

Linguagens de Programação

Java Orientado a Objetos

Prof. Waldeck Lindoso Jr.

Programação Orientada a Objeto (POO)

A **Programação Orientada a Objeto (POO)** tem como pilar teórico o conceito de **objeto**, ou seja, um sistema orientado a objeto é um conjunto de objetos que representam os seres e coisas do mundo real, interagindo computacionalmente com as mesmas características e comportamentos reais.

À estrutura computacional que define o modelo de um objeto do mundo real damos o nome de **classe**, e a partir dela cópias são construídas para serem utilizadas para cada objeto real que tenho no meu sistema.

As **características** dos objetos do mundo real são chamadas de **atributos** no mundo computacional, enquanto a seus **comportamentos** chamamos métodos



Classes

Classes são estruturas que definem as características e comportamentos dos seres ou coisas do mundo real; quando implementados em Java nas classes, essas características passam a se chamar atributos, e os comportamentos se transformam nos nossos métodos.



Objetos

Objetos são as instâncias das minhas classes, os seres que existem no meu mundo real quando são inseridos no mundo computacional.



Atributos

Atributos são as características inerentes ao meu ser ou coisa do mundo real quando são definidas no mundo computacional.



Métodos

Métodos são os comportamentos ou funções de um objeto do mundo real na forma como ele é tratado no mundo computacional.



Por exemplo...

Um objeto carro no mundo real tem as propriedades cor, ano, número de portas e tipo de combustível, entre outras. Além disso, tem os comportamentos acelerar, frear, virar à esquerda e outros. Assim, quando modelamos esse carro computacionalmente para um jogo de corridas, por exemplo, definimos uma classe carro cujas propriedades são transformadas nos atributos cor, ano, número de portas, tipo de combustível e seus comportamentos são transformados no código do programa nos métodos acelerar, frear, virar à esquerda e outros.



Classes e Objetos

Uma das primeiras confusões que quem está começando a estudar orientação a objeto faz é confundir os conceitos de classes e objetos. Vamos ver mais um pouco desses conceitos para não deixarmos dúvidas.

Classe representa o modelo do que queremos utilizar no nosso sistema; seria como um gabarito, um protótipo, a planta de uma casa ao realizarmos uma construção, a receita de um bolo.



Classes e Objetos

Veja o exemplo da tabela abaixo, na qual temos as definições de uma classe que representa o carro do mundo real; ao lado, os objetos carros que iremos implementar no nosso sistema:

CLASSE CARRO		OBJETO CARRO A	OBJETO CARRO B
Atributos de objeto	Marca	Ford	Mitsubishi
	Modelo	Fiesta	L-200
	Cor	branco	azul royal
	Combustível	gasolina	diesel
Métodos	ligar		
	acelerar		
	frear		

Classes e Objetos

Assim, durante a execução do sistema, a cada carro que cadastramos uma cópia nova da classe carro é criada, e damos a ela características que irão diferenciá-la dos demais carros, ou seja, criamos a cada vez um novo objeto do tipo de carro. Ah! Então quer dizer que classe é um tipo? **Isso mesmo.**

CLASSE CARRO		OBJETO CARRO A	OBJETO CARRO B
Atributos de objeto	Marca	Ford	Mitsubishi
	Modelo	Fiesta	L-200
	Cor	branco	azul royal
	Combustível	gasolina	diesel
Métodos	ligar		
	acelerar		
	frear		

Classes e Objetos

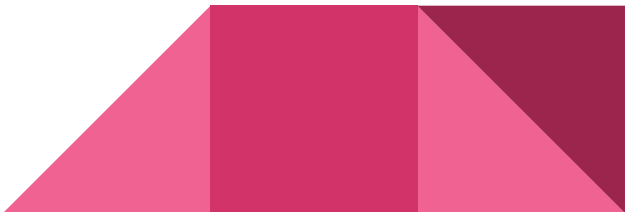
O principal benefício da utilização da programação por meio de classes é a reutilização do código, pois a cada objeto criado você não precisa criar sua estrutura novamente.



Criando uma classe

Vamos criar o exemplo baseado na tabela anterior, criando uma classe **Carro** com seus atributos, marca, modelo, cor e combustível.

```
3 public class Carro {  
4  
5     String marca;  
6     String modelo;  
7     String cor;  
8     String combustivel;
```



Criando uma classe

A partir de um modelo de uma classe, para criar cada objeto, utilizamos o operador new. Por exemplo, para criar uma instância do Carro “carro” a partir da classe Carro, utilizamos o seguinte código:

```
3 public class CarroTeste {  
4     public static void main(String[] args) {  
5         Carro carro = new Carro();  
6     }  
7 }
```

Criando uma classe

O operador **new** reserva (aloca) a memória para o objeto conforme a estrutura de sua classe e retorna uma referência para essa área que foi alocada. O código que vem após o operador new é o que chamamos de construtor da classe e tem o mesmo nome da classe seguido de () (abre e fecha parênteses); dentro deles podem vir parâmetros, dependendo da classe a ser utilizada e de sua definição.



Criando uma classe

Através da variável “carro”, podemos acessar objeto recém criado para alterar sua marca, modelo, cor ou até mesmo seu tipo de combustível.

```
3 public class CarroTeste {  
4     public static void main(String[] args) {  
5         Carro carro;  
6         carro = new Carro();  
7  
8         carro.marca = "Ford";  
9         carro.modelo = "Ka";  
10        carro.cor = "Branco";  
11        carro.combustivel = "Gasolina";  
12    }  
13 }
```

Criando uma classe

Através da variável “carro”, podemos acessar objeto recém criado para alterar sua marca, modelo, cor ou até mesmo seu tipo de combustível.

```
3 public class CarroTeste {  
4     public static void main(String[] args) {  
5         Carro carro;  
6         carro = new Carro();  
7  
8         carro.marca = "Ford";  
9         carro.modelo = "Ka";  
10        carro.cor = "Branco";  
11        carro.combustivel = "Gasolina";  
12    }  
13 }
```


Hands on!



Obrigado!

