



Introdução ao Java

Lógica de Programação I

Paradigma de Programação **não** é Linguagem de Programação!



O que são paradigmas de programação?

Paradigmas **são como regras** de determinadas linguagens de programação, como devem se encaixar.

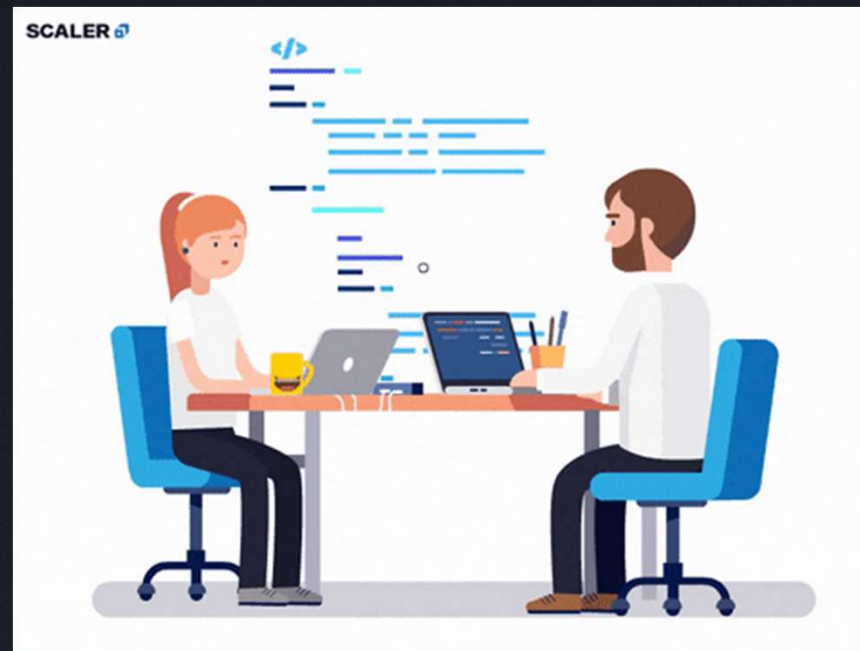
Podem ser entendidos como um estilo, modelo ou metodologia de programação, que **apontam a melhor forma de solucionar problemas usando determinada linguagem.**

```
var scrollHeight =  
  element.clientHeight + 0.02 * window.innerWidth  
window.scroll(0, scrollHeight);  
)
```

Quais são os tipos de paradigmas de programação?

Os tipos mais comuns

1. Paradigma Imperativo
2. Paradigma Funcional
3. Paradigma Lógico
4. Paradigma Orientado a Objetos



Introdução ao Java

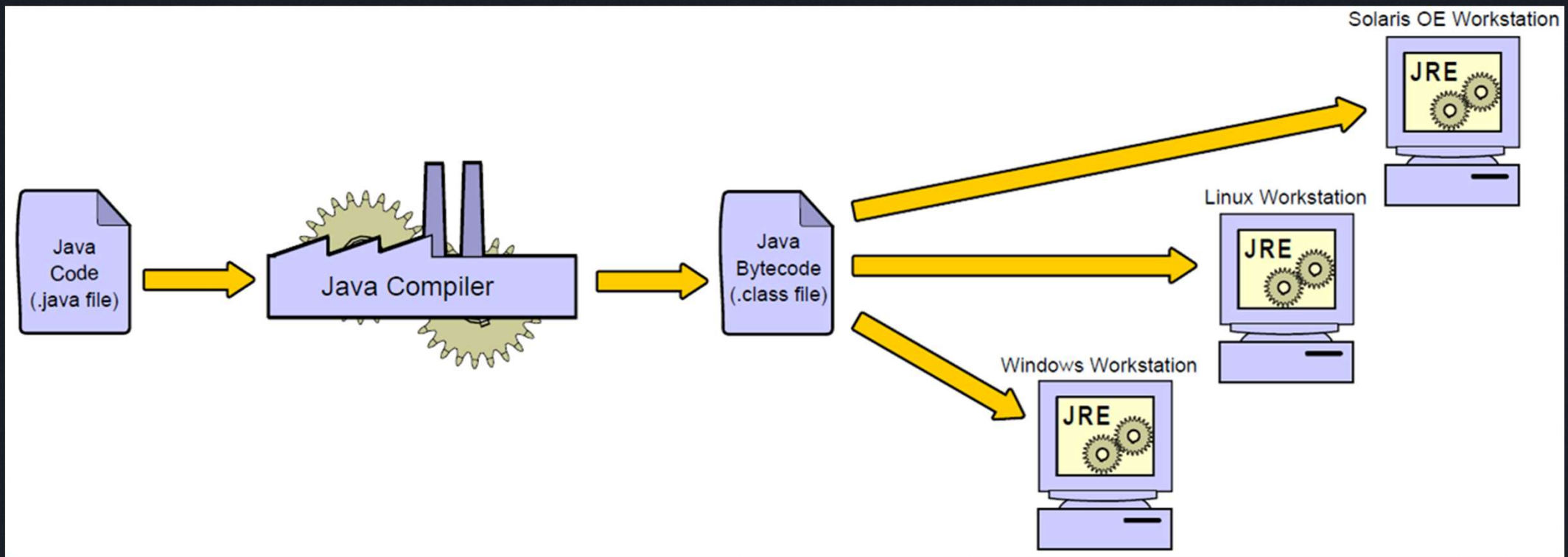
Sucesso da Internet



Introdução ao Java

Sucesso da Internet

"Write once, run everywhere"



Introdução ao Java

O que é **JDK** e **JRE**?

- *JDK* (*Java Development Kit*) é um pacote que inclui tudo o que é necessário para escrever aplicações.
- *JRE* (*Java Runtime Environment*) é a camada de software, que é executado sobre um sistema operacional, e nos fornece os recursos necessários para ter um ambiente que seja possível executar programas em Java.
- *JVM* (*Java Virtual Machine*) é o coração do Java, responsável por executar os programas em formato de *bytecode* e por fornecer a capacidade multi-plataforma.

Introdução ao Java

Por que Java?

- Java é uma das **linguagens** de Orientação a Objetos mais usadas;
- Java é **simples**, versátil, robusta e segura;
- **Portável** (independente de sistema operacional);
- **Gratuita** e *open-source*;
- **Popular**, rodeada por uma comunidade muito ativa;
- Muitas **ferramentas** disponíveis;
- Muita **documentação** disponível.

Introdução ao Java

O que é Java?

- Uma **tecnologia**;
- Uma **linguagem** de programação;
- Uma **plataforma** de desenvolvimento;
- Um **software** distribuído pela *Oracle*;
- Um **ambiente** de execução de programas;
- Uma **ilha** da Indonésia (é o **mar** ao norte da ilha).



Tipos, variáveis e constantes

Java é uma **linguagem de programação fortemente tipada**, sendo assim, cada variável declarada precisa ter um tipo associado, definindo os valores compatíveis que poderão ser armazenados pela variável.

Existem 8 tipos:

- **Primitivos:**
 - o 4 para números inteiros
 - o 2 para números de ponto flutuante
 - o 1 para caracteres
 - o 1 para lógicos, verdadeiro ou falso

Tipos

Tipos **Inteiros**

- **byte** (1 byte): -128 a 127
- **short** (2 bytes): -32.768 a 32.767
- **int** (4 bytes): -2.147.483.648 a 2.147.483.647
- **long** (8 bytes): -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807

Tipos

Tipos de **ponto flutuante**, **lógico** e **unicode**

- **float** (4 bytes): aprox. $\pm 3.40282347E+38$ (6-7 dígitos decimais significativos)
- **double** (8 bytes): aprox. $\pm 1.797691313486231570E+308$ (15 dígitos decimais significativos)
- **boolean**: true / false
- **char**: único caractere: 'A'

Variáveis

Variáveis são responsáveis por armazenar valores. Assim, fica mais fácil referenciar de forma dinâmica permitindo um comportamento mais flexível do nosso código.

```
boolean done;  
  
int age;  
  
long earthPopulation;  
  
double pi;
```

Variáveis

Strings

Além dos tipos primitivos, também existe também o tipo `String`. Qualquer sequência de caracteres (qualquer Unicode válido) representada entre aspas duplas `"` é uma string Java. Portanto, podemos declarar uma variável da seguinte forma:

```
String s = "this is a string";
```


Constantes

Como se pode imaginar, ao contrário das variáveis, onde podemos alterar o seu conteúdo, as constantes representam valores fixos e imutáveis. Sua principal contribuição é referenciar de forma fácil um valor fixo e, caso desejemos alterar, fazê-lo em apenas um ponto do nosso código.

```
final int QUANTIDADE_TENTATIVAS = 3;
```

Constantes

Se estivermos definindo cálculos trigonométricos, podemos definir uma constante para representar o PI. Na verdade, essa constante já existe e está disponível na classe `java.lang.Math`. Para visualizar seu valor, tente o seguinte código:

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println(Math.PI);  
        // expected result 3.141592653589793  
    }  
}
```

Enums

Outro tipo especial do Java são as *enumerations*, ou apenas **enums**. Esse tipo representa valores finitos de uma determinada representação. Por exemplo, dias da semana.

Podemos escrever um *enum* contendo os sete dias da semana, da seguinte forma:

```
public enum DiaDaSemana {  
    SEGUNDA, TERCA, QUARTA, QUINTA, SEXTA  
}
```


Wrapper Classes

As *Wrapper Classes* em Java são um conjunto de classes que fornecem um "wrapper" para os tipos primitivos. Em outras palavras, elas permitem que você trate tipos primitivos como objetos. Cada tipo primitivo tem sua classe de wrapper correspondente.

Wrapper Classes

Wrapper Classes e os tipos primitivos correspondentes:

1. **Byte**: Correspondente ao tipo primitivo **byte**
2. **Short**: Correspondente ao tipo primitivo **short**
3. **Integer**: Correspondente ao tipo primitivo **int**
4. **Long**: Correspondente ao tipo primitivo **long**
5. **Float**: Correspondente ao tipo primitivo **float**
6. **Double**: Correspondente ao tipo primitivo **double**
7. **Character**: Correspondente ao tipo primitivo **char**
8. **Boolean**: Correspondente ao tipo primitivo **boolean**

Obrigado!