



# LPOO – Apresentação

---

Linguagem de Programação Orientada a Objetos

MSc. Olavo Ito

- Linguagem de Programação Orientada a Objetos
- Teoria 3ª feira quinzenal
- Laboratório: 5ª Feira

- Programação baseada no paradigma da orientação a objetos.
- **OBJETIVOS ESPECÍFICOS**
- Disseminar os princípios da Orientação a Objetos, aplicando-os através de da linguagem de programação Java. Ao término desta disciplina o aluno deverá demonstrar compreensão dos aspectos fundamentais do paradigma.
- **COMPETÊNCIAS**
- Compreender os conceitos fundamentais da programação orientada a objetos. Apropriar-se dos recursos básicos da linguagem Java para implementar produtos de software utilizando a orientação a objetos. Entender a transposição de um algoritmo para uma ferramenta de implementação.

- Introdução a orientação a objetos;
- Classe, Atributos e Métodos;
- Objetos;
- Encapsulamento;
- Herança;
- Método Construtor;
- Polimorfismo;
- Estruturas de Controle;
- Classe Abstrata;
- Interface;
- Exceções;
- Threads.

- **Básica**
- SANTOS, Rafael; Introdução à programação orientada a objetos usando Java / Campus; Rio de Janeiro, 2003.
- ALVES, William Pereira; Java 2 - Programação Multiplataforma – Ed. Érica, 2006.
- HORSTMANN, Cay; Conceitos de Computação com o Essencial de Java. Ed. Bookman – Porto Alegre, 2005.

- Java é uma linguagem de programação
- Java é um ambiente de desenvolvimento
- Java é uma completa plataforma de soluções para tecnologia

- ❑ **Editores e ambientes de produção gratuitos**
- ❑ **(NetBeans, Eclipse, Jcreator, etc)**
  
- ❑ **Servidores de Aplicação Gratuitos**
- ❑ **(TomCat, Jboss, Jetty, etc)**

## Arquitetura

- ❑ ***Orientado a Objetos (OO)***
  - ❑ A Linguagem segue o paradigma OO, o que a torna uma ferramenta extremamente poderosa.
- ❑ ***Multi-Threaded***
  - ❑ ***Suporta processamento paralelo múltiplo.***



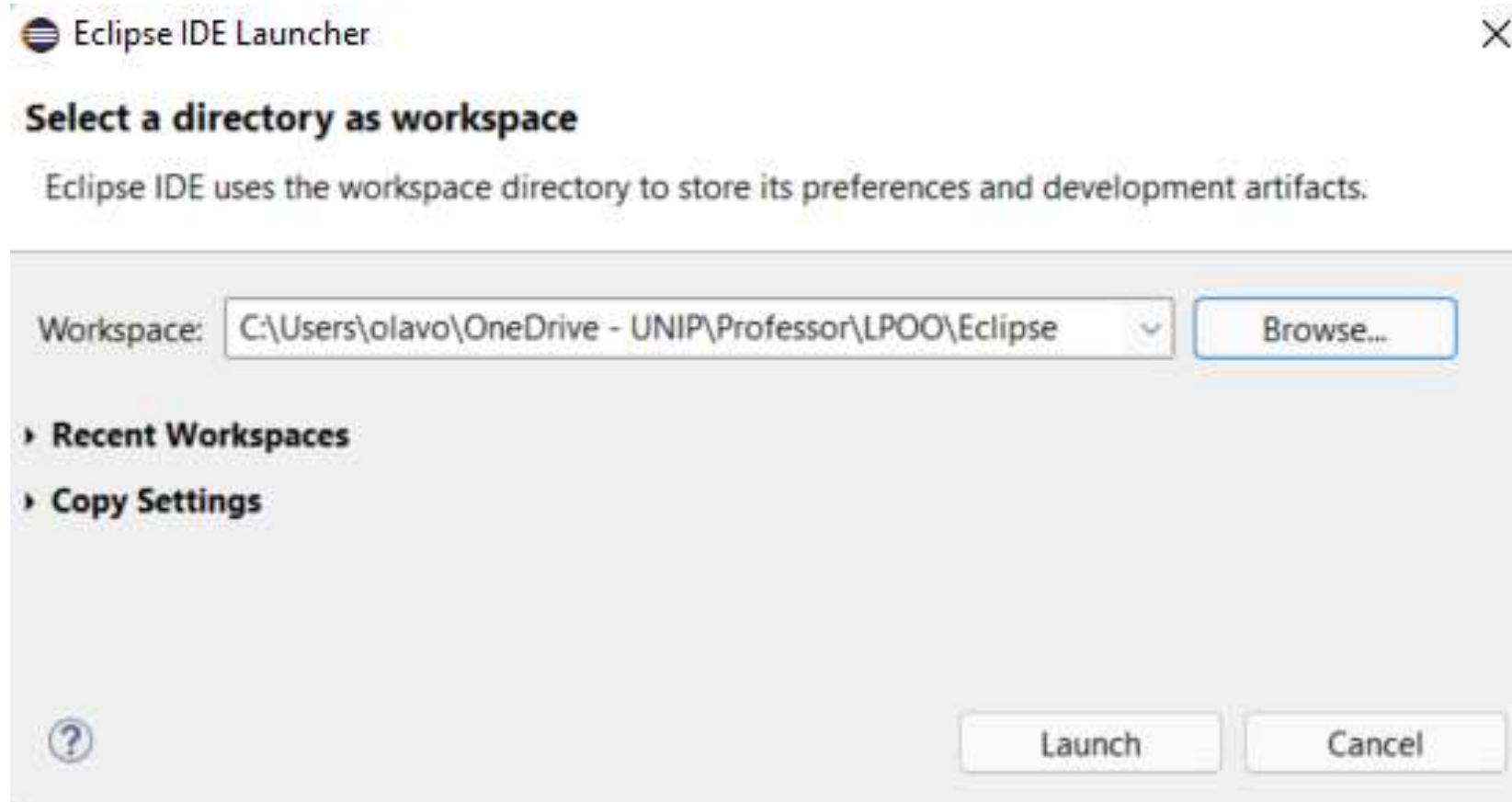
# Vantagens do JAVA

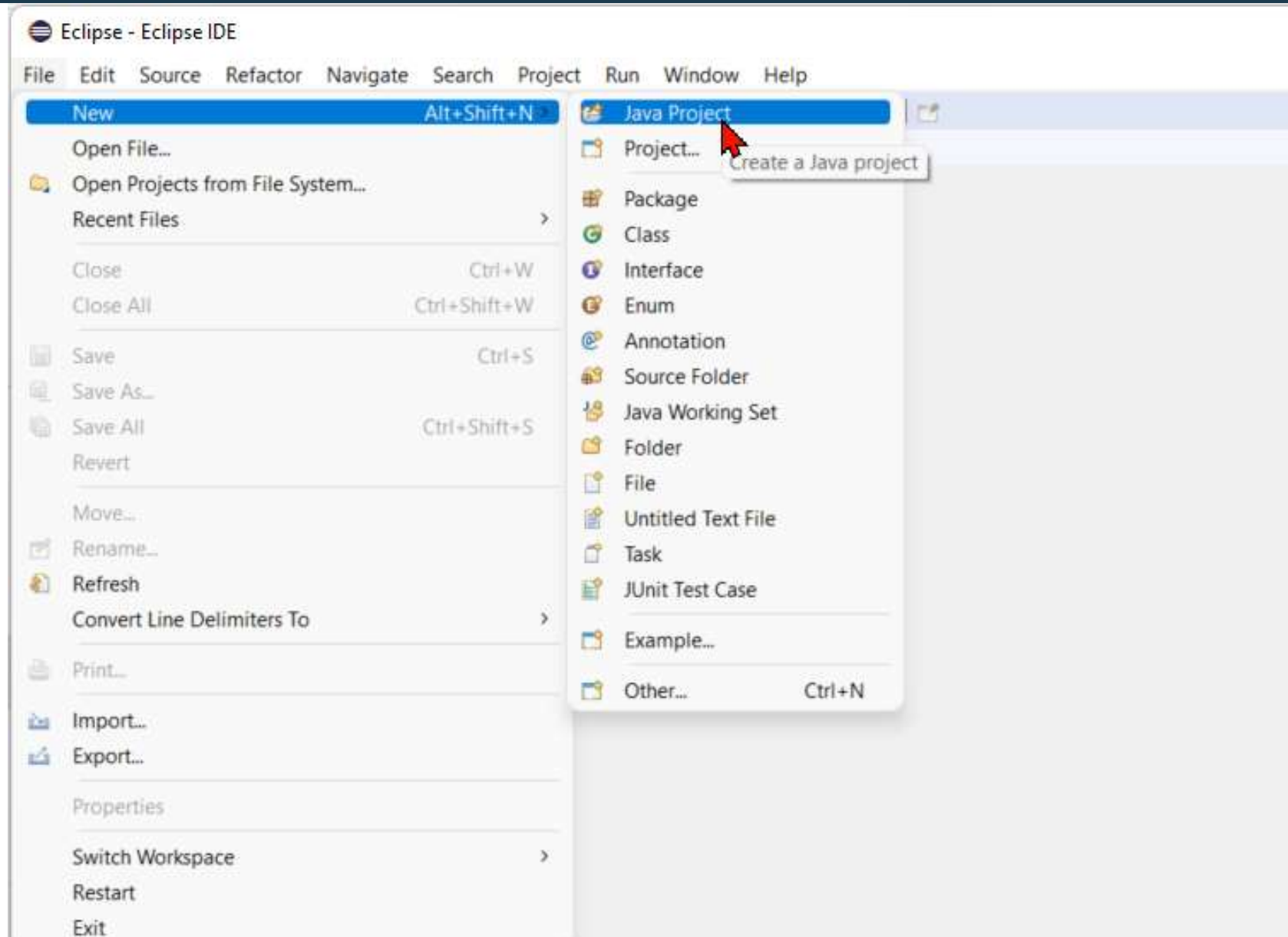
- ❑ O mesmo código Java roda em diversas plataformas sem a necessidade de alteração de código.
- ❑ As aplicações podem ser facilmente migradas entre servidores. Você não fica preso a somente um fabricante.
- ❑ Existem inúmeros JUGs (Java User Groups), que promovem eventos, disponibilizam material, artigos, revistas especializadas, fórum de discussões e etc.

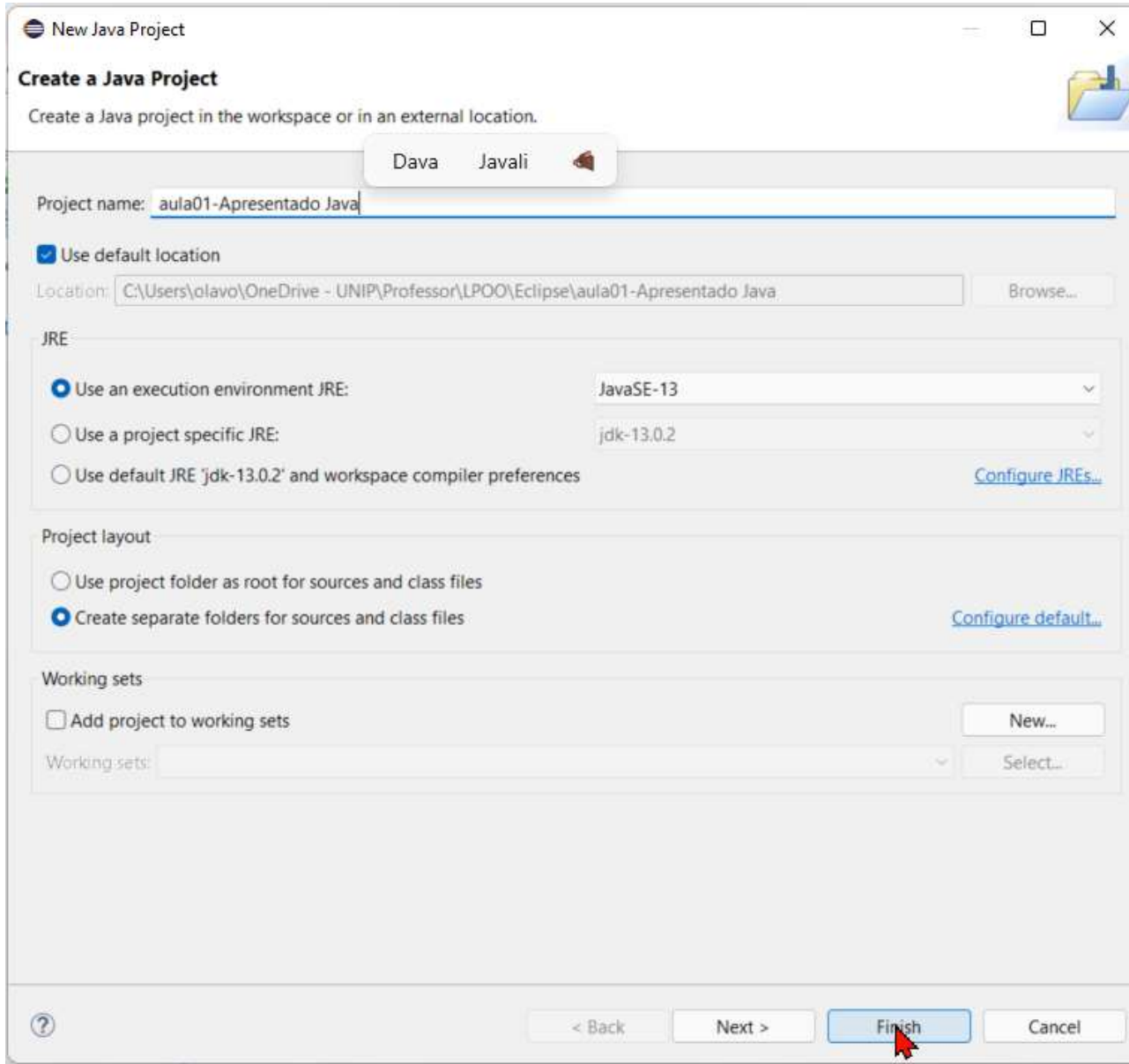
- Portabilidade
- Segurança
- Ex: quando os arquivos *bytecodes* vão ser executadas numa máquina, a JVM verifica se o programa obedece aos critérios e restrições de segurança da linguagem,
- como, por exemplo, não provocar o acesso direto à memória do computador em programas baixados pela rede.
- Com isso um programa em Java nunca pode conter um vírus que possa deteriorar a máquina do usuário.

- IDE
- <https://www.eclipse.org/downloads/packages/release/oxygen/3a/eclipse-ide-java-developers>
- <https://www.bluej.org/> Tem versão portátil
- <https://www.greenfoot.org/download>

# Linguagem







**New 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 'jdk-13.0.2' and workspace compiler preferences [Configure JREs...](#)

Project layout

☐ Use project folder as root for sources and class files

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

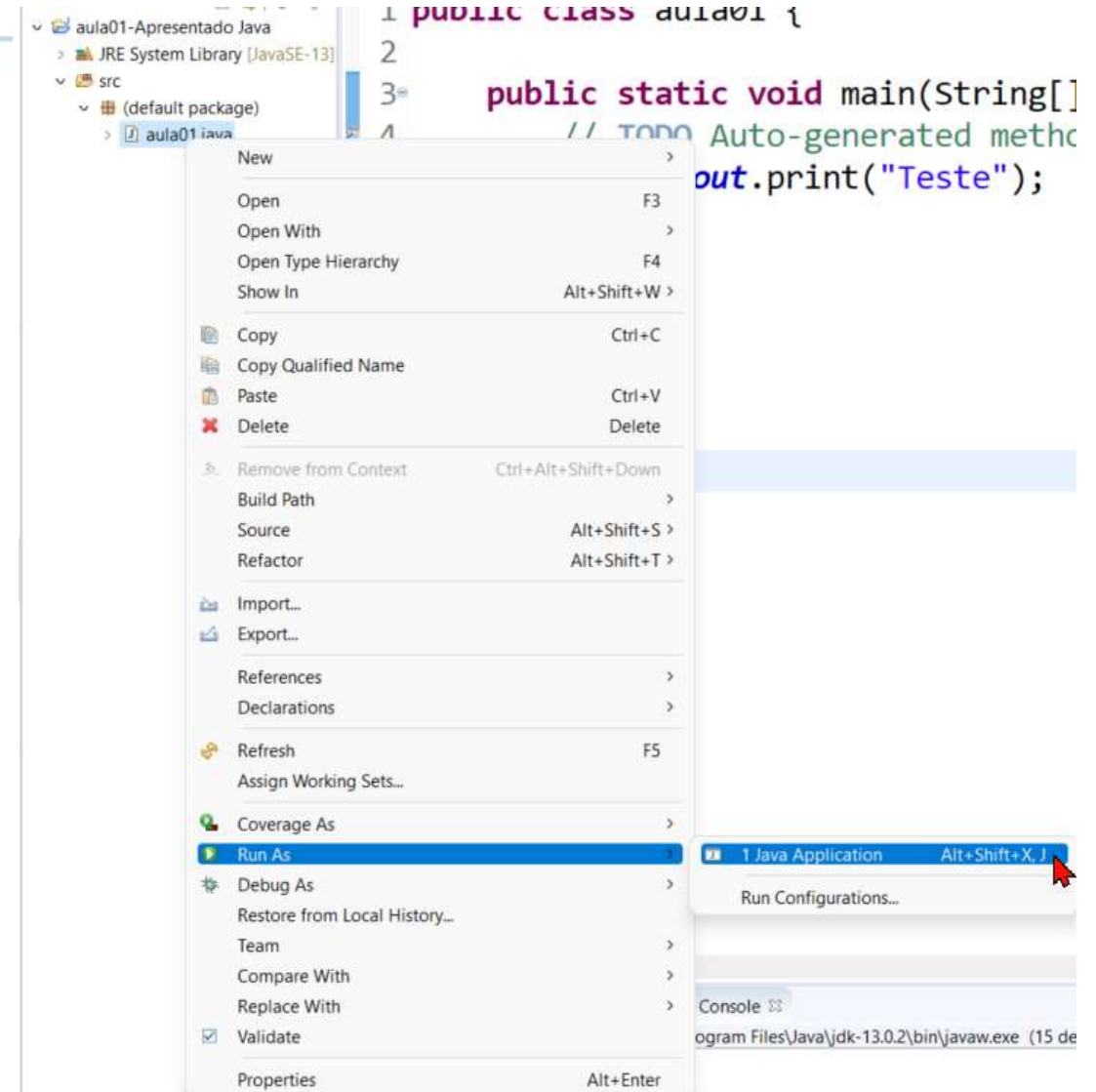
Working sets

☐ Add project to working sets [New...](#)

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

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

```
1 public class aula01 {  
2  
3     public static void main(String[] args) {  
4         // TODO Auto-generated method stub  
5         System.out.print("Teste");  
6  
7     }  
8 }
```





- A linguagem de programação, a linguagem JAVA é muito rígida na sua sintaxe.
  - Sintaxe são regras detalhadas para que um programa possa ser executado
- A sintaxe do JAVA pertence à família da linguagens C (C++, C# etc)
- A linguagem é *case sensitive*; isso quer dizer que as letras maiúsculas são diferentes das letras minúsculas, na identificação de comandos, variáveis e funções

Abacaxi ≠ abacaxi

- Os comandos são separados por ponto e vírgula (“;”), que deve ser usado com muito cuidado, principalmente, antes de blocos de comandos



# Iniciando

The screenshot illustrates the initial steps of creating a new Java class in the Eclipse IDE. On the left, the 'aula01-Apresentado Java' project is open, showing the 'src' folder and the 'module-info.java' file. The 'New' menu is open, and the 'Class' option is selected, with a tooltip that says 'Create a Java class'. The 'New Java Class' dialog is displayed on the right, showing the following configuration:

- Source folder:** aula01-Apresentado Java/src
- Package:** (default)
- Enclosing type:** (empty)
- Name:** aula01
- Modifiers:** ☒ public, ☐ package, ☐ private, ☐ protected, ☐ abstract, ☐ final, ☐ static
- Superclass:** java.lang.Object
- Interfaces:** (empty)
- 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

The 'Finish' button is highlighted at the bottom right of the dialog.

Tipo	Descrição	Tamanho (“peso”)
<b>byte</b>	Valor inteiro entre -128 e 127 (inclusivo)	1 byte
<b>short</b>	Valor inteiro entre -32.768 e 32.767 (inclusivo)	2 bytes
<b>int</b>	Valor inteiro entre -2.147.483.648 e 2.147.483.647 (inclusivo)	4 bytes
<b>long</b>	Valor inteiro entre -9.223.372.036.854.775.808 e 9.223.372.036.854.775.807 (inclusivo)	8 bytes
<b>float</b>	Valor com ponto flutuante entre $1,40129846432481707 \times 10^{-45}$ e $3,40282346638528860 \times 10^{38}$ (positivo ou negativo)	4 bytes
<b>double</b>	Valor com ponto flutuante entre $4,94065645841246544 \times 10^{-324}$ e $1,79769313486231570 \times 10^{308}$ (positivo ou negativo)	8 bytes
<b>boolean</b>	true ou false	1 bit
<b>char</b>	Um único caractere Unicode de 16 bits. Valor inteiro e positivo entre 0 (ou ‘\u0000’) e 65.535 (ou ‘\uffff’)	2 bytes

- Na linguagem de programação Java, as variáveis devem ser declaradas para que possam ser utilizadas.
- A declaração de uma variável envolve definir um nome único (identificador) dentro de um escopo e um tipo de valor.
- As variáveis são acessadas pelos nomes e armazenam valores compatíveis com o seu tipo
- Em geral, as linguagens de programação possuem **convenções** para definir os nomes das variáveis.
  - Essas convenções ajudam o desenvolvimento de um código mais legível.
  - Na convenção de nomes da linguagem Java, os nomes das variáveis devem seguir o padrão **camel case** com a primeira letra minúscula (**lower camel case**). Veja alguns exemplos:

nomeDoCliente  
numeroDeAprovados

```
1 public class aula01 {  
2  
3     public static void main(String[] args) {  
4         // TODO Auto-generated method stub  
5         int numero = 12;  
6         // Uso da variável  
7         System.out.println(numero);  
8         // Outra Declaração com Inicialização double preco = 137.6;  
9         double preco = 123.4;  
10        System.out.print(preco+" "+numero);  
11  
12    }  
13 }
```

- Aritmético (+, -, \*, /, %)
- Atribuição (=, +=, -=, \*=, /=, %=)
- Relacional (==, !=, <, <=, >, >=)
- Lógico (&&, ||, !) corresponde ao and, or e not do python

- Você pode converter os valores de um tipo para outro explicitamente usando o operador de conversão: (<tipo>) operação.

```
1 public class aula01 {  
2  
3     public static void main(String[] args) {  
4         int soma=17,qtd=5;  
5         double media;  
6         media=soma/qtd;  
7         System.out.println (media);  
8  
9     }  
10 }
```

3.0

```
1 public class aula01 {  
2  
3     public static void main(String[] args) {  
4         int soma=17,qtd=5;  
5         double media;  
6         media=(double)soma/qtd;  
7         System.out.println (media);  
8  
9     }  
10 }
```

3.4

- No Python é determinado pela indentação
- No Java por meio de { }

Python	JAVA
<pre>XXXXXXXXXXXXXXXXXXXXX     bbbbbbbbbbbbbbb     bbbbbbbbbbbbbbb     bbbbbbbbbbbbbbb         cccccccccc         cccccccccc</pre>	<pre>XXXXXXXXXXXXXXXXXXXXX{     bbbbbbbbbbbbbbb     bbbbbbbbbbbbbbb     bbbbbbbbbbbbbbb{         cccccccccc         cccccccccc     } }</pre>



# IF-ELSE

```
if (<condição>){  
    condição verdadeira;  
}  
else{  
    condição falsa;  
}
```

## Python

```
def main():  
    numero=10  
    if numero == 10:  
        print("\n-----\n")  
        teste=numero*10  
        print(teste)  
    else:  
        numero+=3  
        if numero > 14:  
            print("\n-----\n")  
            teste=numero*14  
            print(teste)  
        else:  
            print("\n-----\n")  
            teste=numero*5  
            print(teste)
```

## JAVA

```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
    int numero = 10;  
    int teste;  
    if (numero==10) {  
        System.out.print("\n-----\n");  
        teste=numero*10;  
        System.out.print(teste);  
    }  
    else {  
        numero+=3;  
        if (numero>14) {  
            System.out.print("\n-----\n");  
            teste=numero*14;  
            System.out.print(teste);  
        }  
        else {  
            System.out.print("\n-----\n");  
            teste=numero*5;  
            System.out.print(teste);  
        }  
    }  
}
```

# switch

Py  
thon

**JAVA**

```
switch ( expressao )
{
case opcao1:
... /* comandos executados se expr == opcao1 */
    break;
case opcao2:
... /* comandos executados se expr == opcao2 */
    break;
case opcao3:
... /* comandos executados se expr == opcao3 */
    break;
default:
... /* executados se expr for diferente de todos */
    break;
}
```

```
public static void main(String[] args) {
    char op = '-';
    int num1 = 3, num2 = 6;
    switch (op) {
        case '+':
            System.out.println( num1 + num2);
            break;
        case '-':
            System.out.println( num1 - num2);
            break;
        case '*':
            System.out.println( num1 * num2);
            break;
        case '/':
            if (num2 != 0)
                System.out.println( num1 / num2);
            else
                System.out.println("divisao por zero\n");
            break;
        default:
            System.out.println("Operador invalido!\n");
            break;
    }
}
```

# WHILE

```
while (<condição>)  
{
```

## Python

```
contador = 0  
while (contador < 5):  
    print("L P O O ", contador)  
    contador=contador+1
```

## JAVA

```
int contador = 0;  
while ( contador < 5)  
{  
    System.out.println (" L P O O "+contador);  
    contador ++;  
}
```

# DO WHILE

do{

} while (<condição>)

Python

JAVA

```
int contador = 0;
do {
    System.out.println (" L P O O "+contador);
    contador ++;
}while ( contador < 5);
```

```
for(varcontrole=<inicio>;<condição>;<incremento>){  
  
}
```

## Python

```
for varAux in range(5):  
    print("L P O O ",varAux)
```

## JAVA

```
int varAux;  
for(varAux=0;varAux<5;varAux++)  
    System.out.println (" L P O O "+varAux);
```

```
<tipo de retorno> nome( <tipo1> var1, <tipo2> var2){  
    Corpo da função;  
    return retorno;  
}
```

## Python

```
def areaTri(base, altura):  
    area=base*altura/2  
    return area  
  
print(areaTri(5,1))
```

## JAVA

```
public static float areaTri(int base,int altura) {  
    float area=(float)base*altura/2;  
    return area;  
}  
public static void main(String[] args) {  
    System.out.println(areaTri(5,1));  
}
```



# Leitura de dados (Classe Scanner)

```
import java.util.Scanner;  
Scanner scan = new Scanner(System.in);  
▪ scan será o nome da instância de leitura
```

## Python

```
nome=input("Informe o seu nome")  
  
idade=int(input("Informe a sua idade"))  
  
nota=float(input("Informe a sua nota"))  
  
print("O nome digitado é:", nome)  
print("A idade digitada é:", idade)  
print("A idade digitada é:", nota)
```

## JAVA

```
import java.util.Scanner;  
public class aula01 {  
    public static void main(String[] args) {  
        Scanner scan = new Scanner(System.in);  
        String nome;  
        int idade;  
        float nota;  
        System.out.println("Informe o seu nome");  
        nome = scan.next();  
        System.out.println("Informe a sua idade");  
        idade = scan.nextInt();  
        System.out.println("Informe a sua nota");  
        nota = scan.nextFloat();  
        System.out.println("O nome digitado é:"+nome);  
        System.out.println("A idade digitada é:"+idade);  
        System.out.println("A nota digitada é:"+nota);  
    }  
}
```

**ATÉ A PRÓXIMA!**