

# Programação Orientada a Objetos em Java

Prof. Rafael Guterres Jeffman  
[rafael.jeffman@gmail.com](mailto:rafael.jeffman@gmail.com)

# Objetivos

- Revisar os conceitos básicos necessários à programação de computadores.
- Revisar os conceitos de Orientação a Objetos.
- Revisar a implementação desses conceitos na Plataforma Java.
- Revisar as técnicas básicas de programação utilizando Java e Orientação a Objetos.

# Paradigmas de Programação

- Imperativa
- Estruturada
- Lógica
- Funcional
- Orientada a Objetos
- Orientada a Aspectos
- Orientada a Eventos
- Dataflow

# Variáveis

- Uma **variável** é um **endereço de memória**, associado a um **identificador**, onde é armazenado um **valor**.

# Referência

- Uma **referência** é um **valor** que permite a um **programa** acessar indiretamente um **dado**.

# Variável vs. Referência

Variável

$a = 1$

$b = 2$

$c = a$

a

1

c

1

b

2



# Variável vs. Referência

Variável

$a = 1$

$b = 2$

Referência

$c = a$

a

1

c

b

2

# Programação Orientada a Objetos (OOP)

- Foi uma evolução natural da programação estruturada, modular e do uso de tipos de dados abstratos.
- OOP não é um “jeito de programar”, mas uma forma de analisar o problema.
- Ao invés de separar o programa em código e dados, na OOP, o código e os dados estão reunidos em um “objeto”.



# Classes

- São abstrações de objetos.
- Definem o **estado** e o **comportamento** de um objeto.
- Classes relacionam com outras classes criando sistemas mais complexos.
- Classes são a descrição dos objetos criados.

# Objetos

- Objetos são instâncias de uma classe.
- Objetos não existem durante o desenvolvimento do software, apenas durante a sua execução.
- Objetos possuem um tempo de vida.
- Objetos podem assumir diferentes “tipos” durante o seu tempo de vida, mas o seu comportamento é definido pela sua classe.

# Atributos

- O conjunto de atributos de um objeto determina o estado deste objetos.
- Atributos podem ser de leitura/escrita, apenas leitura, e, às vezes, apenas escrita.
- Para acessar um atributo, utilizamos mensagens, que com o tempo se convencionou chamar de “getters” (get/leitura) e “setters” (set/escrita).
- Não confunda atributos com variáveis. Um atributo pode não ser armazenado, mas calculado apenas quando necessário.

# Mensagens

- Uma mensagem é enviada por um objeto a outro, como requisição de um serviço.
- Normalmente, uma mensagem possui uma assinatura própria.
- Mensagens são a única forma de objetos se comunicarem em um sistema.

# Métodos

- Métodos implementam o comportamento de objetos.
- Métodos são executados em resposta a uma mensagem recebida pelo objeto.
- Métodos podem receber parâmetros e retornar valores.
- Assim como na programação estruturada, um método recebe um conjunto de dados, processa-os e retorna um novo conjunto de dados.

# Tipos de Dados

- Em linguagens de programação orientada a objetos, normalmente todos os tipos de dados representam objetos.
- Em algumas linguagens OOP, existem dois tipos de dados, os “tipos abstratos de dados” (as classes) e os “tipos fundamentais” (números, boolean, caracteres).

# Em Java...

- Java possui tipos fundamentais de dados (que não criam objetos) e tipos abstratos de dados (classes).
- Todos os tipos fundamentais de dados são utilizados como variáveis.
- Todos os objetos trabalham como referências.

# Java não copia objetos!

- Uma atribuição, em Java, envolvendo objetos, é uma atribuição de **referência**, logo, o objeto (**valor**) não é copiado.
- Não existe cópia implícita de objetos em Java, é necessário enviar uma mensagem de cópia para um objeto.



# Declaração, Instancição e Atribuição

- Tome como exemplo o código em Java abaixo:
  - `Pessoa contato = new Pessoa("João", "555-3124");`
- Neste exemplo, estão ocorrendo três operações:
  - Declaração da variável 'contato'
  - Instancição de um objeto Pessoa
  - Atribuição da referência do objeto criado à 'contato'

# Impacto no programa

```
int a = 2;  
int b = 1;  
int c = a;
```

```
String x = "Hello";  
String y = "World";  
String z = a;
```

a	2	
b	1	
c	2	
x	"Hello"	
y	"World"	z

# Passagem de Parâmetros

- Toda passagem de parâmetro em Java é feita **por valor**.
- Isto implica em uma **cópia** do valor sendo passado como argumento para o **método**.
- Como o “*valor*” de uma variável que possui como tipo de dado um objeto é na realidade uma **referência**, apenas a referência é copiada, não o objeto em si.