



Programação Orientada a Objetos

Unidade 2 - Conceitos básicos de POO



Prof Ronald Chiesse de Souza
ronaldsouza@dcc.ufrj.br



Conceitos importantes de POO

Classes

Objetos

Atributos

Métodos



Programação Orientada a Objetos

- É um paradigma de programação cujo foco está na representação de “objetos” dentro do programa.
- Diferente da programação procedural, o objetivo não é criar um “fluxo” de código que seja executado do começo ao fim, mas sim pensar em cada “objeto” de forma independente, assumindo-se que o fluxo de regras pode variar conforme o conjunto de dados analisado.
- O conceito de programação orientada a objetos existe desde a década de 60, porém, só começou a se popularizar na década de 80. E hoje é o paradigma dominante.

Classes

- Molde dos objetos
- São como “contratos” que definem o que se pode esperar de um objeto.
- Define as variáveis/atributos e funções/métodos que os objetos vão poder exercer
- São concebidas a partir de entidades do mundo real (carros, animais, pessoas, brinquedos etc) ou até de conceitos abstratos (ideias, tarefas, sonhos etc).

Objetos

- “materialização” das classes; são chamados **instâncias** da classe
- **Instanciar** uma classe é criar um novo objeto a partir dela.
- É possível instanciarmos múltiplos objetos a partir da mesma classe, cada um com propriedades distintas.
- Embora diferentes objetos de uma mesma classe possam ter diferentes propriedades, a classe estabelece um molde em comum.



Fonte: <https://polymerclaylatvia.com/silicone-mold-chocolate-a-3-4-2-6-cm>

Classes x Objetos

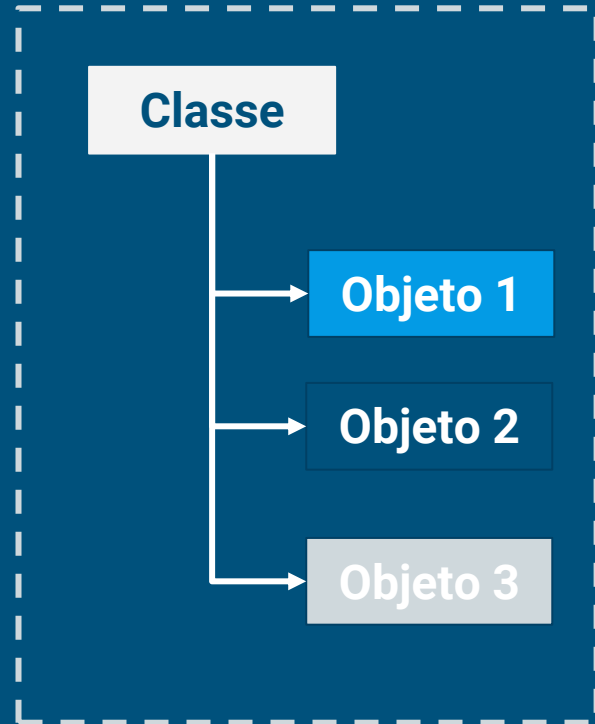
| Classe | Objeto |
|--|---|
| É um template usado para criar objetos | É uma materialização de uma classe |
| Só é declarada uma vez | Podemos instanciar múltiplos objetos da mesma classe |
| Não pode ser armazenada numa variável | Pode ser armazenado em variáveis |

Atributos

- Atributos são variáveis que pertencem a uma classe ou aos objetos dessa classe, mas nunca ao mesmo tempo. Um atributo ou é de classe ou é de objeto.
- Só existem, por padrão, dentro de um objeto, mas podem também pertencer à classe como um todo.
- Podem ser de qualquer tipo: inteiros, racionais, strings, listas, dicionários e até outras classes.
- Todas as instâncias (i.e. objetos) de uma classe possuem os mesmos atributos, mas estes não necessariamente terão os mesmos valores.

Instanciação de objetos

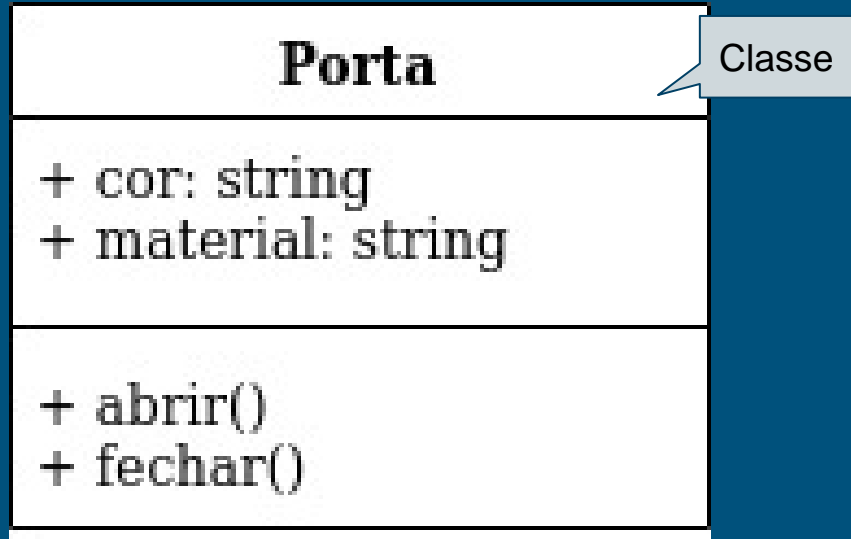
- Cada instanciação gera um objeto novo
- Eles podem ter os mesmos valores de atributos, mas são instâncias diferentes



Métodos

- Métodos são funções exclusivas de uma classe.
- Por padrão, eles só são acessíveis através dos objetos instanciados.
- Geralmente podem ser classificados como **mutadores** (quando mudam atributos do objeto) e **acessadores** (quando apenas acessam os atributos). Nem todo método está nessa classificação! Por ex.: um método que apenas imprime “Bom dia” incondicionalmente, não acessa e nem muda atributos.
- Todos os objetos instanciados da mesma classe sempre têm os mesmos métodos.

Exemplo de Classe vs. objeto



Implementação em Java da Classe

- O nome da classe começa sempre com letra maiúscula
- Atributos e métodos são do objeto por padrão
- Podemos criar atributos e métodos de classe usando a palavra **static**
 - Os atributos de classe tem um único valor para todos os objetos (podendo ser modificados)
 - Os métodos de classe são chamados sem a instanciação do objeto

```
class Porta{
    String cor;
    String material;
    public static int qtde = 0;

    public void abrir(){
        // codigo aqui
    }
    public void fechar(){
        // codigo aqui
    }
    public static int qtdePortas(){
        return Porta.qtde;
    }
}
```

Implementação em Java da Classe

- Podemos criar métodos com o mesmo nome e parâmetros diferentes
 - **Sobrecarga de métodos**, um tipo de polimorfismo
- Podemos criar um método especial que é chamado quando instanciamos o objeto, com ou sem parâmetros
 - **Método construtor**, nome igual ao da classe
 - Pode ter mais de um método construtor

```
class Porta{
    String cor;
    String material;

    public Porta(){
        // método construtor
    }
    public Porta(String cor_atual){
        this.cor = cor_atual;
    }

    public void abrir(){
        // código aqui
    }

    public void fechar(){
        // código aqui
    }
}
```

Como definir uma classe?

- Fase de planejamento:
 - Quais atributos os objetos dessa classe devem possuir?
 - Quais métodos (ações) eles devem ser capazes de fazer?
- Vamos criar a classe Cliente juntos

Como definir uma classe?

- Atributos com - são atributos privados, que só podem ser acessados pelo próprio objeto
 - É necessário fazer métodos de acesso para todos os atributos privados (get)
 - Se fizer sentido para a classe, fazer também métodos de atualização desses atributos (set)
- Atributos e métodos sublinhados são atributos e métodos de classe (**static**)
- Atributos em caixa alta são constantes (**final**)
 - Em Java, constantes são atributos que recebem um primeiro valor na declaração ou no construtor do objeto e não pode ser alterado depois

| Cliente |
|---|
| - nome: String - cpf: String - tem_cartao: boolean - credito_disponivel: double - num_cartao: String + <u>qtde_clientes</u> : int = 0 |
| + Cliente(String, String) + Cliente() + atualizaCpf(String) + acessaCpf(): String + getNome(): String + setNome(String) + temCartao(): boolean + adicionaCartao() + getCredito(): double + atualizaLimiteCredito(double) + getNumCartao() + <u>getQtdeClientes</u> (): int |