



**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
RIO GRANDE DO SUL
Câmpus Feliz

Java

Prof. Moser Fagundes

Programação II

Sumário

- O que é o Java?
- Eclipse
- Programando em Java: primeiros passos
- Exercícios
- Instalação do Java e do Eclipse

Java

Histórico

- Em 1991, foi realizado o projeto Green na Sun Microsystems, resultando em uma linguagem baseada em C/C++.
- Linguagem inicialmente chamada Oak, porém o nome foi mudado para Java (café) pois já existia uma linguagem chamada Oak.
- O Java ganhou notoriedade em 1995 com o crescimento da World Wide Web.

Java

Versões

- 1.0 (1996)
- 1.1 (1997)
- 1.2 (1998)
- 1.3 (2000)
- 1.4 (2002)
- Java 5 (2004), Java 6 (2006), Java 7 (2011)
- **Java 8 (2014) [ainda suportada]**
- Java 9 (2017)
- **Java 10 (2018)**

Java

Bibliotecas

- Os programas Java são compostos por **classes**, e estas por sua vez, possuem **métodos** que realizam tarefas.
- Os programadores tiram proveito através do **reuso** de tais classes, as quais são agrupadas em **bibliotecas**.
 - Tais **bibliotecas** são acessadas por meio de **APIs** (*Application Programming Interfaces*).

Java

Java Virtual Machine (JVM):

- Fornece um conjunto de especificações de plataforma de hardware para qual todo o código Java é compilado;
- Interpreta *bytecodes* (que são independentes de plataforma de hardware);
- Torna a linguagem Java portátil para diversas plataformas;
- A JVM pode ser implementada na forma de software ou em hardware.

Java

Garbage Collector:

- Toda a alocação de memória exige a sua liberação, e nas linguagens de programação tradicionais essa liberação é responsabilidade do programador;
- Através do **garbage collection** é verificada e liberada automaticamente a memória que não está sendo utilizada;
- O programador Java não precisa se preocupar com a alocação e desalocação de memória.

Eclipse

Eclipse é um **IDE** para desenvolvimento **Java**, porém suporta várias outras linguagens, incluindo Python.

Está instalado na pasta de **Arquivos de Programa**, na subpasta **eclipse**.

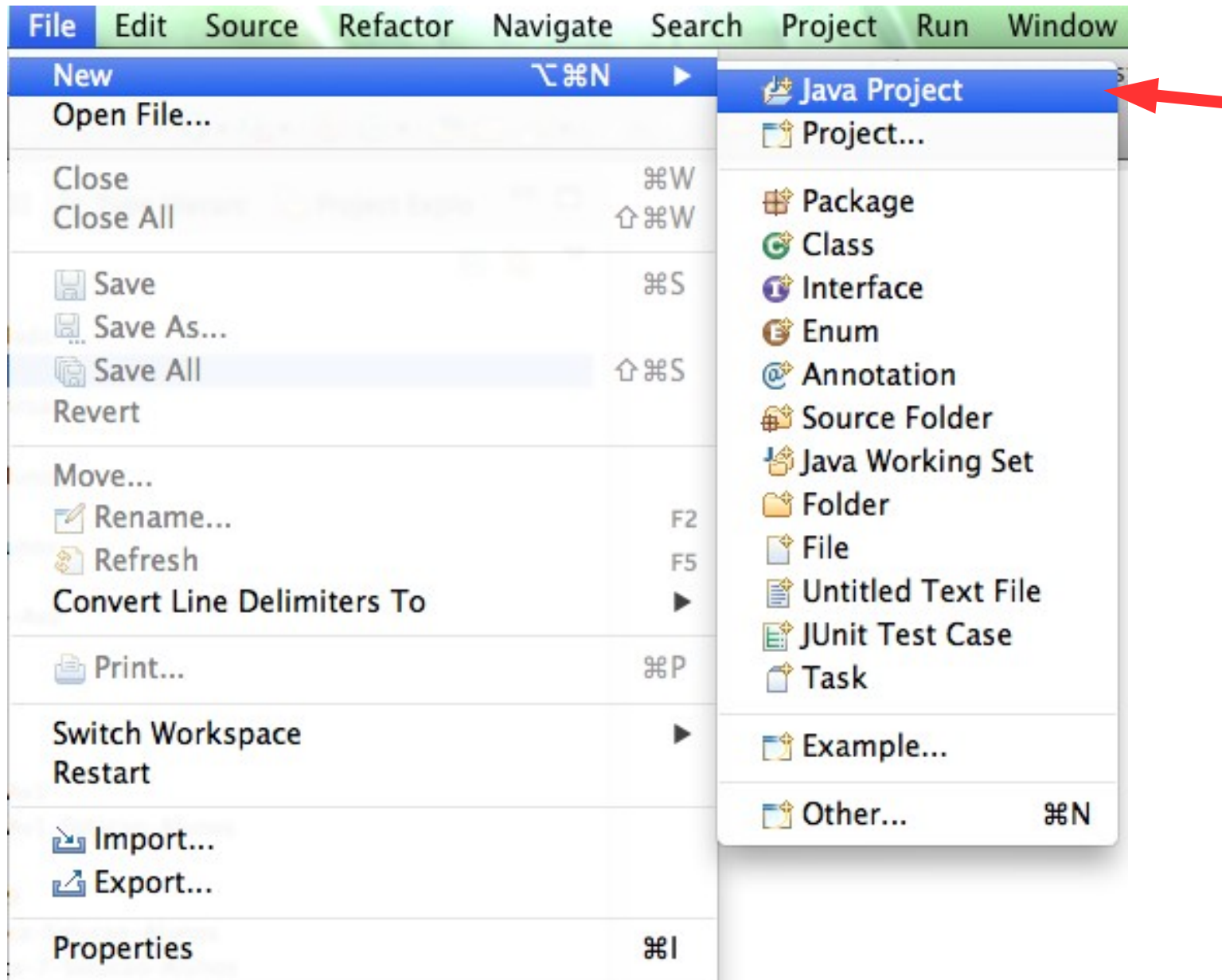


Um programa em Java

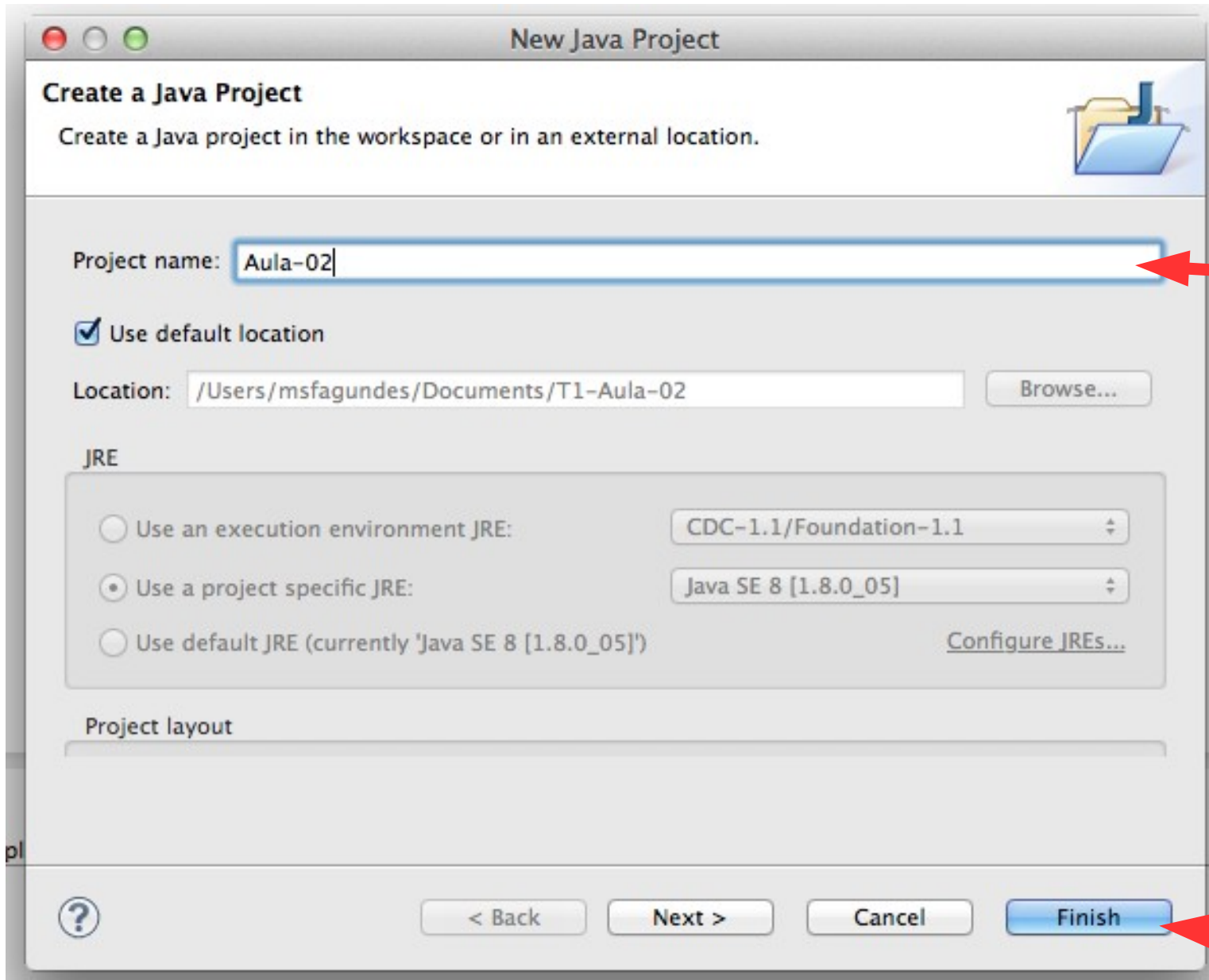
Para criar um programa, vamos necessitar os seguintes elementos:

- Projeto Java no Eclipse
- Um pacote
- Uma classe

Criando um projeto Java



Criando um projeto Java



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

Project name:

☒ Use default location

Location:

JRE

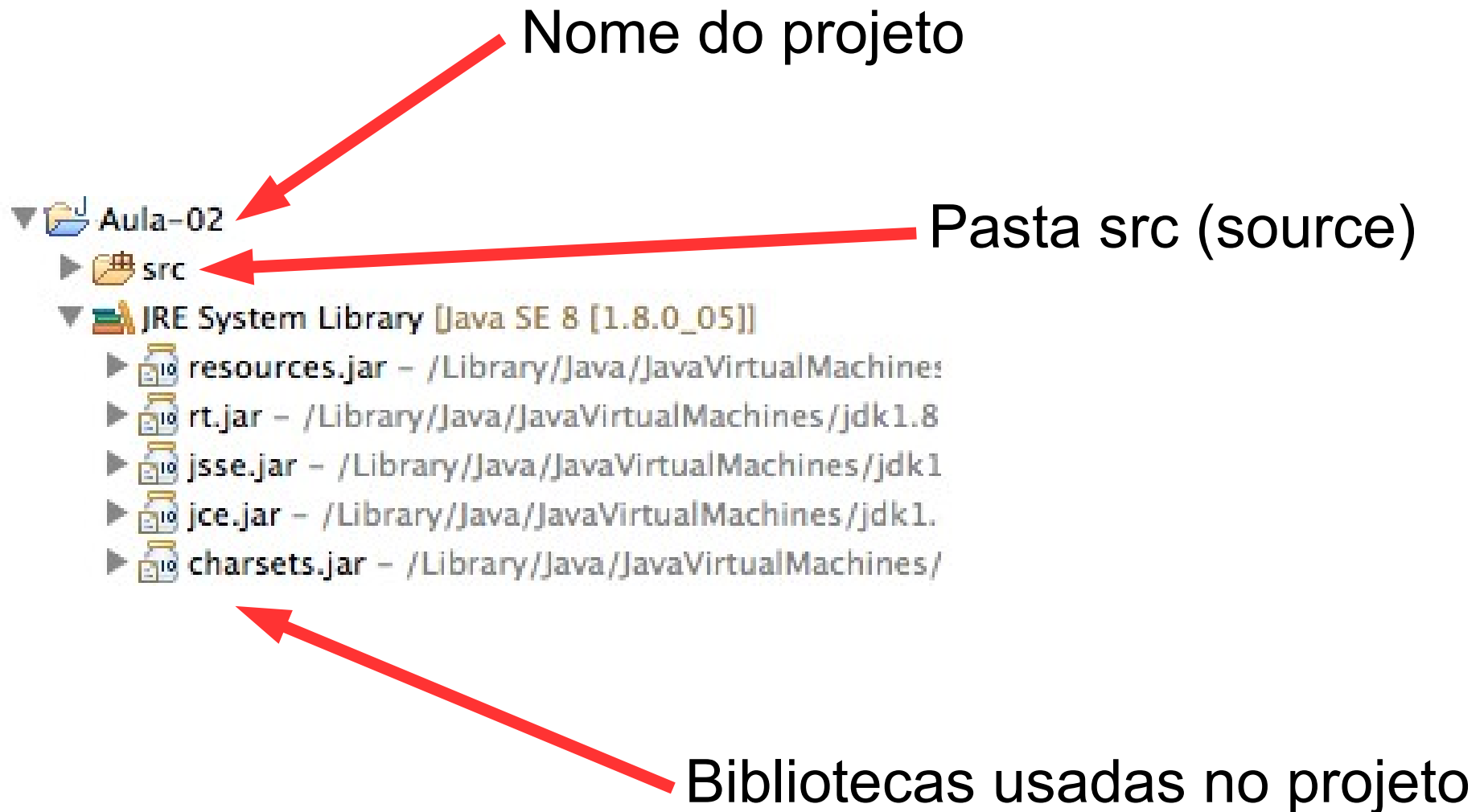
☐ Use an execution environment JRE:

☒ Use a project specific JRE:

☐ Use default JRE (currently 'Java SE 8 [1.8.0_05]') [Configure JREs...](#)

Project layout

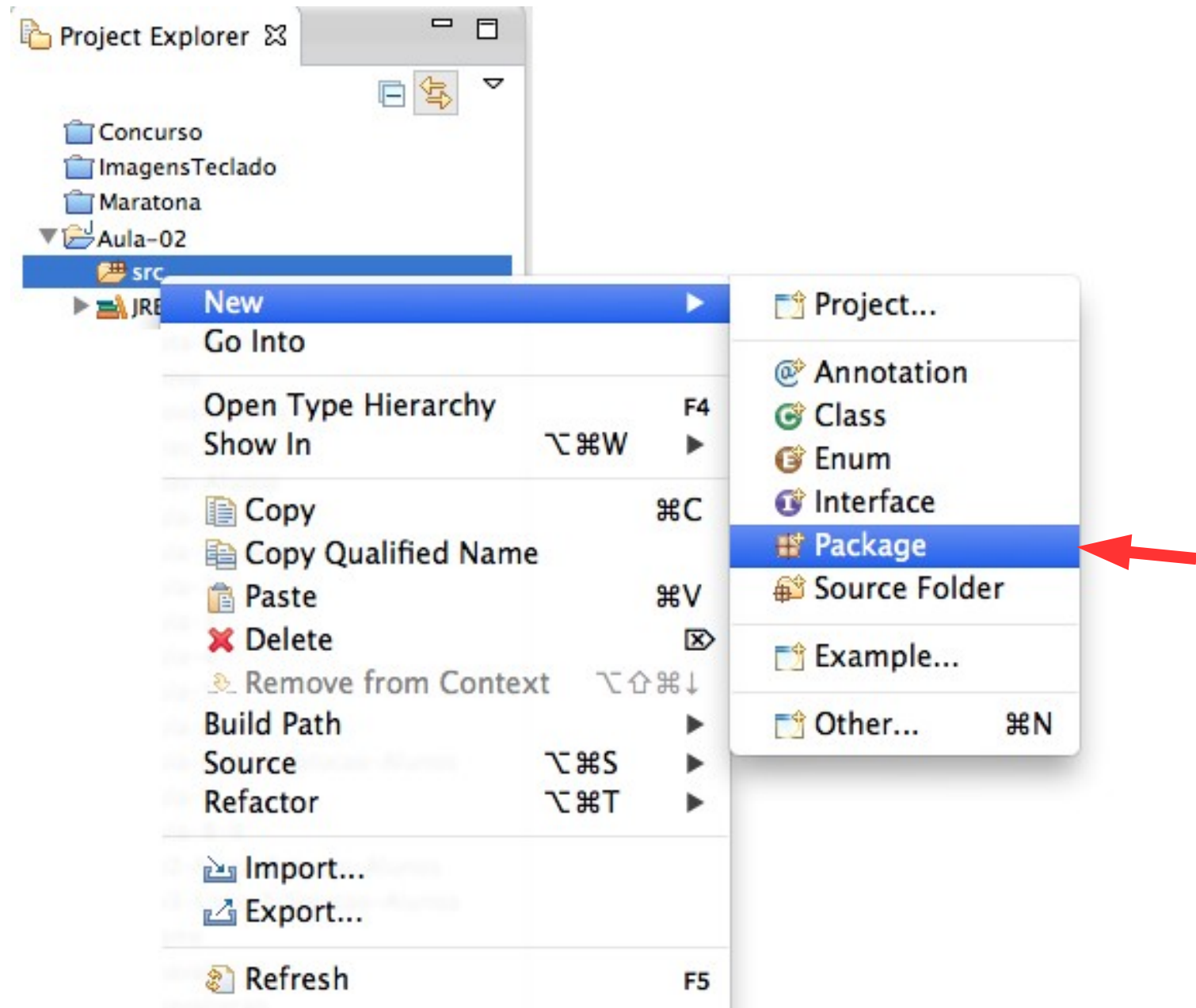
Estrutura de um projeto



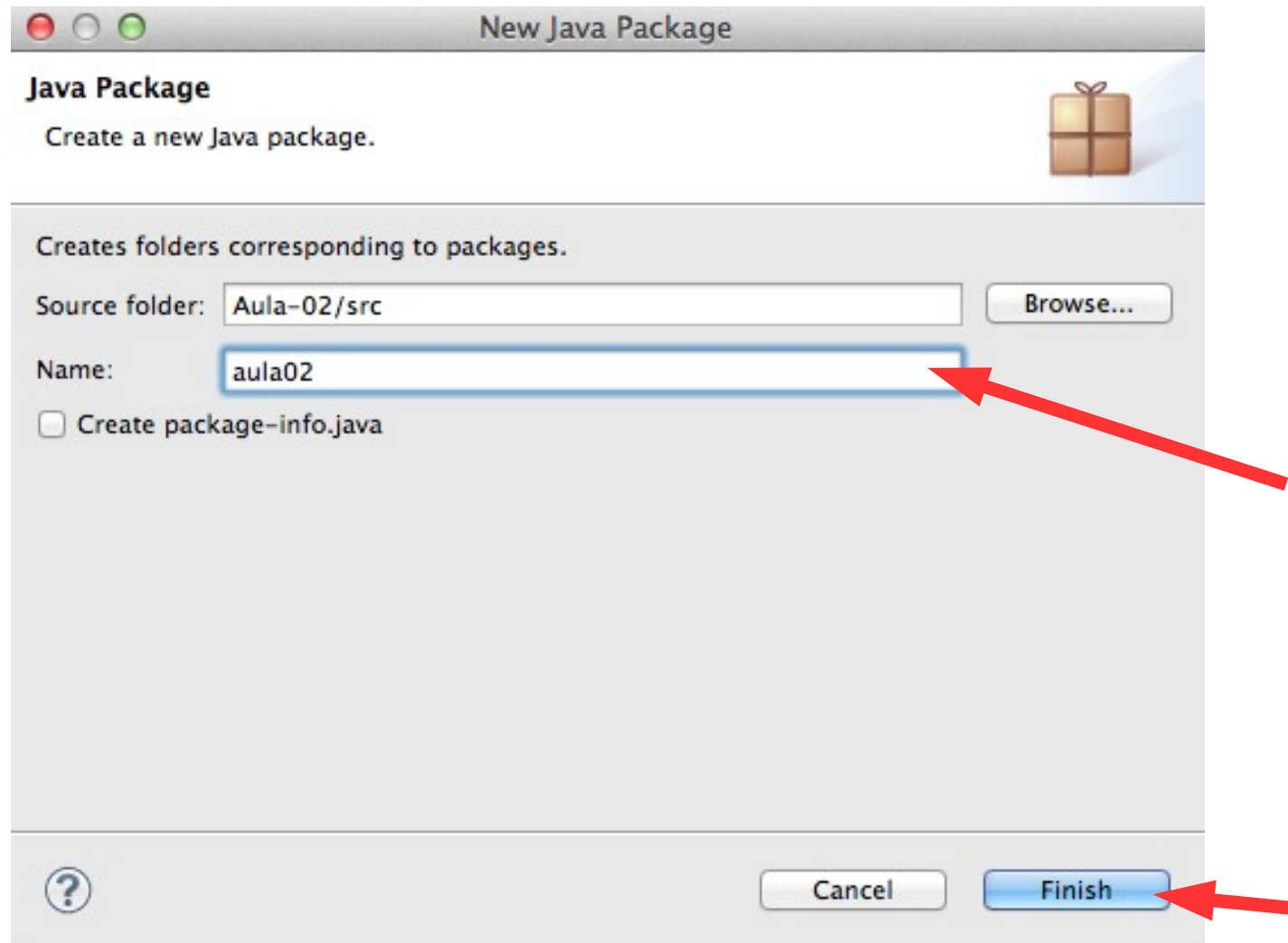
Pacotes

Os pacotes no Java, como o próprio nome sugere, são **estruturas** criadas para dividir o código fonte, assim como promover a separação de classes.

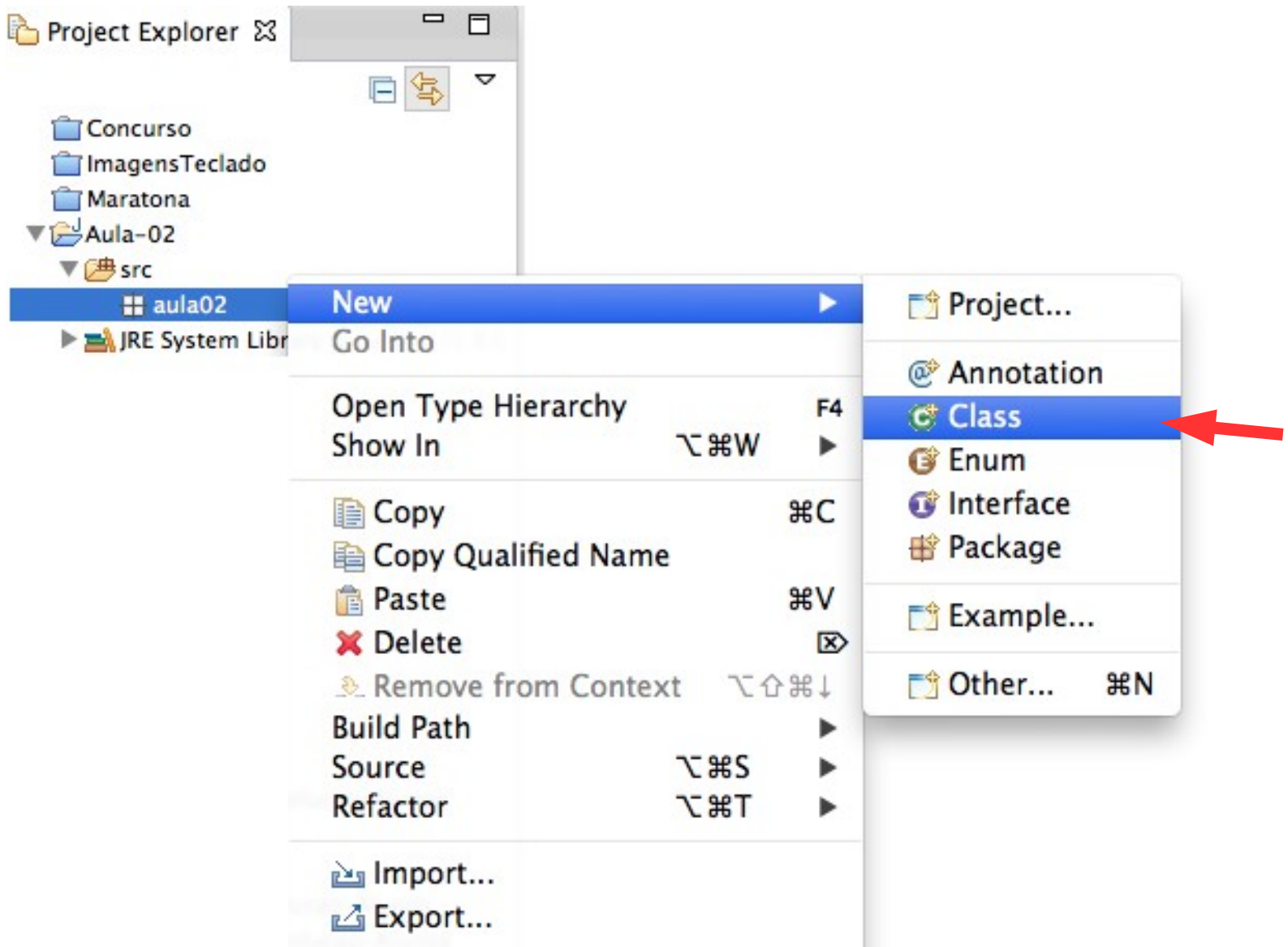
Criando um pacote



Criando um pacote



Criando uma classe



Criando uma classe

Java Class
Create a new Java class.

Source folder:

Package:

☐ Enclosing type:

Name:

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

Superclass:

Interfaces:

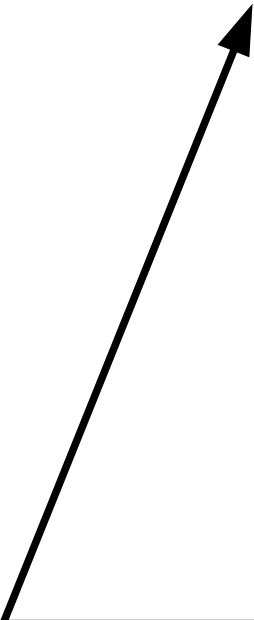
Which method stubs would you like to create?

☒ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

Primeiro programa

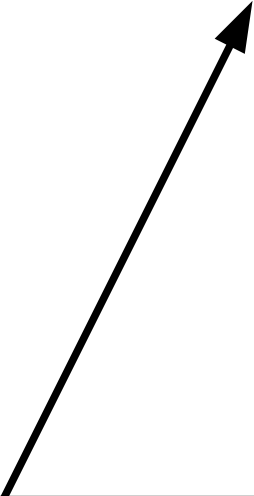
```
package aula02;  
  
public class PrimeiroPrograma {  
  
    public static void main(String args[]) {  
        System.out.println("Ola Mundo!");  
    }  
}
```



A classe inicia com o nome do **pacote** a qual pertence.

Primeiro programa

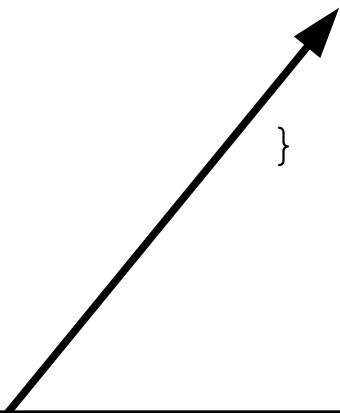
```
package aula02;  
  
public class PrimeiroPrograma {  
    public static void main(String args[]) {  
        System.out.println("Ola Mundo!");  
    }  
}
```



Um programa em Java sempre terá ao menos uma **classe**. Após a palavra **class**, colocamos o nome da classe. Neste caso o nome da classe é **PrimeiroPrograma**. Nomes de classes SEMPRE começam com letras MAIÚSCULAS.

Primeiro programa


```
package aula02;  
  
public class PrimeiroPrograma {  
  
    public static void main(String args[]) {  
        System.out.println("Ola Mundo!");  
    }  
}
```



Todo programa inicia pela função/ método principal **main()**.
Nomes de funções/métodos SEMPRE começam com letras minúsculas.

Primeiro programa

Os parâmetros **args[]** permitem que uma função/método inicie com valores recebidos externamente.



```
package aula02;

public class PrimeiroPrograma {

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

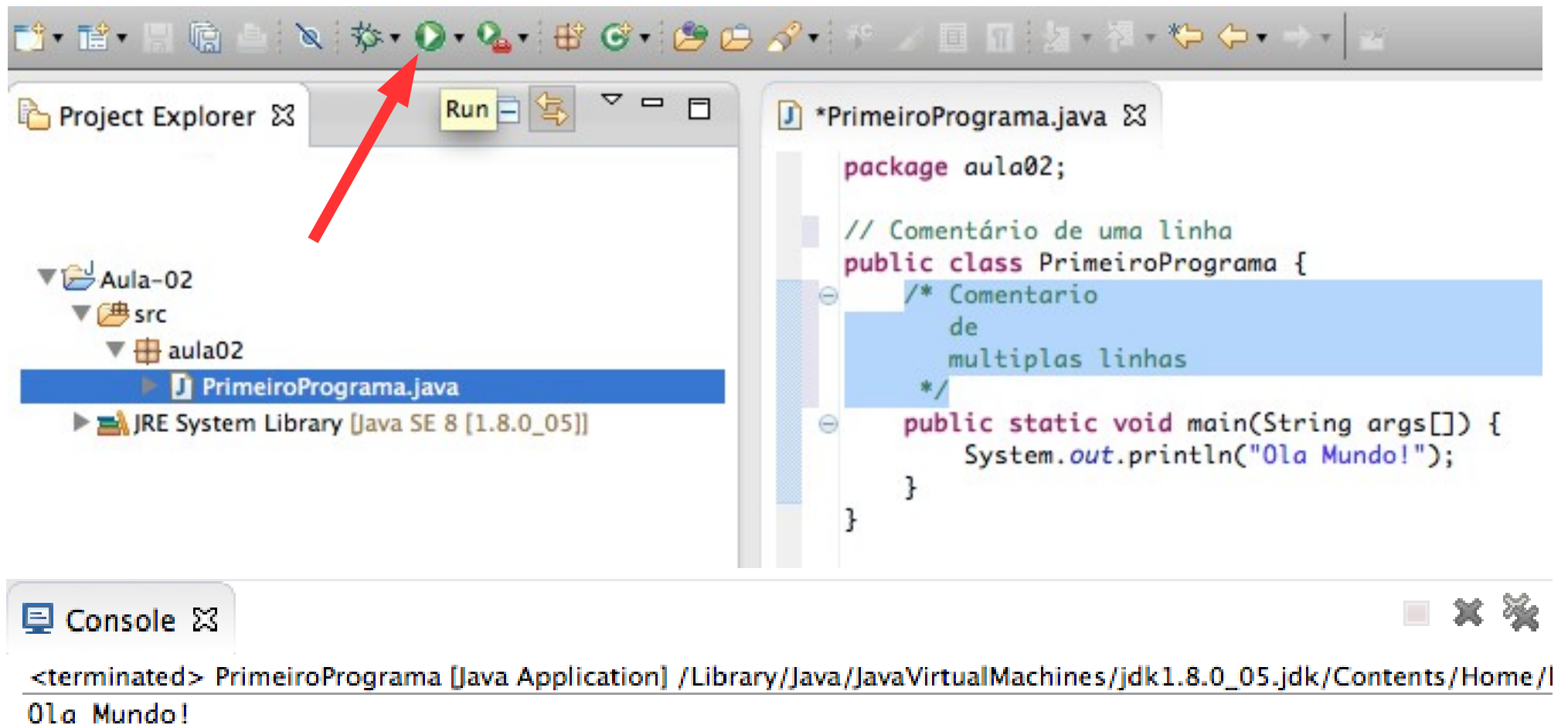
Primeiro programa

```
package aula02;  
  
public class PrimeiroPrograma {  
  
    public static void main(String args[]) {  
        System.out.println("Ola Mundo!");  
    }  
}
```

Os comandos Java terminam com ponto e virgula ;

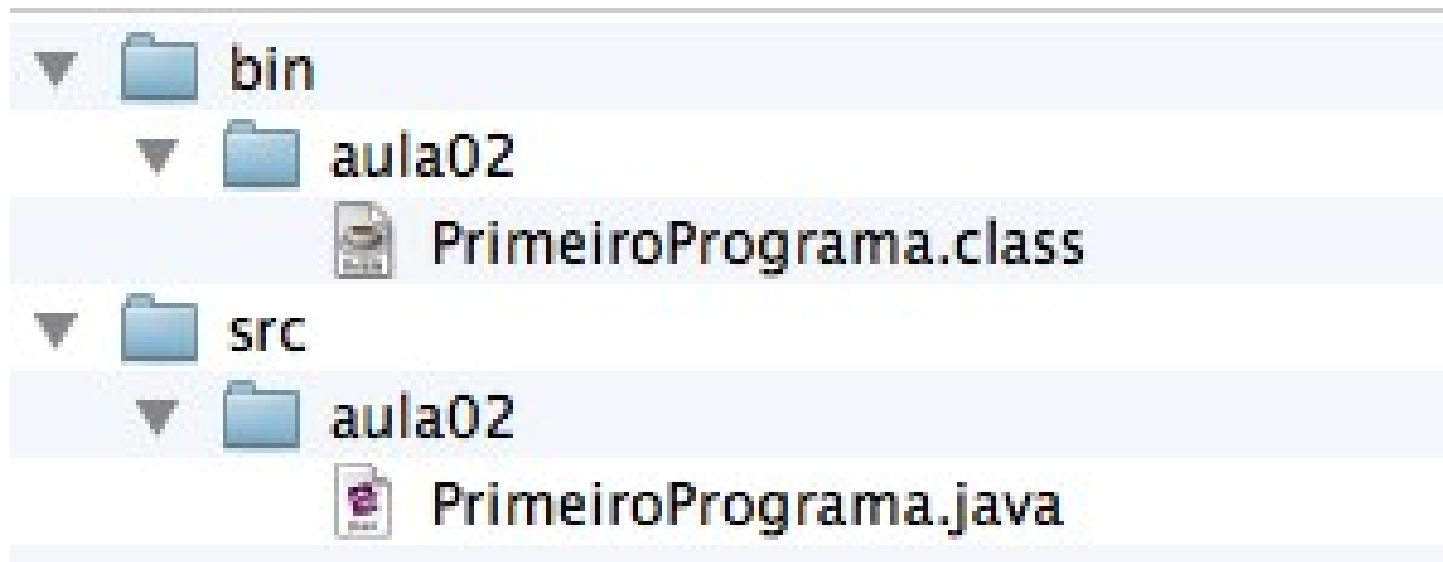
Esta linha imprime na saída padrão a String "Olá Mundo!". No nosso caso, a saída é o Console do Eclipse.

Primeiro programa – Execução



Estrutura do projeto – Arquivos

Para encontrar os arquivos dos seus projetos no seu computador, localize o diretório do seu **workspace**.



Múltiplas linhas

```
package aula02;

public class PrimeiroPrograma {

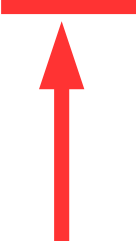
    public static void main(String args[]) {
        System.out.println("Ola Mundo!");
        System.out.println("Hello World!");
        System.out.println(";Hola Mundo!");
    }
}
```

Várias impressões na mesma linha

```
package aula02;

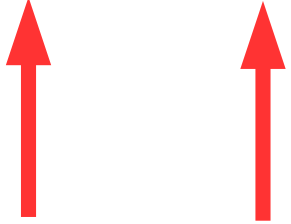
public class PrimeiroPrograma {

    public static void main(String args[]) {
        System.out.print("Ola ");
        System.out.print("Mundo ");
        System.out.print("Cruel!");
    }
}
```



Várias linhas na mesma instrução

```
package aula02;  
  
public class PrimeiroPrograma {  
  
    public static void main(String args[]) {  
        System.out.println("Ola\nMundo\nCruel!");  
    }  
}
```



O comando `\n` faz a quebra de linha em uma String

Caixa de diálogo

```
package aula02;

import javax.swing.JOptionPane;

public class PrimeiroDialog {

    public static void main(String args[]) {

        JOptionPane.showMessageDialog(null, "Ola Mundo");

    }
}
```

Caixa de diálogo com múltiplas linhas

```
package aula02;

import javax.swing.JOptionPane;

public class PrimeiroDialog {

    public static void main(String args[]) {

        JOptionPane.showMessageDialog(null, "Ola\nMundo");

    }
}
```

Comentários

```
package aula02;

// Comentário de uma linha
public class TestandoComentarios {
    /* Comentário
    de
    múltiplas linhas
    */
    public static void main(String args[]) {
        System.out.println("Comments!");
    }
}
```

Tipos de dados

Tipos de dados primitivos do Java:

- **Inteiros:** `byte` / `short` / `int` / `long`
- **Reais:** `float` / `double`
- **Outros:** `char` / `boolean`

Texto: `String` (vamos estudar mais adiante...)

Nomes de variáveis:

- Não pode ser palavra-chave
- Não pode ser nome de outra variável ou classe

Tipos de dados

Exemplo de código com inteiros:

```
package aula02;

public class TesteDeInteiros {

    public static void main(String args[]) {
        byte n1 = 10;
        short numero2 = 70;
        int i = 0;
        long var4 = 10000000;

        System.out.println(n1);
        System.out.println(numero2);
        System.out.println(i);
        System.out.println(var4);

        // Quais são as faixas de valores que cada tipo aceita?
    }
}
```


Tipos de dados

Exemplo de código com reais:

```
package aula02;

public class TesteDeReais {

    public static void main(String args[]) {

        float f1 = 19.9f;
        double var4 = 3.1415;

        System.out.println(f1);
        System.out.println(var4);

        // Quais são as faixas de valores que cada tipo aceita?
    }
}
```

Tipos de dados

Exemplo de código com boolean e char:

```
package aula02;

public class TesteCharBoolean {

    public static void main(String args[]) {

        boolean a = true;
        boolean b = false;
        char c = 'A';

        System.out.println(a);
        System.out.println(b);
        System.out.println(c);
    }
}
```

Operadores

Operadores

+	adição
-	subtração
*	multiplicação
/	divisão
%	resto

Operadores de igualdade

x == y	x é igual a y
x != y	x é diferente y

Operadores

Operadores relacionais

x<y x é menor que y

x>y x é maior que y

x<=y x é menor ou igual a y

x>=y x é maior ou igual a y

Operadores lógicos

&& e (and)

|| ou (or)

! negação (not)

Soma de inteiros

```
package aula02;

public class SomaInteiros {

    public static void main(String args[]) {

        int i1 = 25;
        int i2 = 75;

        int soma = i1 + i2;

        System.out.println(soma);
    }
}
```

Divisão de reais

```
package aula02;

public class DivisaoReais {

    public static void main(String args[]) {

        double d1 = 333.333;
        double d2 = 123.99;

        double valor = d1 / d2;

        System.out.println(valor);
    }
}
```

IF com operadores

```
package aula02;

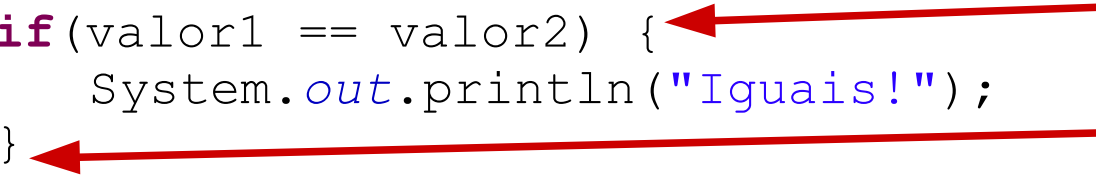
public class TesteIF {

    public static void main(String args[]) {
        int valor1 = 25;
        int valor2 = 75;

        if(valor1 == valor2) {
            System.out.println("Iguais!");
        }

        if(valor1 != valor2) {
            System.out.println("Diferentes!");
        }
    }
}
```

Usamos chaves para
determina o início e o
final do bloco do IF



IF com operadores

Altere o programa do slide anterior para que ele escreva somente o maior dos dois números.

IF com operadores

```
package aula02;

public class TesteIF {

    public static void main(String args[]) {
        int valor1 = 25;
        int valor2 = 75;

        if(valor1 == valor2) {
            System.out.println("Iguais!");
        }

        if((valor1 > valor2) || (valor1 < valor2)) {
            System.out.println("Diferentes!");
        }
    }
}
```

IF e ELSE

```
package aula02;

public class TesteIF {

    public static void main(String args[]) {
        int valor1 = 25;
        int valor2 = 75;

        if(valor1 == valor2) {
            System.out.println("Iguais!");
        } else {
            System.out.println("Diferentes!");
        }
    }
}
```

Leitura de inteiro do teclado

```
package aula02;

import java.util.Scanner;

public class LeitorInteiro {

    public static void main(String args[]) {

        Scanner leitor = new Scanner(System.in);
        System.out.print("Digite um numero: ");
        int valor = leitor.nextInt();
        System.out.println("O valor digitado foi " + valor);
        leitor.close();
    }
}
```

Incremento / decremento

```
package aula02;

public class TesteIncremento {

    public static void main(String[] args) {

        int i = 0;
        i++;
        System.out.println(i);

        i++;
        i++;
        System.out.println(i);

        i--;
        System.out.println(i);
    }
}
```

for

```
package aula02;

public class TesteFor {

    public static void main(String[] args) {

        for(int i=0; i < 10; i++) {
            System.out.println(i);
        }
    }
}
```

for

```
package aula02;

public class TesteFor {

    public static void main(String[] args) {

        for(int i=10; i > 0; i--) {
            System.out.println(i);
        }
    }
}
```

while

```
package aula02;

public class TesteWhile {

    public static void main(String[] args) {

        int x = 0;

        while(x <= 10) {
            System.out.println(x);
            x++;
        }
    }
}
```

while

```
package aula02;

public class TesteWhile {

    public static void main(String[] args) {

        int x = 10;

        while(x > 0) {
            System.out.println(x);
            x--;
        }
    }
}
```


do ... while

```
package aula02;

public class TesteDoWhile {

    public static void main(String[] args) {

        int x = 0;

        do {
            System.out.println(x);
            x++;
        } while (x<10);
    }
}
```

Pergunta

Qual é a principal diferença entre o laço **while** e o laço **do ... while** ?

switch

```
package aula02;

public class TesteSwitch {

    public static void main(String[] args) {
        char letra = 'a';
        switch(letra) {
            case 'a':
                System.out.println("Digitou a!");
                break;
            case 'b':
                System.out.println("Digitou b!");
                break;
            default:
                System.out.println("Digitou outra coisa!");
                break;
        }
    }
}
```

Teste

Troque o valor da variável `letra`

Pergunta

O que acontece se removemos os `break` ?

Java

Dicas:

- Sempre digitar o par de chaves { } juntas.
- Indentar o código usando TAB.
- Java diferencia maiúsculas e minúsculas.
- Nome de classe sempre começar por maiúscula.
- Nome de variável sempre começa por minúscula.
- Nome de pacote sempre começa por minúscula.

Instalação do JavaSE e Eclipse

Antes de instalar o Eclipse no nosso computador, temos que instalar o Java SE:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk10-downloads-4416644.html>

Após instalar o Java SE, instale o Eclipse:

<http://www.eclipse.org/downloads/>

Há diversas versões do Eclipse para diversas linguagens: certifique-se que você está baixando o **Eclipse IDE for Java Developers**.

Exercícios

- Exercícios no Moodle!

Sugestão:

Crie um projeto chamado **Lista-01** no Eclipse.
Crie um pacote para cada exercício.

Por exemplo:

ex1

ex2

. . .