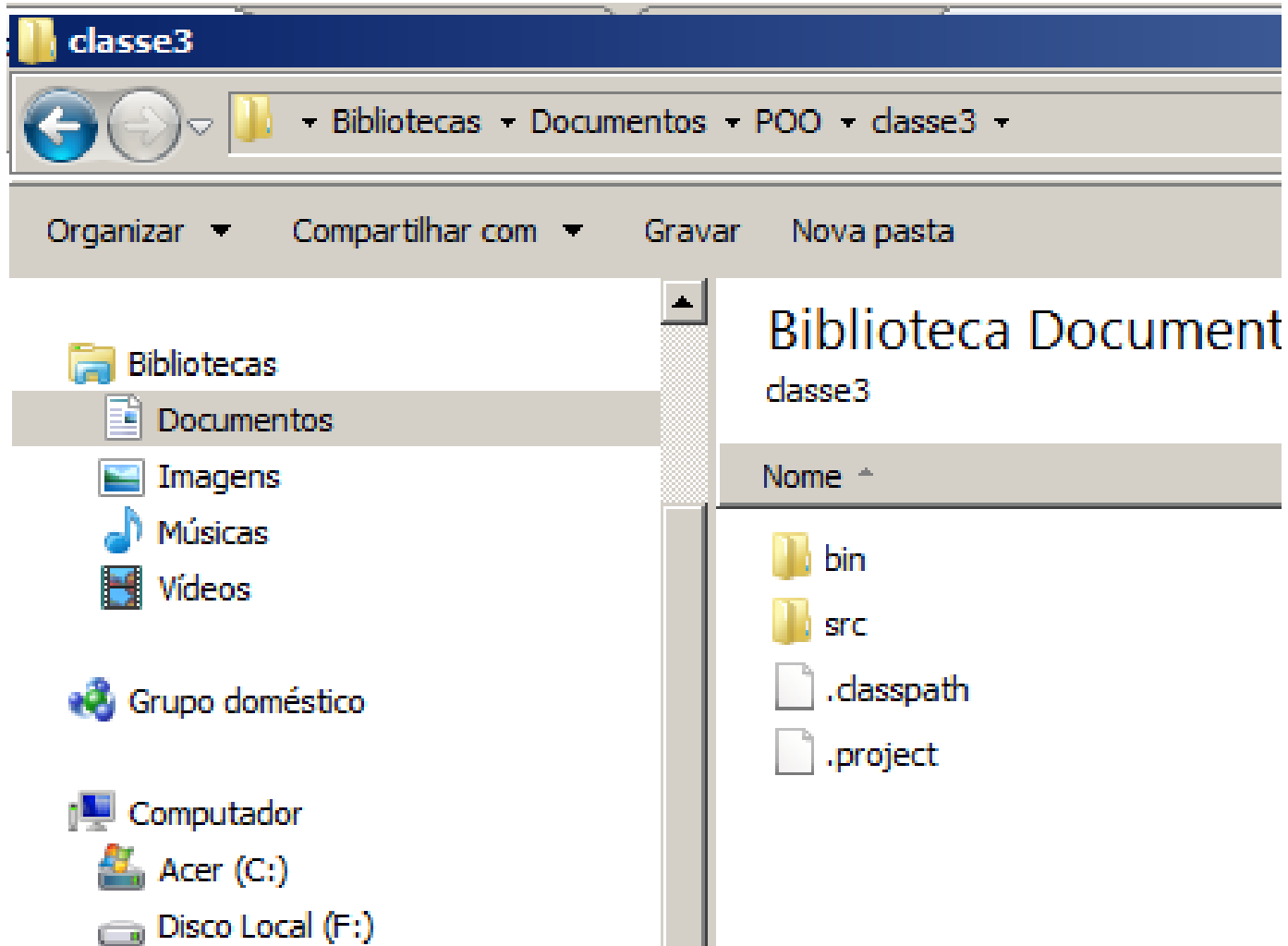
<terminated> Dados_Aluno [Java Application] C:\Program Files\Java\jre1.8.0_20\bin\javaw.exe (05/02/2015 08:19:18)

NOME DO ALUNO: Mauricio
IDADE DO ALUNO: 49

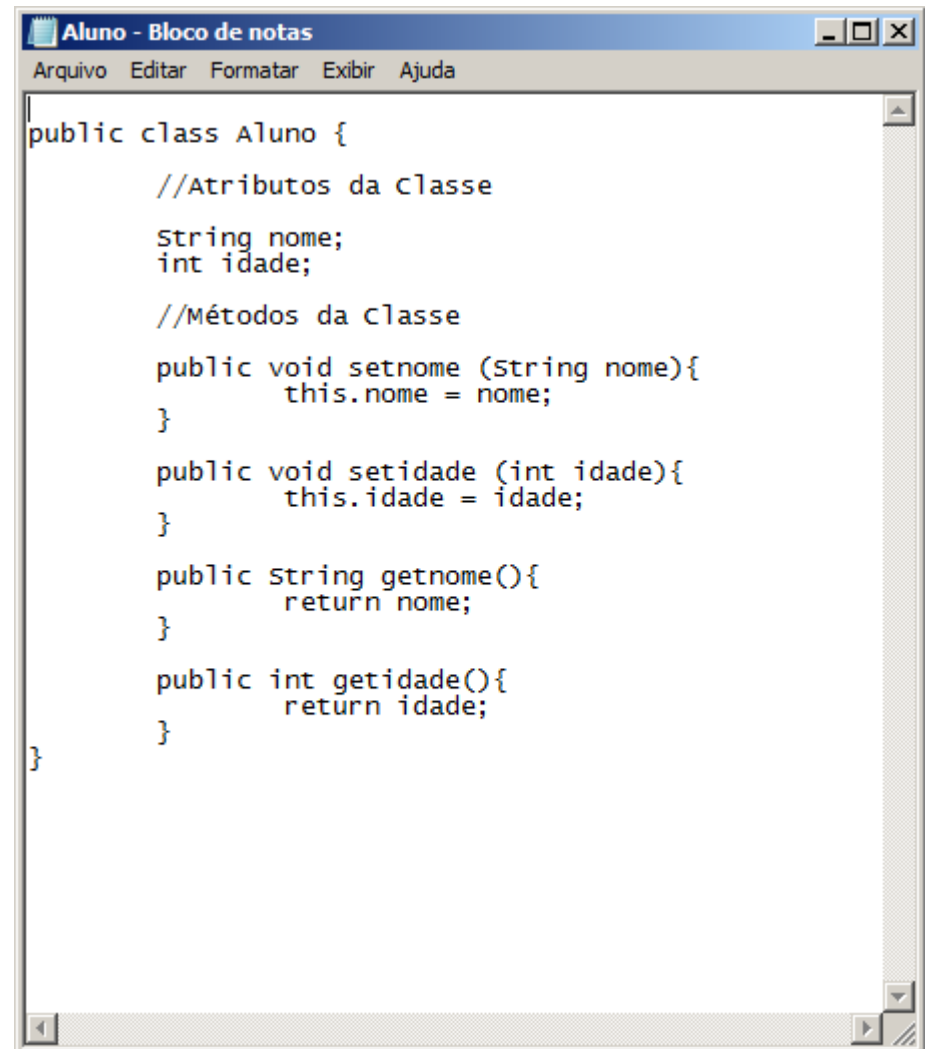
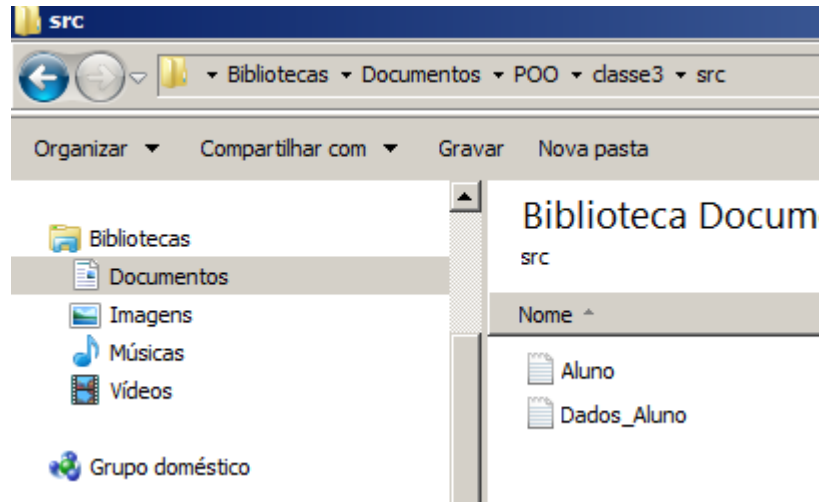
Projeto no Eclipse



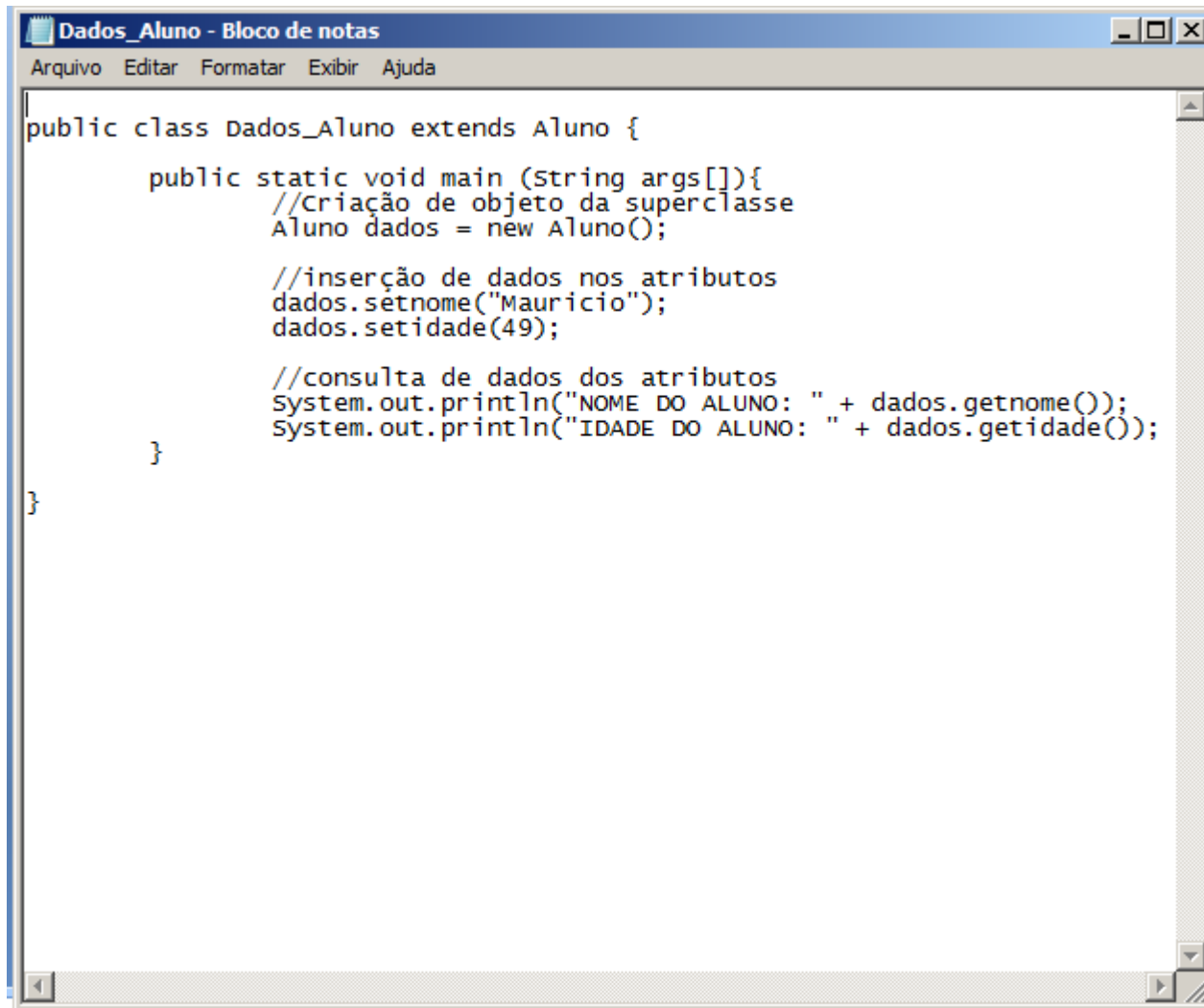
Projeto no Eclipse



Projeto no Eclipse

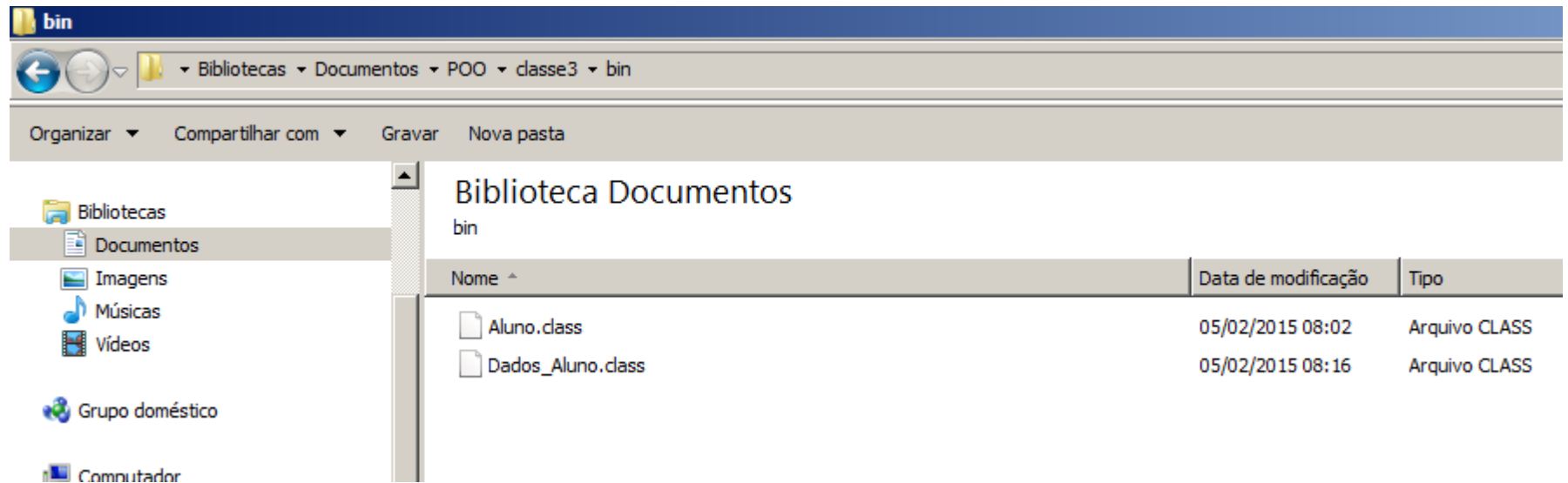


Projeto no Eclipse



```
public class Dados_Aluno extends Aluno {  
    public static void main (String args[]){  
        //Criação de objeto da superclasse  
        Aluno dados = new Aluno();  
  
        //inserção de dados nos atributos  
        dados.setnome("Mauricio");  
        dados.setidade(49);  
  
        //consulta de dados dos atributos  
        System.out.println("NOME DO ALUNO: " + dados.getnome());  
        System.out.println("IDADE DO ALUNO: " + dados.getidade());  
    }  
}
```

Projeto no Eclipse



Exercícios

1. Inserir na classe Aluno o atributo “curso” (o curso que o aluno frequenta na instituição de ensino).
2. Modificar a classe Dados_Aluno para que possa ser inserido e consultado o nome do curso.
3. Criar uma classe “Automovel” com os atributos e métodos set/get.
4. Criar uma classe Dados_Automovel para inserir e consultar dados de um automóvel.

Conceitos de Compilação

Definições

Gramática: determina a forma ou sintaxe das instruções válidas de uma linguagem. É uma descrição formal da sintaxe.

Sintaxe: é o formato das instruções.

Semântica: está relacionada ao significado das instruções.

Exemplo: diferença entre sintaxe e semântica. Dadas as instruções a) e b), onde as variáveis I, J e K são do tipo inteiro, e as variáveis X e Y são do tipo real:

a) $I := J + K$

b) $I := X + Y$

Gramática, sintaxe e semântica:

As duas instruções têm a mesma sintaxe, ou seja, são compostas por um operador de atribuição “:=”, um operador aritmético “+”, e três variáveis, onde a soma de duas é atribuída a uma terceira. As instruções têm semânticas diferentes: em a) a semântica especifica que as variáveis da expressão serão somadas usando operador aritmético de cálculo inteiro, e o resultado será armazenado na variável I. Em b) há uma operação de soma com ponto flutuante, com a conversão do resultado para inteiro antes de ser armazenado na variável I.

Exemplo de notação utilizada para registrar a gramática de linguagens: Backus-Naur Form (BNF). Uma gramática BNF consiste em um conjunto de regras, cada uma das quais definindo a sintaxe de alguma construção de uma linguagem de programação.

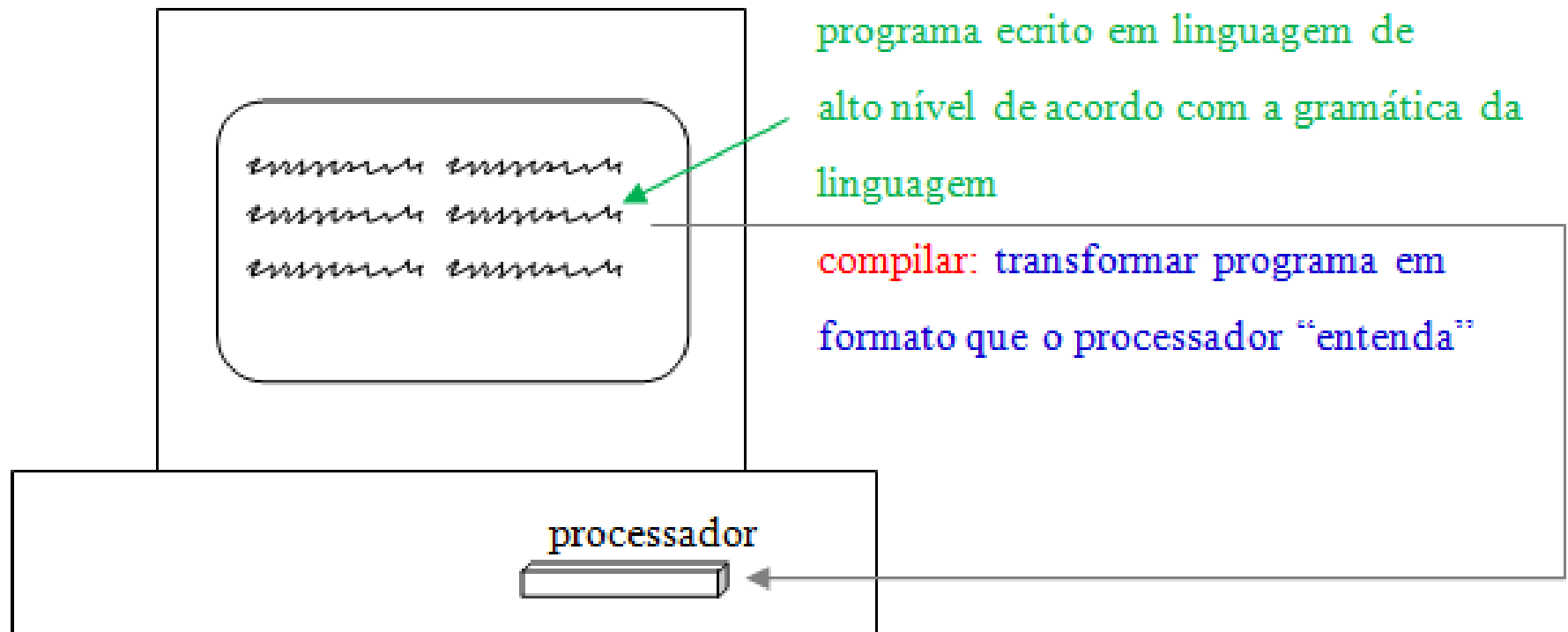
Exemplo: gramática BNF para um subconjunto da linguagem PASCAL (definição da sintaxe de uma instrução READ do PASCAL, identificada na gramática como <read>).

Gramática, sintaxe e semântica:



Conceitos de Compilação

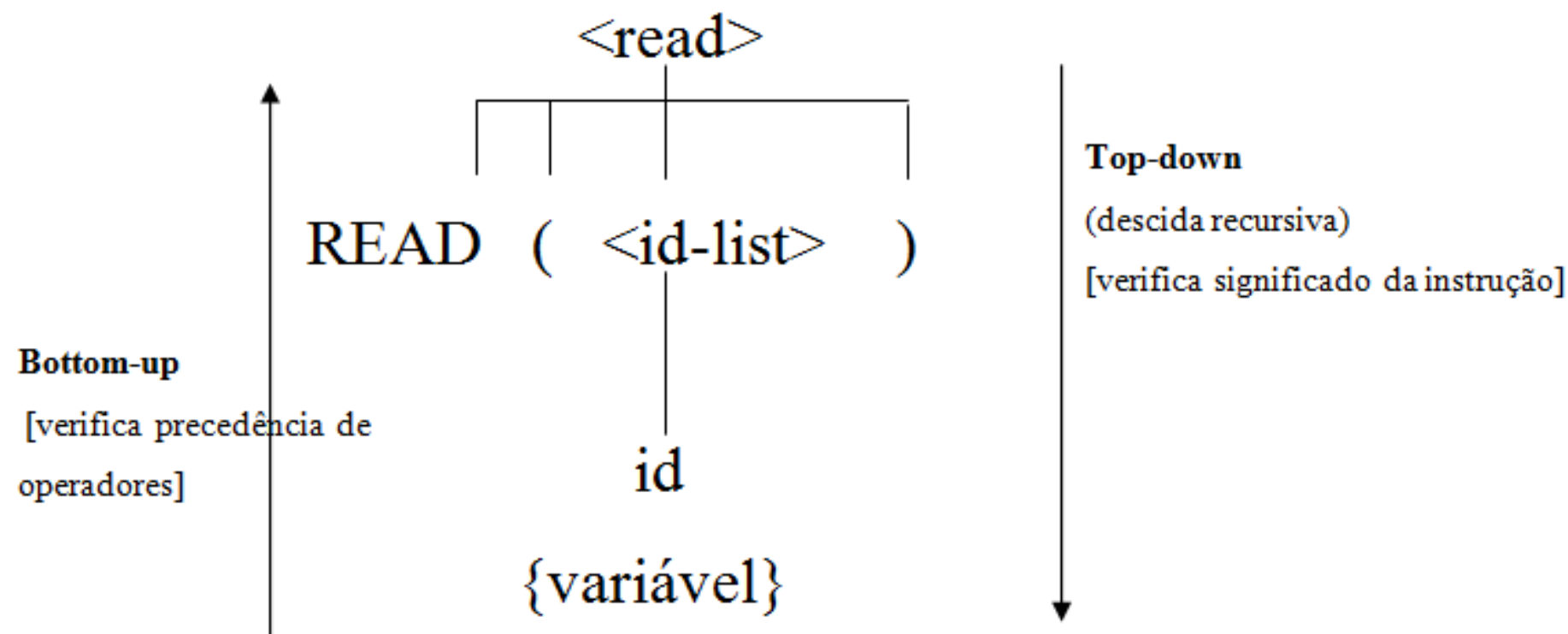
Etapas da Compilação



Etapas da Compilação:

Análise Léxica: leitura do programa → reconhecimento dos tokens (cada token tem um código numérico equivalente). Ferramenta: SCANNERS ou ANALISADORES LÉXICOS (software).

Análise Sintática: Após análise dos tokens, cada instrução deve ser reconhecida como uma das construções válidas da gramática. Ferramenta: PARSER. → construção da Árvore Sintática ou Árvore de Parsing.



Etapas da Compilação:

Geração do Código: após o reconhecimento da instrução como sendo pertencente à gramática da linguagem, é executada uma rotina semântica para a geração do código (código objeto), que depois é transformado em linguagem Assembly (linguagem de baixo nível, de máquina, de acordo com o processador). A linguagem Assembly, quando compilada, gera um arquivo binário de “0” e “1”, onde o zero equivale a um nível lógico digital = zero (tensão de 0V) e o um equivale a um nível lógico digital = um (tensão de $\cong 5V$). O processador só reconhece uma sequência de sinais digitais.

Etapas da Compilação:

Exemplo:

