

Desenvolvimento para Servidores 2

Prof. Orlando Saraiva Júnior
Orlando.nascimento@fatec.sp.gov.br

Desenvolvimento para Servidores 2

Orientação a objetos é uma abordagem para a concepção de sistemas, um paradigma de programação. Essa abordagem envolve desde a modelagem do sistema até sua programação por meio do uso e relacionamento entre esses objetos.

Um objeto é uma instância de classe.

Classe uma forma de definir um tipo de dado em uma linguagem orientada a objeto. Ela é formada por dados e comportamentos.

É um conceito que encapsula abstrações de dados e procedimentos que descrevem o conteúdo e o comportamento de entidades do mundo real representadas por objetos.

É um “gabarito”. Exemplo, se a classe é a planta de uma casa, o conjunto de casas contruídas seriam os objetos.

Atributos são características da instância da classe (objeto).

São conhecidos como “variáveis” de classe, e podem ser divididos em 2 tipos: atributos de instância e de classe.

Comportamentos, ações e reações dos objetos.

Os métodos determinam o comportamento dos objetos de uma classe e são análogos a funções ou procedimentos da programação estruturada.

Define habilidades/comportamentos do objeto.

Um exemplo

Guilherme

```
public class Carro {  
    Double velocidade;  
    String modelo;  
    public Carro(String modelo) {  
        this.modelo = modelo;  
        this.velocidade = 0;  
    }  
    public void acelerar() {  
        /* código do carro para acelerar */  
    }  
    public void frear() {  
        /* código do carro para frear */  
    }  
    public void acenderFarol() {  
        /* código do carro para acender o farol */  
    }  
}
```

Um exemplo

Jean Carlo

```
<?php
class MyClass {
    public $prop1 = "Sou um propriedade de classe!";
    public function setProperty($newval) {
        $this->prop1 = $newval;
    }
    public function getProperty() {
        return $this->prop1 . "<br />";
    }
}

$obj = new MyClass;
echo $obj->prop1;
?>
```


Um exemplo

Jean Lucas

```
public class Conta {  
    private Double saldo;  
    public void setSaldo(Double saldo) {  
        this.saldo = saldo;  
    }  
    public Double getSaldo() {  
        return saldo;  
    }  
    public void depositar(Double valor){  
        saldo += valor;  
    }  
    public void verificaSaldo(){  
        System.out.println("Valor do Saldo: "+getSaldo());  
    }  
}
```

Um mecanismo que permite fazer inicializações no objeto assim que ele é declarado com o new.

Construtor determina que ações devem ser executadas quando da criação de um objeto.

Construtores são basicamente funções de inicialização de uma classe, as quais são invocadas no momento em que objetos desta classe são criadas. Eles permitem inicializar campos internos da classe e alocar recursos que um objeto da classe possa demandar, tais como memória, arquivos, semáforos, soquetes, etc.

Construtores

```
public class Carro{

/* CONSTRUTOR DA CLASSE Carro */
    public Carro(){
        //Faça o que desejar na construção do objeto
    }
}

public class Aplicacao {
    public static void main(String[] args) {
        //Chamamos o construtor sem nenhum parâmetro
        Carro fiat = new Carro();
    }
}
```

Formas de definir a visibilidade das propriedades e dos métodos de um objeto.

`public` → permite acesso a qualquer código externo a classe.

`protected` → permite acesso às classes filhas, mas proíbe a qualquer outro acesso externo.

`private` → proíbe qualquer acesso externo à própria classe, inclusive das classes filhas.

Relacionamento entre objetos

- Associação
- Composição
- Agregação
- Herança

Agradecimentos



GUILHERME PALMEIRA BISPO TAIRA

JEAN LUCAS SEGOVIA

JEAN CARLO ROSA TENCA JUNIOR

MARIA EMILIA OLIVEIRA

RAFAEL ALVES DOS SANTOS