

Algoritmos - Aula 0

Fernando Raposo

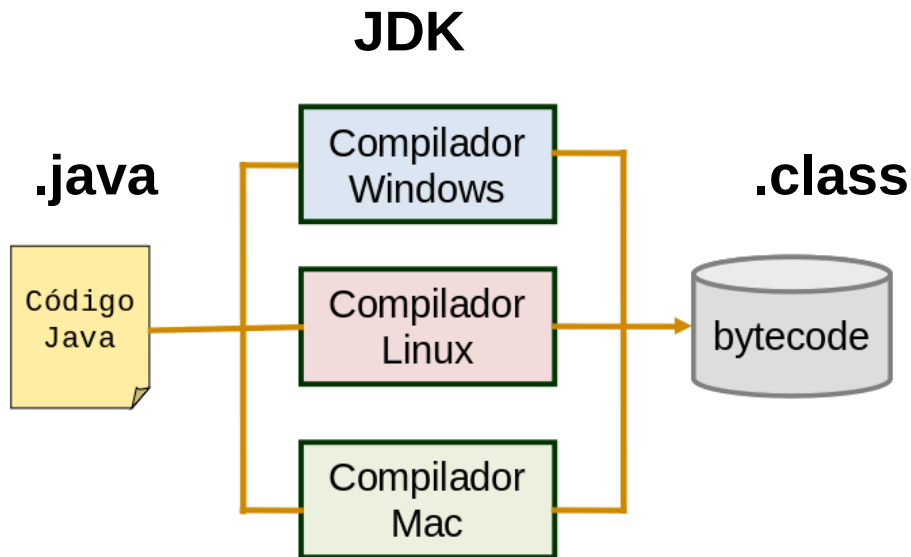
Apresentação

- Por que Java?
- Uma das mais populares linguagens de programação em [2020](#);

Aug 2020	Aug 2019	Change	Programming Language	Ratings	Change
1	2	▲	C	16.98%	+1.83%
2	1	▼	Java	14.43%	-1.60%
3	3		Python	9.69%	-0.33%
4	4		C++	6.84%	+0.78%
5	5		C#	4.68%	+0.83%
6	6		Visual Basic	4.66%	+0.97%
7	7		JavaScript	2.87%	+0.62%
8	20	▲	R	2.79%	+1.97%

O que eu preciso para começar?

- NADA podemos programar “on-line” pelo repl.it
- Mas queremos algo mais rápido... Então, dentro da sua máquina no mínimo precisamos instalar o JDK (Java Development Kit).



Instalando JDK

- [Como instalar JDK no Windows \(vídeo\)](#)
- O JDK seria suficiente para transformarmos os .java em .class e rodar...
 - Ou seja converter o código-fonte (em texto) em código objeto (em *bytecodes*)

.java  .class (Compilação)

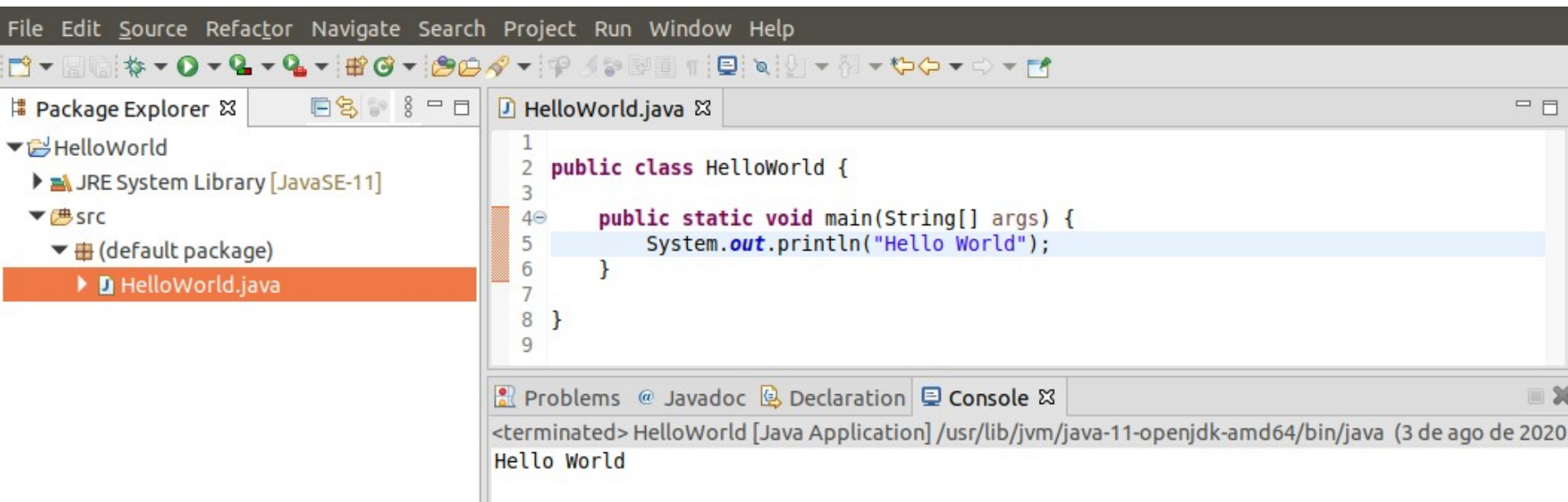
- O JDK também vem com o JRE (Java Runtime - JVM)

.class  Hello World! (Interpretação)

- Mas sempre podemos usar uma IDE para tal...

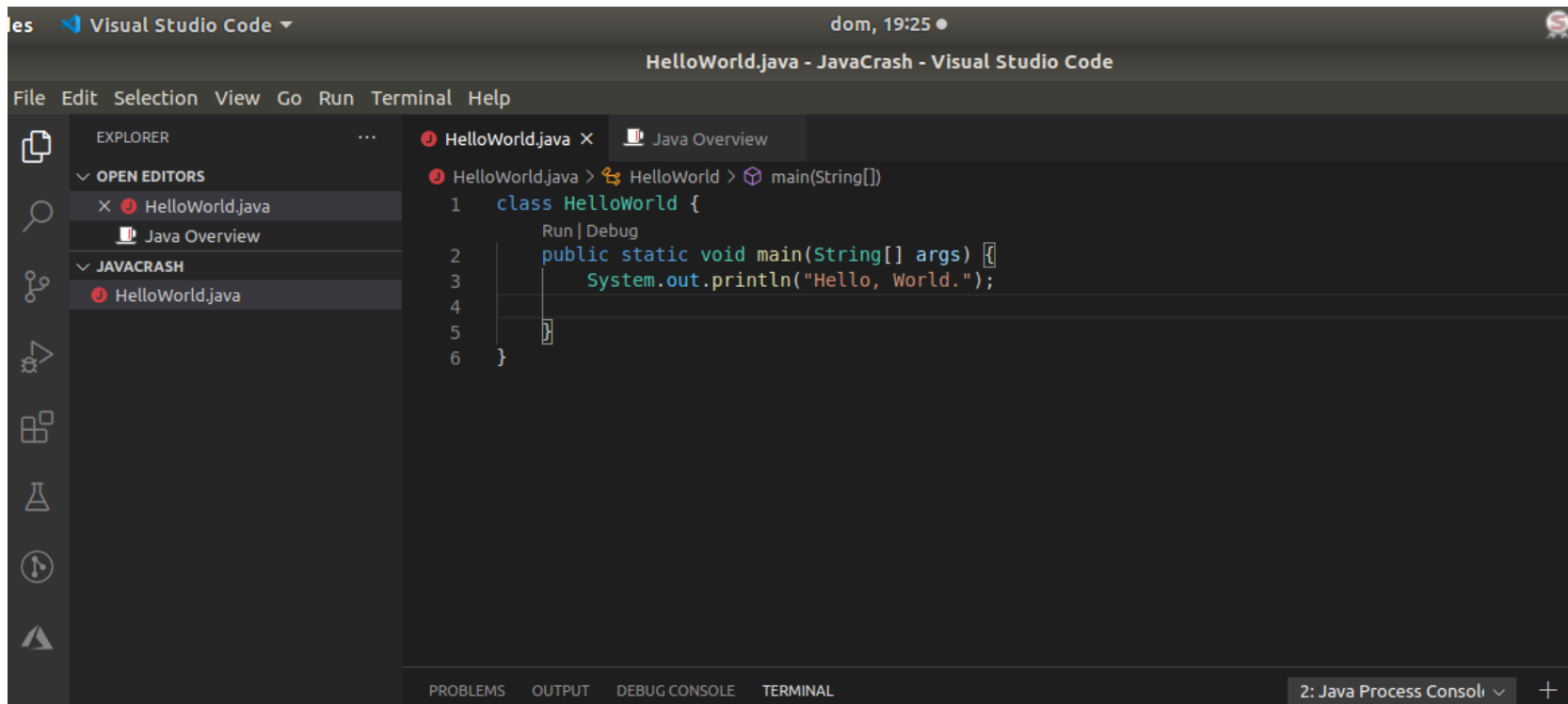
IDE

- [Eclipse](#)
- [Como instalar Eclipse numa máquina Windows \(vídeo\)](#)



IDE (Windows pode ser usado o WSL)

- VSCode



Qual IDE usar?

- Você decide, de acordo com seu gosto, com as condições do seu computador, etc...
- O Eclipse tem uma longa história, mas tem perdido um pouco de espaço para o VSCode e IntelliJ IDEA.
- Na verdade o ideal é sempre usar **Software Livre** [independente da linguagem que se usa :)]

Vantagens de JAVA

- Write **once**, run **anywhere**;
- Comunidades enormes, extensa bibliotecas sobre tudo que se pode imaginar;
- Java **NÃO** é lento (se você quiser);
- Fácil de usar Concorrência (*Threads*);
- Sintaxe próxima de **Javascript** (mas não são a mesma coisa!);

Mão na Massa

- **Variáveis**

- Java é fortemente tipada, ou seja você tem que declarar o tipo da variável, tipo o que será retornado, o tipo que será recebido, etc..

- `x = 1`

- `x = 'Aluno'`

- `x = [1,2,3,4]`

Errado em JAVA



- **Forma correta:**

- `int x = 1;`
- `String x = "Aluno";`
- `int[] x = {1,2,3,4};`

Tipos em JAVA

Em Java temos 8 tipos primitivos que podem ser usados como tipos de atributos, sendo:

byte - 8 bits (inteiro)

short - 16 bits (inteiro)

int - 32 bits (inteiro)

long - 64 bits (inteiro)

float - ponto flutuante de 32 bits (real)

double - ponto flutuante de 64 bits (real)

char - Unicode de 16 bits (1 caractere)

boolean – true / false (lógico)


Funções

- Em JAVA chamamos de métodos (mas é a mesma coisa)

Você tem que declarar **explicitamente** o que a função (ou método) recebe e retorna, inclusive os tipos dos dados. Veja abaixo um método que soma dois números.

Tipos de retorno

Tipos de entrada

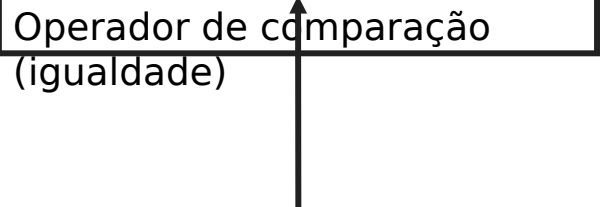


```
public int metodo (int a, int b){  
    return a + b;  
}
```

Controle: if

```
public class SeSenao {  
    public static void main(String[] args)  
    {  
        if(1 == 2)  
            System.out.println("Você jamais vai ler essa mensagem!!!");  
        if(1 == 1)  
            System.out.println("1 é igual a 1. Mas cê jura!");  
    }  
}
```

Operador de comparação
(igualdade)



Estrutura sequencial

Estrutura de controle

```
NotaAluno.java
1 import java.util.Scanner;
2
3 //Programa que recebe as notas de um aluno e calcula a média
4
5 public class NotaAluno {
6     public static void main(String[] args)
7     {
8         Scanner entrada = new Scanner(System.in);
9         float nota1, nota2, media;
10
11         System.out.println("Entre com a nota da 1ª avaliação: ");
12         nota1 = entrada.nextFloat();
13
14         System.out.println("Entre com a nota da 2ª avaliação: ");
15         nota2 = entrada.nextFloat();
16
17         media = (nota1+nota2)/2;
18
19         System.out.println("A nota da 1ª avaliação foi: " +nota1);
20         System.out.println("A nota da 2ª avaliação foi: " +nota2);
21         System.out.println("A média do aluno é: " +media);
22
23         //Estrutura de controle if/else
24         if(media >= 7.0)
25             System.out.println("O aluno foi aprovado por média!");
26         else
27             System.out.println("O aluno precisa realizar o exame final!");
28     }
29 }
```

Repetição com **while** (enquanto)

- Exemplo:
 - Crie um novo projeto chamado REPETICAO
 - Crie uma nova classe chamada TesteWhile

- Enquanto o valor de n for menor que 5
 - Imprime valor de n
 - Incrementa o valor de n em mais 1

```
1 public class TesteWhile {
2     public static void main(String[] args) {
3         int n = 0;
4
5         while (n < 5)
6             System.out.println(n++);
7     }
8 }
```

Qual foi a
saída?

Repetição: for

```
for(condição inicial; teste condicional ; após iteração){  
    //código  
}
```

1º - Condição inicial: geralmente se inicia o contador. Esse primeiro estágio SEMPRE acontece .

2º - Teste condicional: Caso seja verdade (TRUE), o código é executado.

3º - Após iteração: Ao término de execução do código, ocorre o fator de mudança, que geralmente é um incremento ou decremento, sempre após cada iteração do looping.

Exemplo 01 – contador de 1 a 10

```
public class ate10For {  
    public static void main(String[] args) {  
        for(int count = 1; count <=10; count++) {  
            System.out.println(count);  
        }  
    }  
}
```

Faça agora um programa que realize uma contagem regressiva

Faça agora um programa que realize uma contagem progressiva e regressiva no mesmo laço **for**

Classe

É uma entidade que descreve um conjunto de objetos com propriedades e comportamentos semelhantes e com relacionamentos comuns com outros objetos.

Usamos as classes para capturar o vocabulário do sistema que está em desenvolvimento.

Objetos

São instâncias de uma classe, ou seja, são quaisquer coisas na natureza que possuam propriedades (*atributos*) e comportamentos (*métodos*).

- Atributos: representam as características dos objetos
- Métodos: representam as ações que os objetos podem executar

Instâncias

Temos uma classe chamada **Casa**, seus atributos são: cor e número.



Instância
01



Instância
02



Instânci
a 3

Podemos observar 03 objetos (instâncias) de Casa.



```
Casa c1 = new Casa( );  
c1.numero = 12;  
c1.cor = "amarelo";
```



```
Casa c2 = new Casa( );  
c2.numero = 56;  
c2.cor = "vermelho";
```






```
Casa c3 = new Casa( );  
c3.numero = 72;  
c3.cor = "branca";
```

Temos então 03 instâncias da classe Casa (03 objetos: c1, c2 e c3).

Para instanciar objetos usamos o operador "new".

Todos os objetos, que são instâncias da mesma classe, compartilham uma semelhança de família suportando o mesmo comportamento.

Este comportamento é definido pelos métodos

	<pre>Casa c1 = new Casa(); c1.numero = 12; c1.cor = "amarelo";</pre>
	<pre>Casa c2 = new Casa(); c2.numero = 56; c2.cor = "vermelho";</pre>
	<pre>Casa c3 = new Casa(); c3.numero = 72; c3.cor = "branca";</pre>

Um comportamento de uma casa é “abrir porta”, portanto, podemos criar um método chamado `abrirPorta()`.

Um objeto da classe pode chamar o método:

- Exemplo: `c3.abrirPorta();`

o objeto `c3` da classe `Casa` invocou o método `abrirPorta()`

```
public class TesteAluno {  
    public static void main(String[] args) {  
  
        //Instanciando objetos  
        Aluno a1 = new Aluno();  
        Aluno a2 = new Aluno();  
        Aluno a3 = new Aluno();  
  
        //Criando um estado do objeto  
        a1.nome = "João";  
        a1.idade = 20;  
        a1.altura = 170;  
  
        a2.nome = "Rodrigo";  
        a2.idade = 27;  
        a2.altura = 165;  
  
        a3.nome = "Luiza";  
        a3.idade = 25;  
        a3.altura = 160;  
  
        //Imprimindo atributos  
        System.out.println(a1.nome + " tem " + a1.idade + " anos e " + a1.altura + " cm de altura");  
        System.out.println(a2.nome + " tem " + a2.idade + " anos e " + a2.altura + " cm de altura");  
        System.out.println(a3.nome + " tem " + a3.idade + " anos e " + a3.altura + " cm de altura");  
    }  
}
```

Exercício 01

Crie uma classe chamada Carro (Carro.java), em seguida crie uma classe chamada TesteCarro para instanciar 03 objetos do tipo Carro e imprimir na tela as informações sobre eles:

Obs: Carro possui os atributos marca (ex: Ford, Chevrolet, Renault), modelo

```
public class Carro {  
    String marca;  
    String modelo;  
    int ano;  
}
```

Arquivo:
Carro.java

