Programação Orientada a Objetos

Módulo 1

Vaux Gomes [1]

[1]Instituto Federal de Educação, Ciência e Tecnologia do Ceará
Campus Jaguaribe

7 de Agosto de 2022

Sumário

Elementos da OO

O objeto

Objeto

Quase qualquer *substantivo* pode ser razoavelmente representado como um objeto de software em termos:

- Dos **atributos** (por exemplo, nome, cor e tamanho) e;
- Dos **comportamentos** (por exemplo, calcular, mover e comunicar).

Exemplos

- Mouse, Controle, Cadeira, Porta
- Compromisso agendado, a venda de um produto, botão na tela do celular

Elementos da OO

O objeto

Atenção

Os objetos da POO não se restringem a elementos físicos, por isso é importante ressaltar que o programador deve estar atento e avaliar se “uma coisa"possui atributos e comportamentos para definir se algo é um objeto ou não.

Elementos da OO

Classe

Classe

- Uma classe é uma estrutura que abstrai um conjunto de objetos com características similares.
- Uma classe define o **comportamento** de seus objetos – através de métodos – e os **estados** possíveis destes objetos – através de atributos.

Elementos da OO

Classe

- Podemos usar os exemplos de objetos anteriores
- Veremos também que poderemos definir classes apenas com o propósito de padronizar e organizar o nosso código.
- O processo de descrever ou, de fato, escrever o código das nossas classes chama-se **declaração**.

Elementos da OO

Classe

Comportamento

Os comportamentos de um objeto são definidos pela classe. Por exemplos:

1. Se tivéssemos uma classe `Carro` poderíamos observar vários comportamentos relacionados à classe, como:
   - Ligar; Desligar;
   - Acelerar; Frear; etc.
2. Se a nossa classe fosse uma `Caneta`:
   - Escrever; Rabiscar;
   - Pintar; etc.

Estes comportamentos dependeriam dos estados dos objetos.

Elementos da OO

Classe

Estado

O estado de um objeto é observado a partir dos valores em seus atributos. Por exemplos:

1. Se tivéssemos uma classe `Carro` e verificássemos ele não possui bateria, podemos dizer que o estado do nosso carro é *quebrado*, por exemplo.
2. Se a nossa classe fosse uma `Caneta` e ela possuísse tinta e pudesse escrever, podemos dizer que a caneta está no estado de *pronta para escrever*.

Atributos

Atributos são as características de um objeto, essas características também são conhecidas como variáveis.

- Em Java, cada atributo possui um tipo pré-declarado
- Os tipos incluem os tipos primitivos (int, double, ...), classes, classes abstratas e interfaces[1]
- Os atributos podem ou não serem inicializados.
  - Erros podem ser ocasionados quando tentamos usar um atributo não inicializado.
- Alguns tipos primitivos já possuem um valor *default*.
- É possível declararmos uma classe sem atributos.

Elementos da OO

Atributos

Exemplo

Para a classe `Carro` poderíamos ter vários atributos de **tipo primitivo** como:

Atributo	**Tipo**
Número de pneus	Inteiro
Total de gasolina	Ponto flutuante
Nome do carro	Vetor de caracteres

Poderíamos também contar com atributos **definidos por outras classes** como:

Atributo	**Tipo**
Motor	Classe Motor
Bateria	classe Bateria

Elementos da OO

Atributos

Atenção

Atributos do tipo Classe contém seus próprios atributos, métodos e estados. A integração de todas as partes dependeria do múltiplos fatores.

Elementos da OO

Atributos

```java
/** Carro.java */
public class Carro {
    String cor;
    String modelo;
    int ano;
    Motor motor;
}

/** Motor.java */
public class Motor {
    boolean ligado;
    float velocidade;
}
```

[1] Algumas linguagens aceitam funções como atributos.

Métodos

Métodos são as ações que os objetos podem exercer quando solicitados, onde podem interagir e se comunicarem com outros objetos

- Os métodos explicitam os comportamentos que os objetos possuem
  - Não necessariamente precisamos definir e implementar todos os possíveis métodos que uma classe poderia ter.
  - Implementaremos o que for de interesse e também aproveitaremos o que for de interesse

Elementos da OO

Métodos

- Os métodos podem retornar e receber valores e objetos (em outras palavras, vários tipos primitivos e classes).
  - Os métodos são funções que têm acesso aos estados (atributos) da classe
  - O acesso aos estados é feito por meio da palavra reservada **this**

Elementos da OO

Métodos

```
public class Carro {
    /* Atributos... */

    public void ligar() { /* Método que liga o motor */ }
    public void desligar() { /* Método que desliga o motor */ }
    public void acelerar() { /* Método que aumenta a velocidade do motor */ }
    public void frear() { /* Método que diminui a velocidade do motor */ }

    /* Método que escreve o nome e o ano do carro na tela */
    public void buzinar() {
        System.out.println("Bibi: carro " + this.modelo + ", ano " + this.ano);
    }
}
```

Instanciação

Instanciar é o processo de **reservar memória** para que possamos manipular um objeto.

- Todo objeto possui um método especial chamado **construtor**. Estes são usados na **instanciação**.
- Uma mesma classe será usada para gerar vários objetos
- Cada um dos objetos serão tratados de maneira diferente pois terão endereços de memória diferentes (ou seja, os valores de seus atributos são diferentes).
- Quem faz o papel de instanciador em Java é o **new**.
  - O **new** retorna uma referência para o objeto criado na posição de memória reservada

Instanciação

Construtores

- O método construtor possui algumas regras:
  1. O método construtor **deve ter o mesmo nome da classe** em que reside
  2. O método construtor não possui retorno, isto é, seu tipo de saída deve ser sempre **void**.
  3. O método construtor em Java **não pode ser** *abstract*, *final*, *static* ou *synchronized*.

Instanciação

Construtores

```
public class Carro {
    /* ...Atributos... */
    /* Construtor Padrão */
    public void Carro() {
        this.cor = "";
        this.modelo "";
        this.ano = -1;
        this.motor = null;
    }

    /* Construtor Completo */
    public void Carro(String cor, String modelo, int ano, Motor motor) {
        this.cor = cor;
        this.modelo = model;
        this.ano = ano;
        this.motor = motor;
    }

    /* ...Métodos... */
}
```

Instanciação

Instanciando com o New

```
// Construtor vazio
Carro carro1 = new Carro();

// Construtor completo
Motor motor = new Motor();
Carro carro2 = new Carro("Amarelo", "Fusca", 1980, motor);

carro1.buzinar(); /* Bibi: carro , ano -1 */
carro2.buzinar(); /* Bibi: carro Fusca, ano 1980 */
```

Exercício

1. Instanciar os 3 carros ao lado:
   - Gurgel, verde, 1964
   - Fusca, amarelo, 1980
   - Uno, azul, 1998
2. Acelerar duas vezes cada carro
   - Crie um método acelerar e frear na classe Motor
   - Acelere o motor a partir da classe Carro
3. Buzinar com cada carro