
Introdução à Programação Orientada a Objetos

Sumário

- O que é um programa?
- Modelo;
- Programação Orientada a Objetos;
- Exemplo;
- Encapsulamento;
- Bibliografia.

O que é um programa?

- As tarefas (cadastramento, armazenamento, localização de dados, etc) **automatizadas** pelo computador são possíveis através de programas;
- **Programas** são conjuntos de comandos e regras que um programador deve conhecer para poder manipular os recursos (hardware) de um computador;
- Programas são escritos usando **linguagem de programação**, que definem regras específicas e bem determinadas e um conjunto de operadores e comandos que podem ser usados.

O que é um programa?

- **Programas processam dados:** valores em uma conta bancária, caracteres entrados por um teclado, pontos de uma imagem, etc;
- O paradigma de programação orientado a objetos considera que os dados a serem processados e os mecanismos de processamento destes dados devem ser **considerados em conjunto**;
- A **criação de modelos** que representam conjuntamente dados e operações nestes dados é a solução.

Modelos

- Modelos são representações simplificadas de objetos, pessoas, itens, tarefas, processos, etc. usados comumente por pessoas no seu dia-a-dia independente do uso de computadores.
- Para simplificar o uso de modelos em tarefas comuns, consideramos o Restaurante Caseiro Hipotético, que serve refeições por quilo, e onde o gerente, que fica na balança e no caixa, anota os pesos dos pratos dos clientes e os pedidos que os garçons trazem em um quadro-branco.

Modelos

- A figura abaixo mostra o quadro-branco do Restaurante Caseiro Hipotético:

Restaurante Caseiro Hipotético		
Mesa 1 <input type="checkbox"/> Kg refeição <input type="checkbox"/> Sobremesa <input type="checkbox"/> Refrigerante lata <input type="checkbox"/> Suco com Leite <input type="checkbox"/> Cerveja	Mesa 2 <input type="checkbox"/> Kg refeição <input type="checkbox"/> Sobremesa <input type="checkbox"/> Refrigerante lata <input type="checkbox"/> Suco com Leite <input type="checkbox"/> Cerveja	Mesa 3 <input type="checkbox"/> Kg refeição <input type="checkbox"/> Sobremesa <input type="checkbox"/> Refrigerante lata <input type="checkbox"/> Suco com Leite <input type="checkbox"/> Cerveja
Mesa 4 <input type="checkbox"/> Kg refeição <input type="checkbox"/> Sobremesa <input type="checkbox"/> Refrigerante lata <input type="checkbox"/> Suco com Leite <input type="checkbox"/> Cerveja	Mesa 5 <input type="checkbox"/> Kg refeição <input type="checkbox"/> Sobremesa <input type="checkbox"/> Refrigerante lata <input type="checkbox"/> Suco com Leite <input type="checkbox"/> Cerveja	Mesa 6 <input type="checkbox"/> Kg refeição <input type="checkbox"/> Sobremesa <input type="checkbox"/> Refrigerante lata <input type="checkbox"/> Suco com Leite <input type="checkbox"/> Cerveja

Modelos

- O modelo do restaurante representa **certos dados ou informações**, que no caso são os itens e a quantidade dos pedidos;
- Um modelo comumente **contém operações ou procedimentos associados** a ele, por exemplo:
 - Inclusão de um pedido para uma mesa;
 - Modificação do status de um pedido de uma mesa;
 - Encerramento de um pedido de uma mesa e
 - Apresentação da conta para os clientes.

Modelos

- Modelos podem conter submodelos e ser parte de outros modelos:
 - o quadro-branco que representa um restaurante pode ser composto de diversos quadrados no quadrado que representa mesas ou comandas;
 - cada uma contendo os dados relativos aos pedidos daquela mesa e ações correspondentes;
- A simplificação inerente aos modelos é, em muitos casos, necessária: dependendo do contexto, algumas informações devem ser ocultadas ou ignoradas;

Modelos

- Em um modelo, a representação das **informações sobre uma pessoa** pode ser feita de maneira diferente dependendo do contexto, veja:
 - **Pessoa como empregada de empresa:** seriam necessários o nome, o cargo, salário, horasExtras. Este modelo poderia conter as operações `calculaSalário` e `aumentaSalário`.
 - **Pessoa como paciente de uma clínica:** seriam necessários o nome, o sexo, a idade, a altura, o peso e o `históricoConsultas`. Este modelo poderia conter a operação `verificaObesidade`.
 - **Pessoa como contato comercial:** seriam necessários o nome, o telefone, o cargo e a empresa. Este modelo poderia conter a operação `mostrarTelefone`.

Modelos

- Modelos podem ser reutilizados para representar diferentes objetos, pessoas ou itens.
 - O mesmo modelo `contatoComercial` poderia ser utilizado para representar cada um dos contatos de um estabelecimento comercial.
- Os contatos podem ser representados pelo mesmo modelo, mas os dados individuais de cada um podem ser diferentes, criando a necessidade de diversos exemplos de cada modelo - assim João, Pedro e Maria;
- A criação e uso de modelos é uma tarefa natural e a extensão desta abordagem à programação deu origem ao paradigma **Programação Orientada a Objetos.**

Programação Orientada a Objetos

- **Programação Orientada a Objetos - POO** é "um paradigma de programação onde se usam classes e objetos, criados a partir dos modelos descritos anteriormente, para representar e processar dados usando programas".
- **Classes** são estruturas das linguagens de programação orientadas a objetos **que contém** um determinado modelo, os dados que devem ser representados e as operações que devem ser efetuadas com estes dados.

Programação Orientada a Objetos

- Em muitos casos, as classes são somente moldes ou formas que representam os modelos abstratamente;
- Para representação de dados específicos usando classes deste tipo será necessárias a criação de **objetos ou instâncias** desta classe;
- Um objeto ou instância é uma **materialização da classe**, e assim pode ser usado para representar dados e executar operações.

Programação Orientada a Objetos

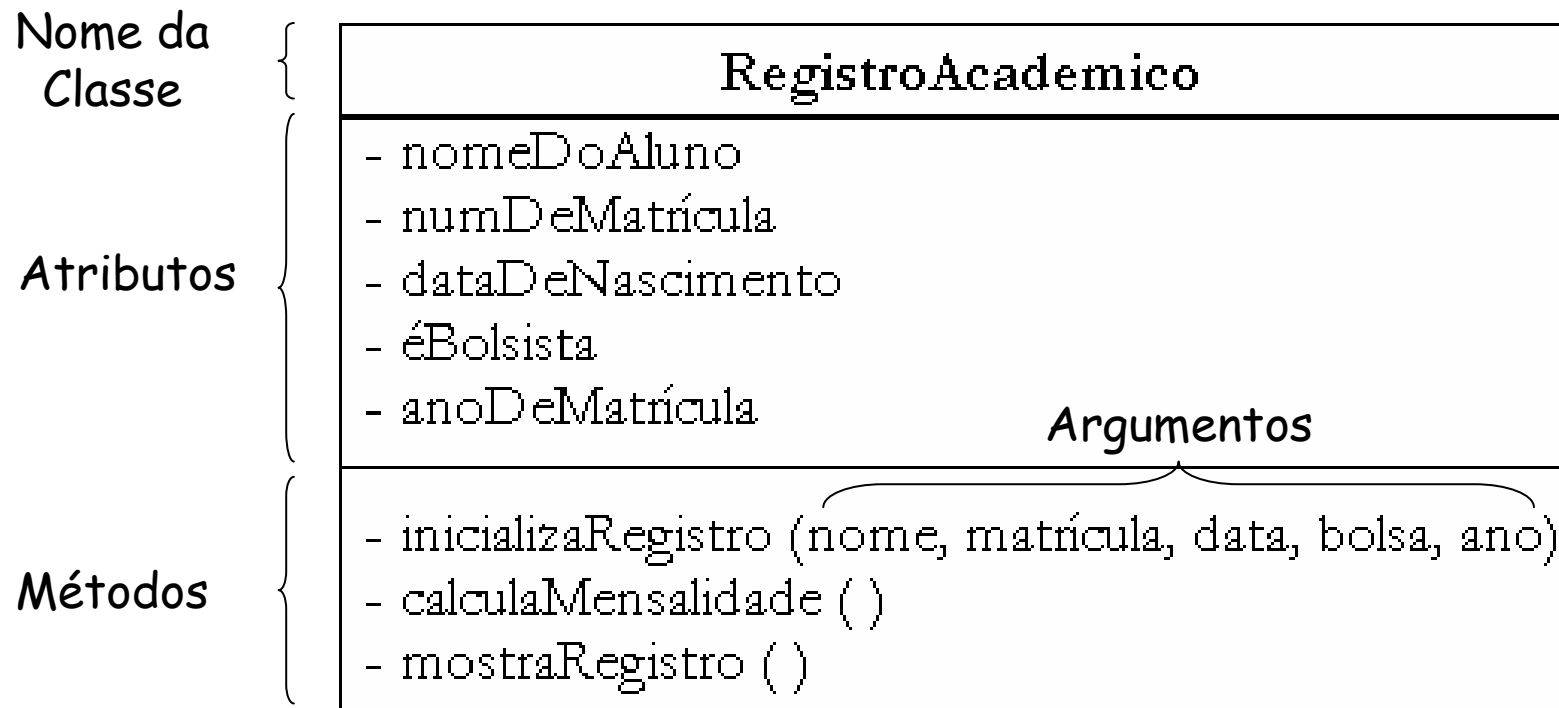
- Para que os objetos ou instâncias possam ser manipulados, é necessária a criação de referências a estes objetos, que são basicamente **variáveis do "tipo" da classe**;
- Os dados contidos em uma classe são conhecidos como **campos ou atributos** daquela classe:
 - Cada atributo deve ter um nome e um tipo (inteiro, ponto flutuante, etc);
- As operações contidas em uma classe são chamadas de **métodos** dessa classe.
- Métodos podem opcionalmente receber **argumentos** e/ou retornar valores ou instâncias de classes.

Exemplo

- Vejamos um exemplo de um modelo para fixação dos conceitos:
- Consideremos um **registro acadêmico de um aluno** e um modelo para representá-lo.
- Este modelo pode ser considerado uma ficha de matrícula ou identificação do aluno na escola contendo os dados e operações conforme mostra a figura do próximo slide

Exemplo

- O modelo RegistroAcademico, seus dados e atributos.



Exemplo

- O modelo RegistroAcademico, em pseudocódigo:

```
modelo RegistroAcademico
início do modelo
  // Dados do registro acadêmico
  dado nomeDoAluno, numDeMatrícula;
  dado dataDeNascimento, éBolsista, anoDeMatrícula;

  // Inicializa simultaneamente todos os dados do modelo, passando argumento
  operação inicializaRegistro (oNome, aMatrícula, aData, temBolsa, qualAno)
  início
    // Usa os argumentos para inicializar os valores no modelo
    nomeDoAluno = oNome;
    numDeMatrícula = aMatrícula;
    dataDeNascimento = aData;
    éBolsista = temBolsa;
    anoDeMatrícula = qualAno;
  fim

  operação calculaMensalidade () // calcula e retorna a mensalidade
  início
    mensalidade = 400;
    se (éBolsista) mensalidade = mensalidade / 2;
    retorna mensalidade;
  fim

  // Por falta de espaço, faltou mostrar o método mostraRegistro
fim do modelo
```

Encapsulamento

- Por fim, temos o **conceito de encapsulamento** que é “a capacidade de ocultar dados dentro de modelos, permitindo que somente operações especializadas ou dedicadas manipulem os dados ocultos”.
- A operação `calculaMensalidade` por exemplo, retorna um valor calculado dentro da operação, usando os dados encapsulados no modelo.
- O uso de uma operação para calcular um valor é uma **outra vantagem** do encapsulamento: deixa-se a cargo do modelo o cálculo de valores associados a ele.

Bibliografia

- SANTOS, R. *Introdução à programação orientada a objetos usando Java*, Editora Campus. 1ª ed. RJ: 2003.