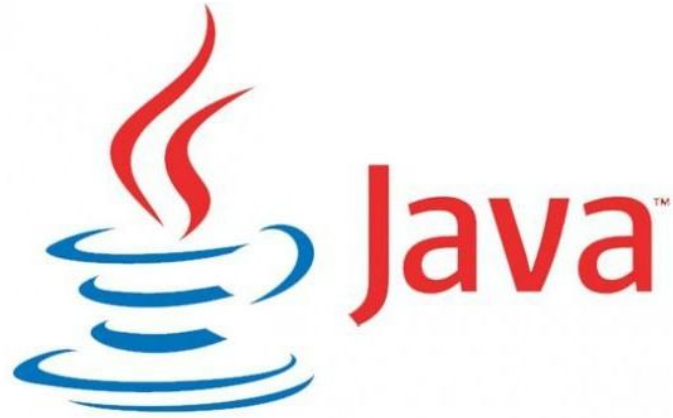
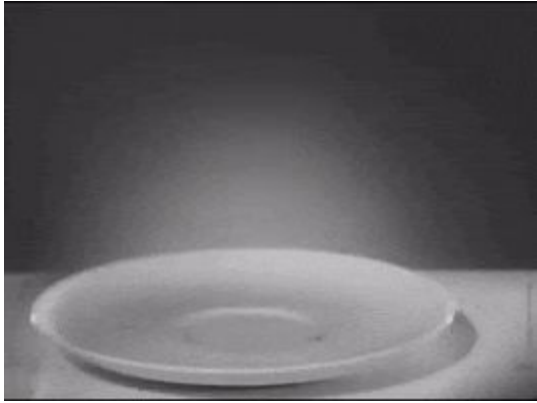


JAVA



JAVA COFFEE



JAVA RING...





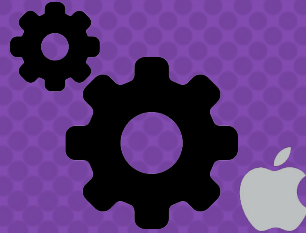
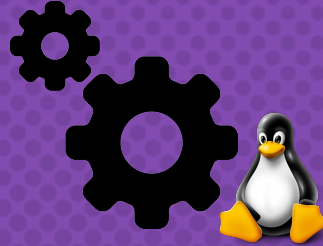
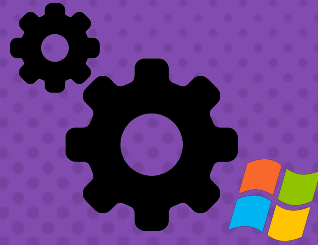
VS



ENTENDENDO

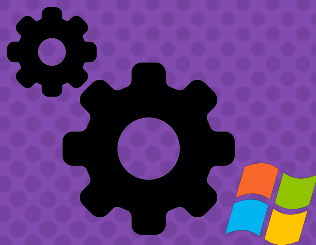
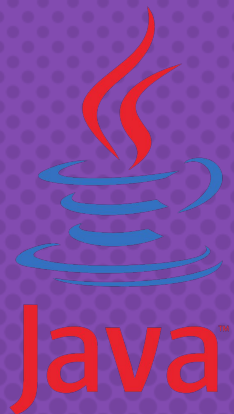
Como funciona o Java





Compilador

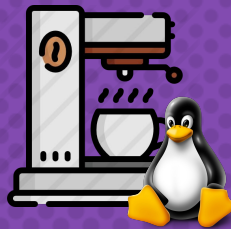
Código
executável



JavaC



Bytecode



JVM



JDK



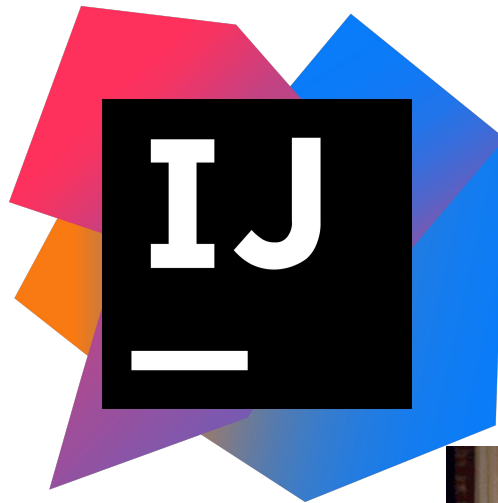
JRE



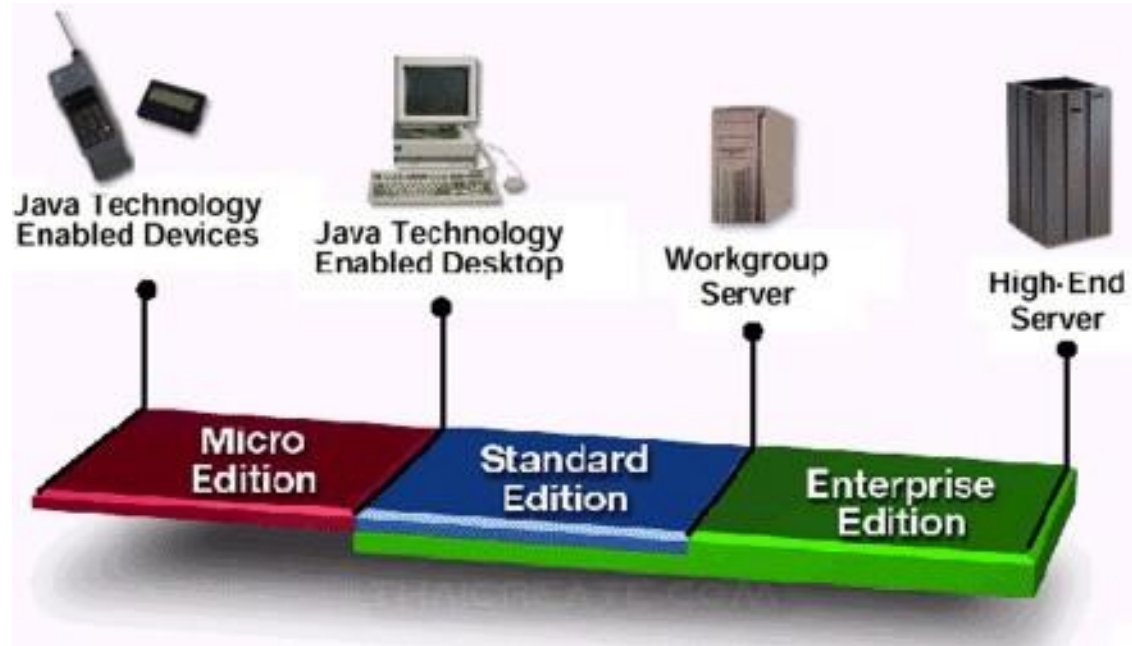
IDE



NetBeans

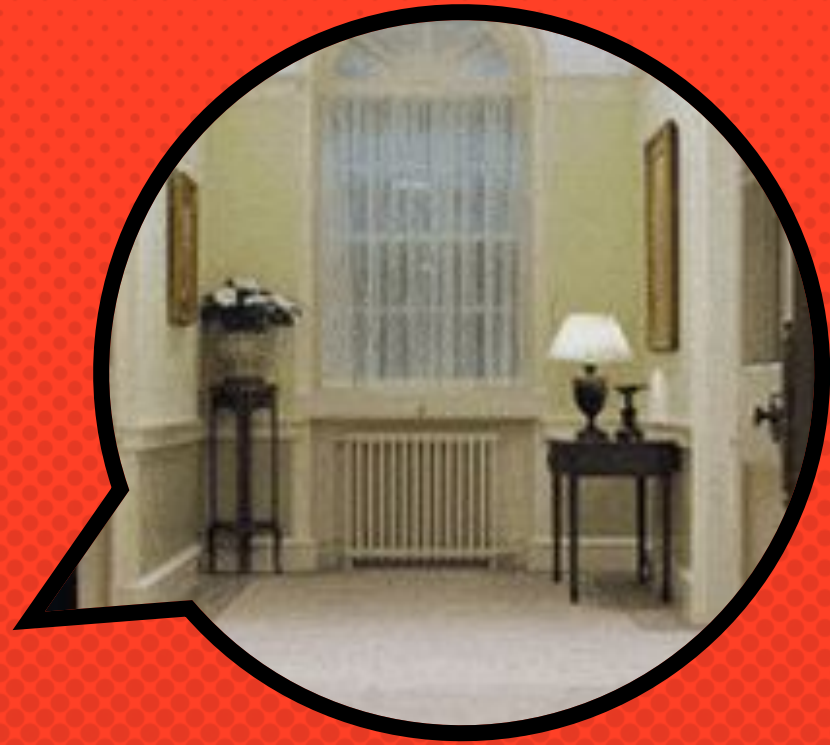


JDK



× Fazer download do Java SE

Entendendo a estrutura da Linguagem Java



```
1  // Comando opcional
2  package firstprogram;
3
4  //Cabeçalho da class
5  public class FirstProgram {
6      //Método principal
7      public static void main(String[] args) {
8
9      }
10 }
```

```
15 // Comando opcional
16 package firstprogram;
17
18 //Cabeçalho da class
19 public class FirstProgram {
20     //Método principal
21     public static void main(String[] args) {
22         System.out.print("Eu sou maravilhosa!!");
23     }
24 }
```


ALGUMAS CONVENÇÕES...

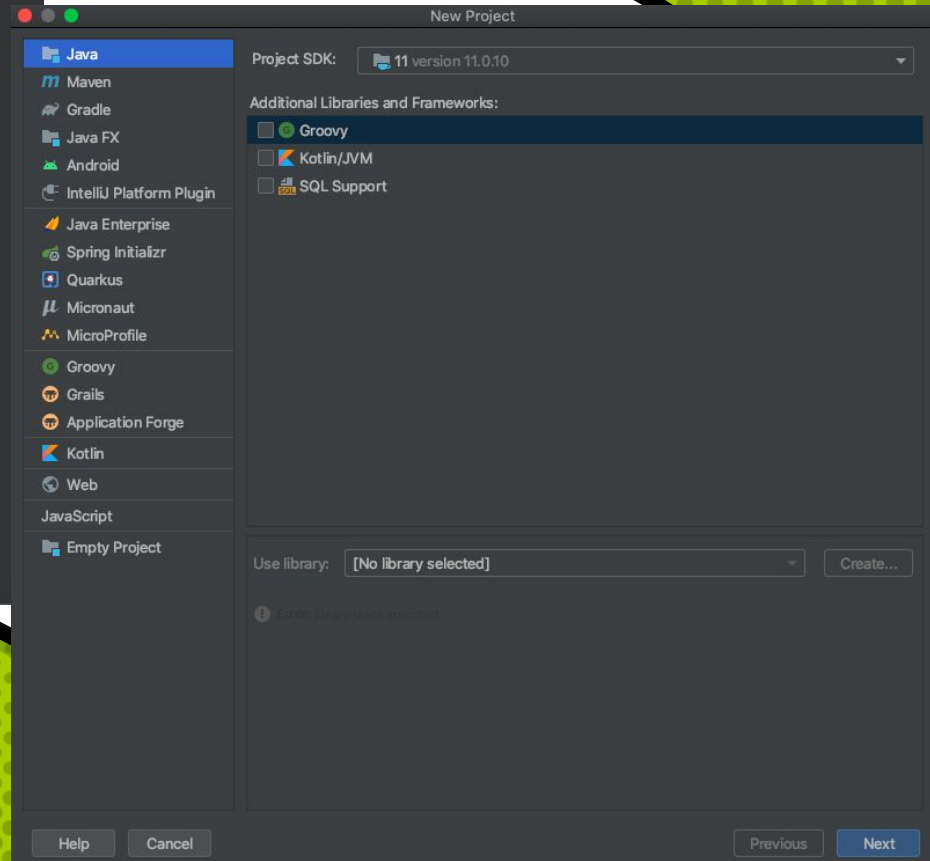
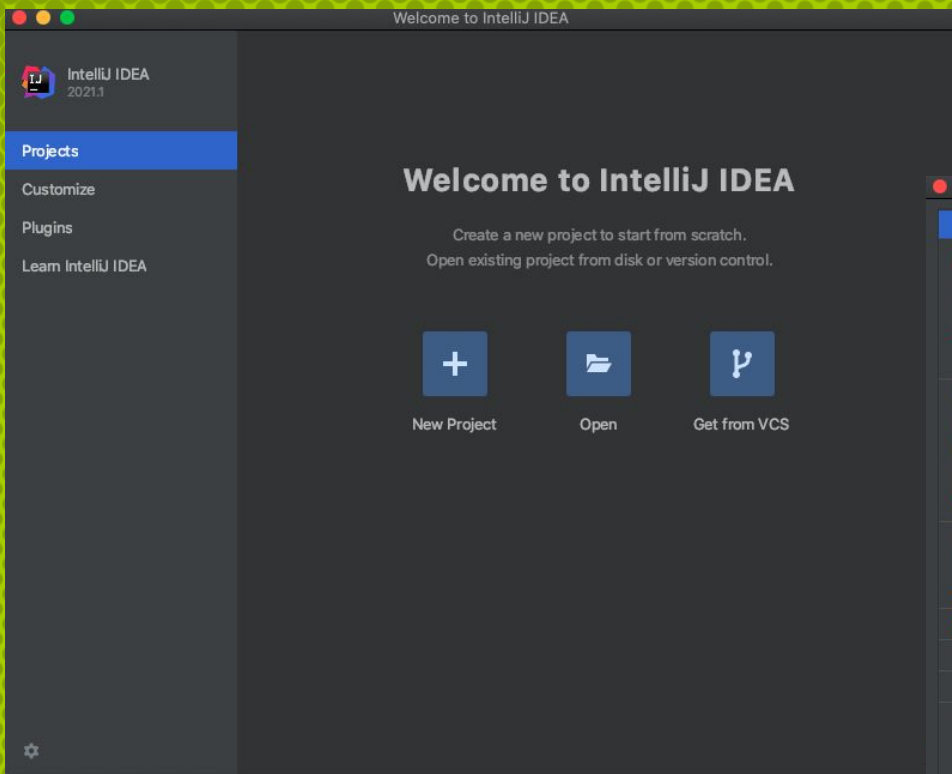
- × **M**inha**C**lasse
- × **M**inha**I**nterface
- × meu**A**tributo
- × minha**V**ariável
- × meu**M**étodo
- × meupacote
- × **MINHA_CONSTANTE**

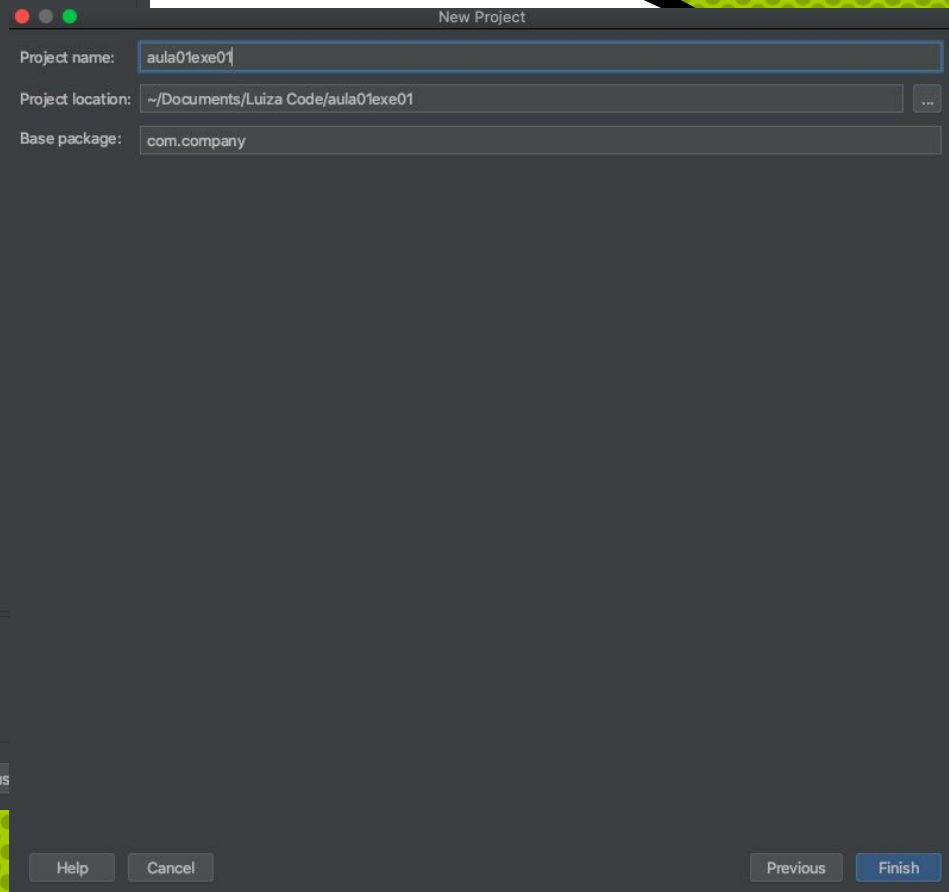
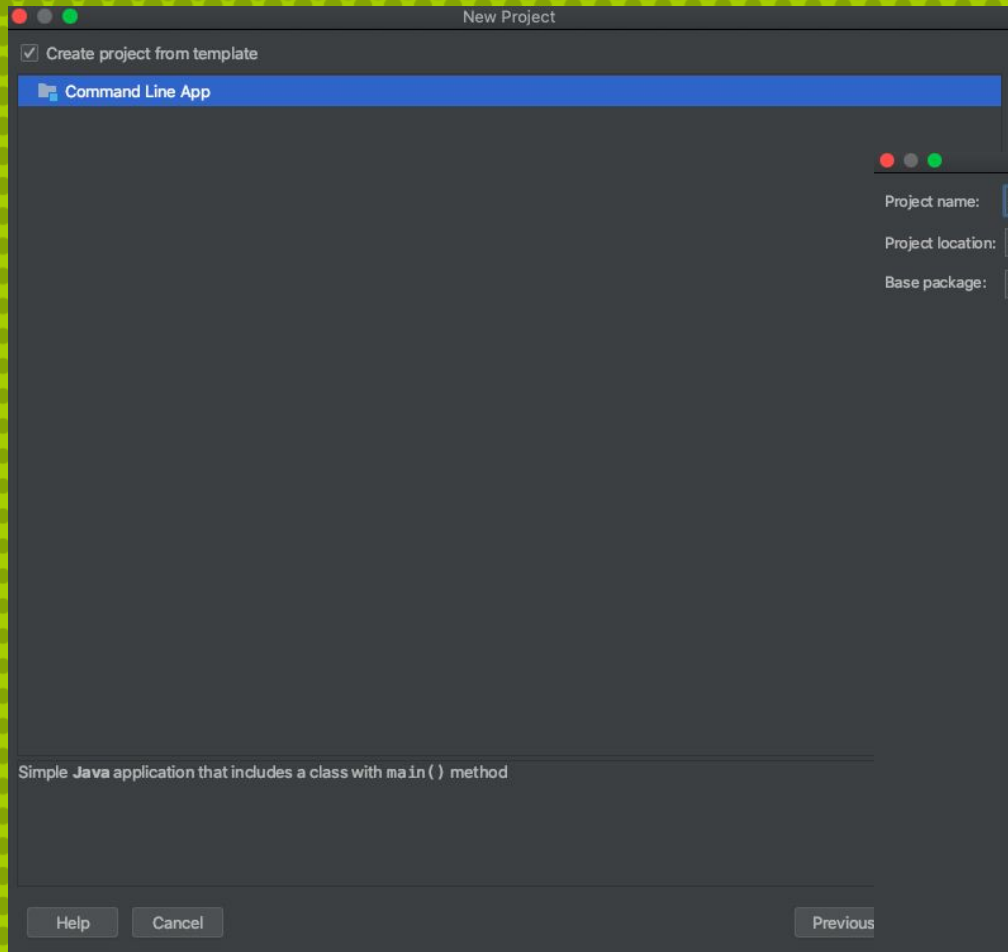
A linguagem Java
possui sintaxe case
sensitive





MÃO NA MASSA





aula01exe01 > src > com > company > Main



Project

Project ▾

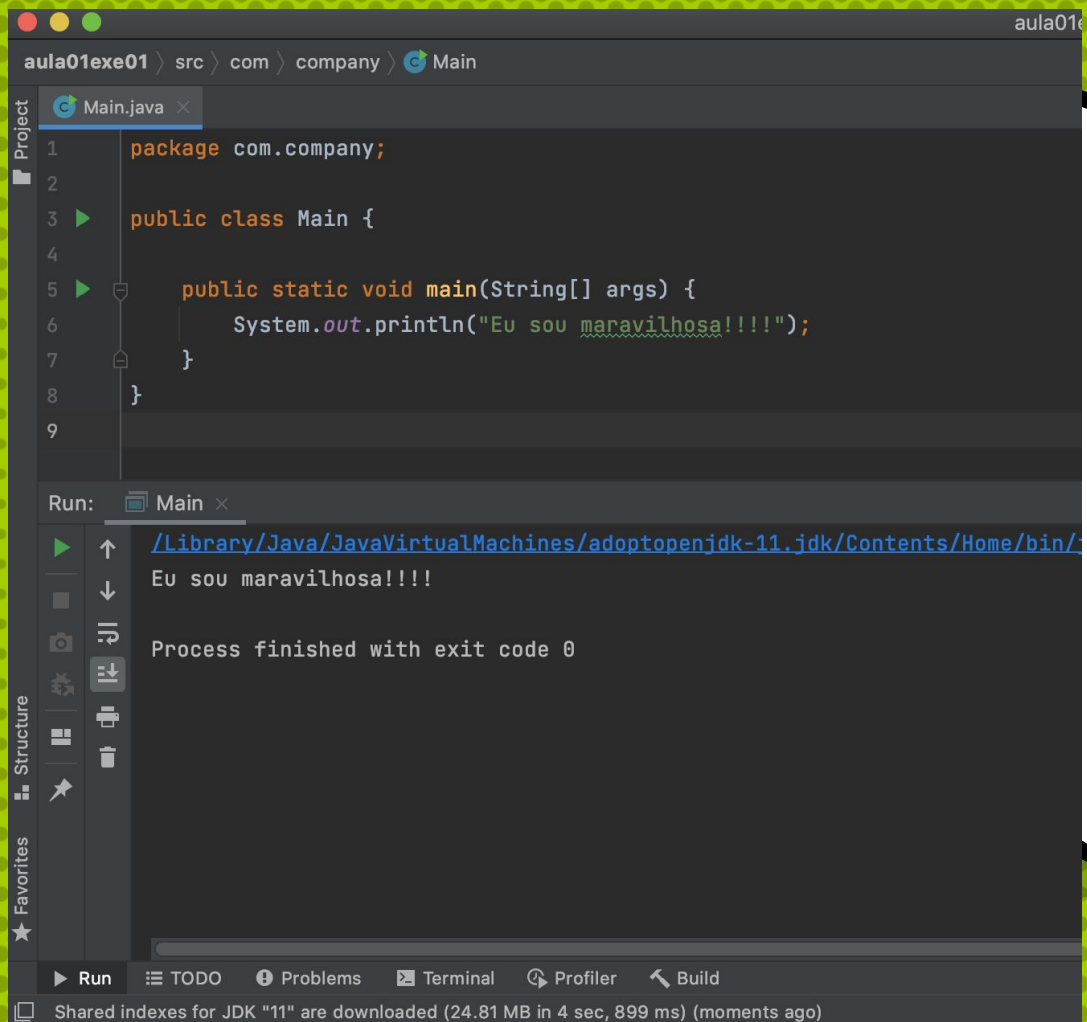
> aula01exe01 ~/Documents/Luiza Code/aula01e

> External Libraries

Scratches and Consoles

Main.java x

```
1 package com.company;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         // write your code here
7     }
8 }
9
```



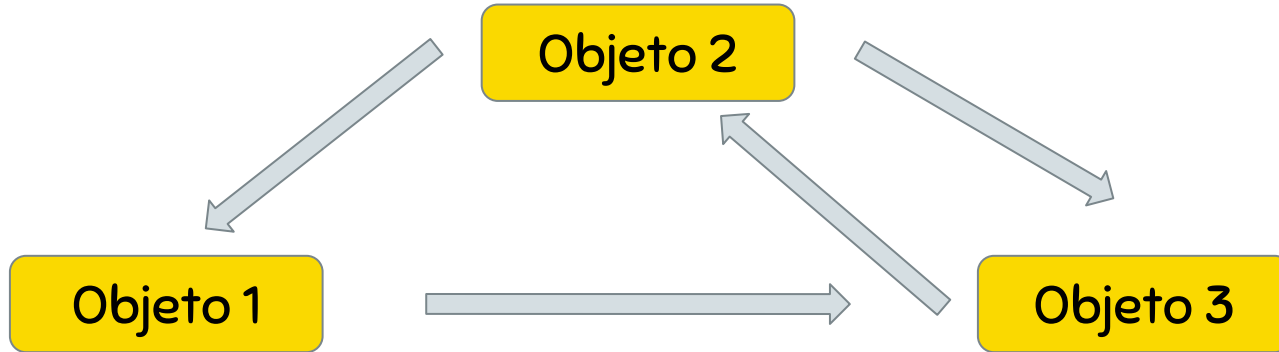
Java é uma linguagem POO



PROGRAMAÇÃO ESTRUTURADA VS PROGRAMAÇÃO ORIENTADA A OBJETOS

- × Pensamento de máquina
 - × Problemas mais simples e diretos
 - × Sequência lógica
- × Pensamento humano
 - × Sequência lógica mas pensando em modelar problemas de uma forma mais natural e parecido com o mundo real

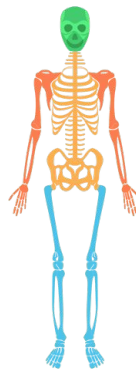
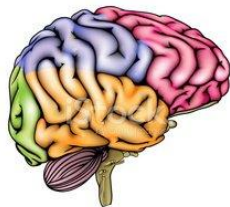
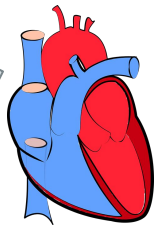
PROGRAMAÇÃO ORIENTADA A OBJETOS



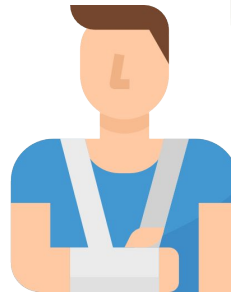
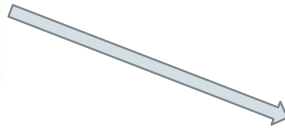
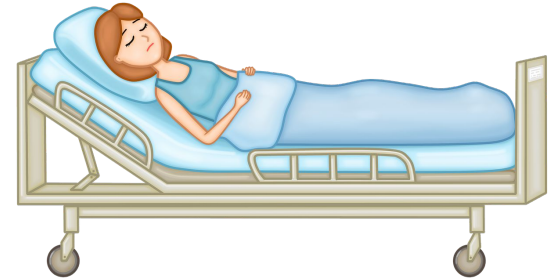
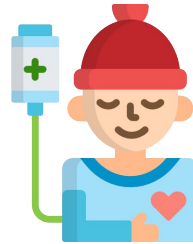
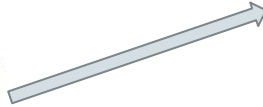
PILARES DA ORIENTAÇÃO A OBJETOS



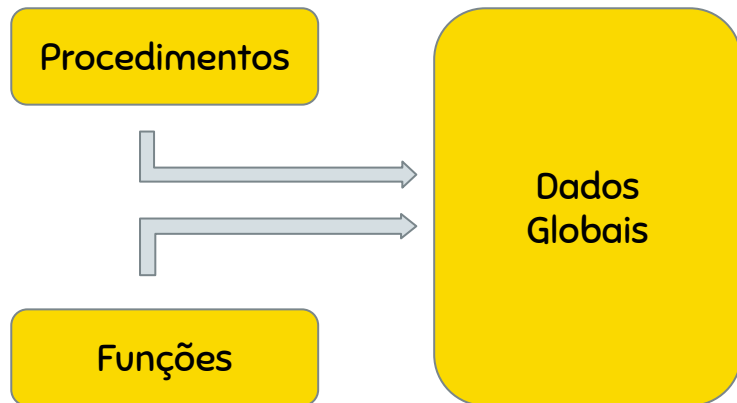
PILARES DA ORIENTAÇÃO A OBJETOS



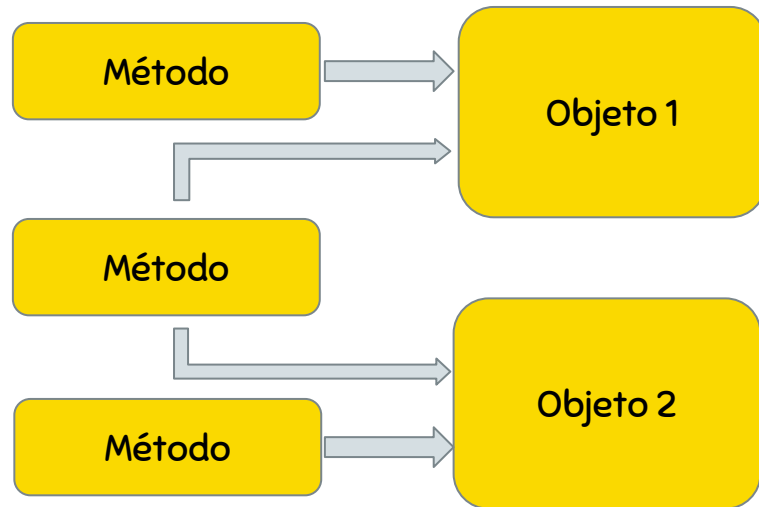
PILARES DA ORIENTAÇÃO A OBJETOS



PROGRAMAÇÃO ESTRUTURADA



PROGRAMAÇÃO ORIENTADA A OBJETO



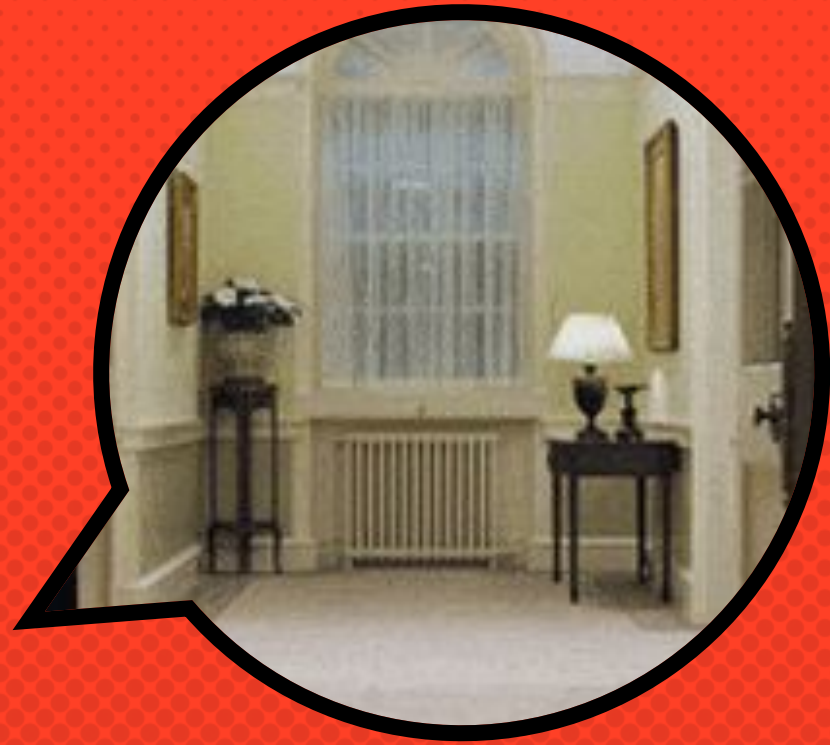
PROGRAMAÇÃO ESTRUTURADA

1	Desligar o interruptor
2	Pegar uma escada
3	Montar a escada
4	Subir na escada
5	Desenroscar a lâmpada queimada
6	Descer da escada
7	Jogar a lâmpada queimada no lixo
8	Pegar uma lâmpada nova
9	Subir na escada
10	Rosquear a nova lâmpada
11	Descer da escada
12	Ligar o interruptor para verificar se a nova lâmpada acende

PROGRAMAÇÃO ORIENTADA A OBJETO

Objetos	Características
Pessoa	Pegar, descer, subir, jogar
Lâmpada	Inteira, quebrada, ligada, desligada

Primeiros passos para trabalhar
com POO



DECLARANDO AS VARIÁVEIS

```
tipoDaVariavel nomeDaVariavel;
```

Exemplos:

```
Int idade;
```

```
String nome;
```


DECLARANDO AS VARIÁVEIS

int idade;



String nome123;



String nome;



String nome_123;



int 3data;



int Idade;



DECLARANDO AS VARIÁVEIS

```
int idade;  
Idade = 10;
```

```
int idadeAtual = 10;  
int idadeFutura = 10 + 5;  
int dobroDaldade = 10 * 2;
```

TIPOS

var

```
idade : Inteiro  
sal : Real  
letra : Caractere  
casado : Logico
```

inicio

```
idade <- 3  
sal <- 1825.54  
letra <- "G"  
casado <- falso
```

```
int idade = 3;  
float sal = 1825.54f;  
char letra = 'G';  
boolean casado = false;
```

typecast

```
int idade = (int) 3;  
float sal = (float) 1825.54;  
char letra = (char) 'G';  
boolean casado = (boolean) false;
```

Wrapper Class

```
Integer idade = new Integer(3);  
Float sal = new Float(1825.54);  
Character letra = new Character('G');  
Boolean casado = new Boolean(false);
```

Família	Tipo Primitivo	Classe Invólucro	Tamanho	Exemplo
Lógico	boolean	Boolean	1 bit	true
Literais	char	Character	1 byte	'A'
	-	String	1 byte/cada	"JAVA"
Inteiros	byte	Byte	1 byte	127
	short	Short	2 bytes	32 767
	int	Integer	4 bytes	2 147 483
	long	Long	8 bytes	2^{63}
Reais	float	Float	4 bytes	$3.4e^{+38}$
	double	Double	8 bytes	$1.8e^{+308}$

DECLARANDO AS VARIÁVEIS

```
private String nome;
```

```
public int idade;
```

```
protect String cpf;
```


CASTING

Atribuição direta:

```
int saldo = 10;
```

//saldo recebe 10

```
int total = saldo;
```

//total recebe uma cópia do
valor de saldo, no caso 10

```
saldo = saldo - 4;
```

// saldo recebe o resultado do
que já tinha em saldo - 4,
no caso 6

CASTING

double pi = 3.14; //pi recebe 3.14

int numeroPi = pi; //erro, não compila

CASTING

```
double num = 3;           //num recebe 3.14  
int num2 = num;           //erro, não compila
```

```
int num = 3;               //num recebe 3.14  
double num2 = num;         //num2 recebe o valor  
                             armazenado em num,  
                             no caso 3
```

CASTING

`double num = 3.14;`

`//num recebe 3.14`

`int num2 = (int)num;`

`//num2 recebe o valor
arredondado de num,
no caso 3`

CASTING

Para / De:	Byte	Short	Char	Int	Long	Float	Double
Byte	-----	automático	(char)	automático	automático	automático	automático
Short	(byte)	-----	(char)	automático	automático	automático	automático
Char	(byte)	(short)	-----	automático	automático	automático	automático
Int	(byte)	(short)	(char)	-----	automático	automático	automático
Long	(byte)	(short)	(char)	(int)	-----	automático	automático
Float	(byte)	(short)	(char)	(int)	(long)	-----	automático
Double	(byte)	(short)	(char)	(int)	(long)	(float)	-----

DECLARANDO CONSTANTES

```
final double NUM = 3.14;
```

```
final int VALUE = 6;
```

```
public static final int GLOBAL = 5;
```

```
private static final int LOCAL = 8;
```



"DICAS"

COMENTANDO SEU CÓDIGO

// Este é um comentário de um linha

/*

Este é um comentário de muitas linhas,
podemos escrever o sem limites

*/

/**

Este é um comentário para documentação

@author Tainá Medeiros

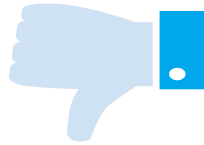
@version 1.0

@since 2021-19-01

**/

ALGORITMO

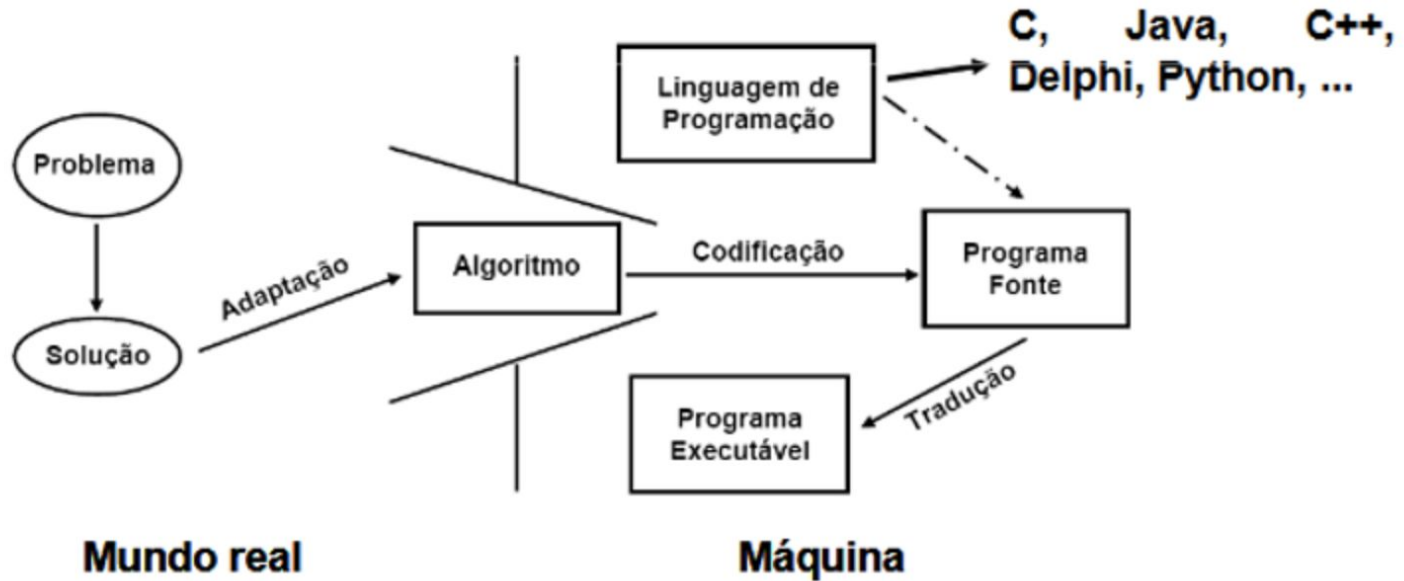
Algoritmo não é a solução do problema, pois, se assim fosse, cada problema teria um único algoritmo.



Algoritmo é o caminho para a solução de um problema, e em geral, os caminhos que levam a uma solução são muitos.



ALGORITMO



REPRESENTAÇÃO DE ALGORITMOS

Descrição Narrativa;

Fluxograma;








Linguagem algorítma, Pseudocódigo;

Descrição Narrativa

Cálculo da média de um aluno

- 1 – Obter as suas 2 notas de provas
- 2 – Calcular a média aritmética
- 3 – Se a média for maior que 7,
o aluno foi aprovado,
Senão
ele foi reprovado

Fluxograma

	Início ou fim do algoritmo
	Indica o sentido do fluxo de execução do algoritmo. Conecta os objetos gráficos
	Representa a entrada de dados
	Indica cálculos e atribuições de valores (processamento)
	Indica desvios ou tomadas de decisões (Por exemplo: SE isso, ENTÃO aquilo)
	Representa a saída de dados, no Portugol IDE
	Também representa a saída de dados

Fluxograma

Cálculo da média de um aluno

Passo 1: informe a primeira nota;

Passo 2: informe a segunda nota;

Passo 3: some a primeira nota com a segunda nota;

Passo 4: divida o resultado da soma por dois;

Passo 5: mostre o resultado.

Fluxograma

Cálculo da média de um aluno

Passo 1: informe a primeira nota;

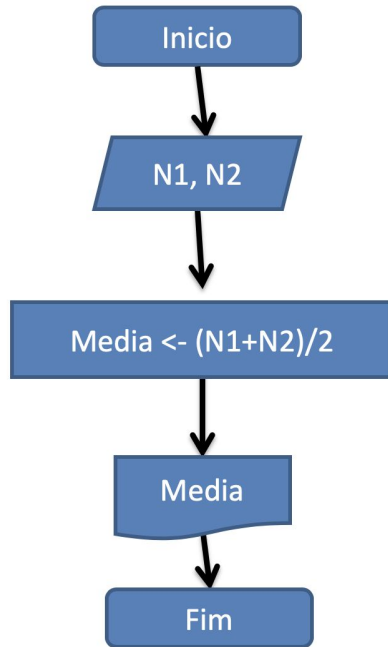
Passo 2: informe a segunda nota;

Passo 3: some a primeira nota com a segunda nota;

Passo 4: divida o resultado da soma por dois;

Passo 5: mostre o resultado.

Fluxograma



Começamos o fluxograma usando o símbolo de início.

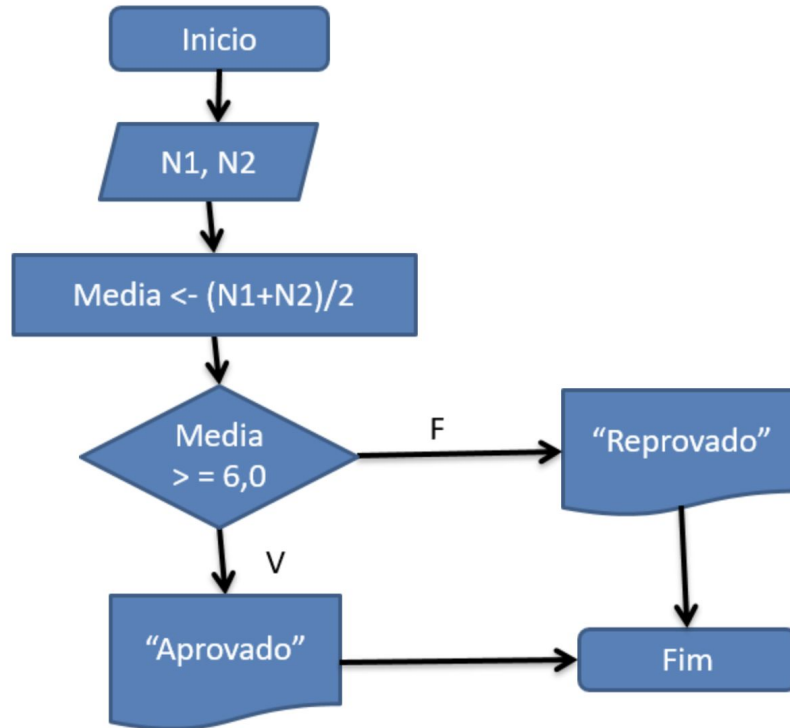
Depois, dentro do símbolo para entrada de dados, colocamos as duas notas do aluno que foram chamadas de N1 e N2

No símbolo de processamento ficou o cálculo da média do aluno. Dessa forma Média recebe (uso do símbolo \leftarrow para indicar a atribuição de um valor) o valor de N1 somado com N2 e dividido por 2.

Depois que a média é calculada, usamos o símbolo de saída de dados para dizer o que queremos que seja apresentado ao usuário

Finalmente, o símbolo de Fim é usado para indicar que o fluxograma acabou.

Fluxograma



Pseudocódigo

```
Algoritmo <nome_do_algoritmo>;  
<declaração_de_variáveis>;  
<subalgoritmos>  
Início  
<corpo do algoritmo>  
Fim
```

Pseudocódigo

algoritmo "Média" *//aqui, definimos o nome do algoritmo*

var *//início das declarações das variáveis*

nota1 : real *//aqui, é definida a entrada da primeira nota do aluno*

nota2 : real *// aqui, é definida a entrada da segunda nota do aluno*

media : real *//declaração da variável em que será armazenado o resultado do cálculo da média*

Início *//início do bloco de execução (local onde instruímos o computador a executar os comandos)*

escreva ("Digite o valor da primeira nota: ") *//o programa exibe na tela o texto entre aspas*

leia (nota1) *// o programa lê o que o usuário escreveu: a primeira nota*

escreva ("Digite o valor da segunda nota: ") *//o programa exibe na tela o texto entre aspas*

leia (nota2) *// o programa lê o que o usuário escreveu: a segunda nota*

$media \leftarrow (nota1 + nota2)/2$ *//é realizado o cálculo da média*

escreva ("A média do aluno é: ", media) *//o valor encontrado é exibido na tela*

fimalgoritmo *//fim do algoritmo*

Entrada, Saída e
Manipulação de dados



SAIDA DE DADOS

```
31  float nota = 8.5f;
32
33  System.out.print("Sua nota é " + nota);
34  System.out.println("Sua nota é " + nota);
35  System.out.printf("Sua nota é %f", nota);
36  System.out.printf("Sua nota é %.2f", nota);
37  System.out.printf("Sua nota é %.2f \n", nota);
38  System.out.format("Sua nota é %.2f \n", nota);
```

ENTRADA DE DADOS

```
40  import java.util.Scanner;
41
42  Scanner teclado = new Scanner(System.in);
43  String nome = teclado.nextLine();
44  float nota = teclado.nextFloat();
45  System.out.println("O aluno %s tem nota %.2f \n", nome, nota);
```

ENTRADA DE DADOS

`nextLine()`

Lê um texto digitado pelo usuário

`nextInt()`

Lê um número inteiro

`nextLong()`

Lê um número inteiro longo

`nextFloat()`

Lê um número float

`nextDouble()`

Lê um número double

STRING ↔ INT

```
int num = 3;
```

```
String value = num;           //erro, não compila
```

```
String value = (String) num;  //erro, não compila
```

```
int num = 3;
```

```
String value = Integer.toString(num);
```

INT ↔ STRING

```
String num = "30";
```

```
int value = num;           //erro, não compila
```

```
int value = (int) num;     //erro, não compila
```

```
String num = "30";
```

```
int value = Integer.parseInt(num);
```



MÃO NA MASSA

PRÁTICA 01

Escreva um algoritmo que leia o peso e a altura de um usuário e informe o seu Índice de Massa Corpórea (IMC).

O cálculo é obtido pela divisão entre o valor do seu peso e o quadrado do valor da sua altura.

```
1 package com.company;
2
3 import java.util.Scanner;
4
5 public class Main {
6
7     public static void main(String[] args) {
8         Scanner teclado = new Scanner(System.in);
9
10        System.out.println("Informe seu peso: ");
11        float peso = teclado.nextFloat();
12        System.out.println("Informe sua altura: ");
13        float altura = teclado.nextFloat();
14
15        float imc = peso / (altura * altura);
16
17        System.out.format("Seu IMC é %f", imc);
18    }
19 }
```

```
↑ /Library/Java/JavaVirtualMachines/adoptop
↓
Informe seu peso:
50
Informe sua altura:
1.50
Seu IMC é 22.222221
Process finished with exit code 0
```



THE END