Introdução POO

Profa Lucília Ribeiro

Seria maravilhoso se as máquinas sustentassem as pessoas.
Mas é preciso estar à altura dessa civilização.
E o que está acontecendo?
Que a tecnologia avançou muito mais do que nós.
Então a humanidade começa a parecer um macaco,
um gorila com uma metralhadora.
E o celular não tem culpa, ele é maravilhoso.
Um menino anda com uma universidade no bolso.
Mas como o menino não está à altura da tecnologia,
a usa pra qualquer lixo.

– José Alberto Cordano

O que é um programa de computador?

- Conjunto de comandos e regras que um programador deve conhecer para manipular os recursos de um computador.
- São escritos usando linguagens de programação.
- Programas processam dados.
- Paradigma de POO → dados e processamento um objeto só

O que são Modelos?

- Representações simplificadas de objetos, pessoas, itens, tarefas, processos, conceitos, ideias ... (do mundo real)
- Os modelos representam dados ou informações
- Contem operações ou procedimentos.
- Podem conter submodelos e ser parte de outros modelos

Modelo

Restaurante Caseiro Hipotético

Mesa 1	Mesa 2	Mesa 3
☐ kg refeição ☐ sobremesa ☐ refrig.2 L. ☐ refrig.600mL. ☐ refrig.lata ☐ cerveja	☐ kg refeição ☐ sobremesa ☐ refrig.2 L. ☐ refrig.600mL. ☐ refrig.lata ☐ cerveja	☐ kg refeição ☐ sobremesa ☐ refrig.2 L. ☐ refrig.600mL. ☐ refrig.lata ☐ cerveja
Mesa 4	**Mesa 5**	**Mesa 6**
☐ kg refeição ☐ sobremesa ☐ refrig.2 L. ☐ refrig.600mL. ☐ refrig.lata ☐ cerveja	☐ kg refeição ☐ sobremesa ☐ refrig.2 L. ☐ refrig.600mL. ☐ refrig.lata ☐ cerveja	☐ kg refeição ☐ sobremesa ☐ refrig.2 L. ☐ refrig.600mL. ☐ refrig.lata ☐ cerveja

Figura 1.1: O quadro-branco do Restaurante Caseiro Hipotético

Exemplos

- Modelo: `RestauranteCaseiro`
- Dados: `quantosGramas`, `quantidadeDoItem`
- Operações: `adicionaItem`, `apresentaConta`

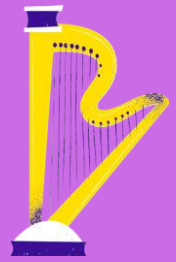
Informações específicas para diferentes contextos:

- Pessoa como empregado de empresa:
  - Dados: `nome, cargo, salario, horasExtras`
  - Operações: `calculaSalario, aumentaSalario`
- Pessoa como paciente
  - Dados: `nome, sexo, idade, altura, peso, historico`
  - Operações: `verificaObesidade, adicionaHistorico`
- Pessoa como contato comercial
  - Dados: `nome, telefone, empresa, cargo`
  - Operações: `mostraTelefone, trabalhaEmpresa`

O QUE É PROGRAMAÇÃO ORIENTADA A OBJETOS?

- Paradigma de programação onde se usam classes e objetos, criados a partir de modelos
- Limitações acontecem por causa do computador
- A modelagem dos dados e suas operações permite o processamento de dados de forma coesa, rápida e menos suscetível a erros

Encapsulamento



- Capacidade de ocultar dados dentro de modelos, permitindo que somente operações especializadas ou dedicadas manipulem tais dados

- Deve ser um dos principais objetivos do programador → programas com menos erros e mais clareza

Exemplos de Modelos

Uma lâmpada incandescente

Lampada
- estadoDaLâmpada
- acende() - apaga() - mostraEstado()

Figura 1.2: O modelo `Lampada`, seus dados e atributos

```
1  modelo Lampada // representa uma lâmpada em uso
2  início do modelo
3    dado estadoDaLâmpada; // indica se está ligada ou não
4
5    operação acende() // acende a lâmpada
6      início
7        estadoDaLâmpada = aceso;
8      fim
9
10   operação apaga() // apaga a lâmpada
11     início
12       estadoDaLâmpada = apagado;
13     fim
14
15   operação mostraEstado() // mostra o estado da lâmpada
16     início
17       se (estadoDaLâmpada == aceso)
18         imprime "A lâmpada está acesa";
19       senão
20         imprime "A lâmpada está apagada";
21     fim
22
23 fim do modelo
```

Uma Conta Bancária simplificada

ContaBancariaSimplificada
- nomeDoCorrentista - saldo - contaÉEspecial
- abreConta(nome,depósito,éEspecial) - abreContaSimples(nome) - deposita(valor) - retira(valor) - mostraDados()

Figura 1.3: O modelo `ContaBancariaSimplificada`, seus dados e atributos

Listagem 1.2: O modelo `ContaBancariaSimplificada`, em pseudo-código.

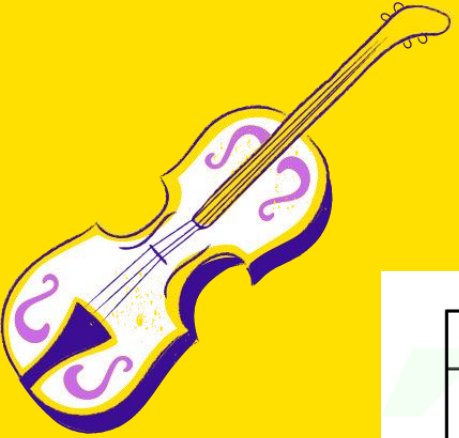
```
modelo ContaBancariaSimplificada
início do modelo
  dado nomeDoCorrentista,saldo,contaÉEspecial; // dados da conta

  // Inicializa simultaneamente todos os dados do modelo
  operação abreConta(nome,depósito,especial) // argumentos para esta operação
    início
      // Usa os argumentos passados para inicializar os dados do modelo
      nomeDoCorrentista = nome;
      saldo = depósito;
      contaÉEspecial = especial;
    fim

  // Inicializa simultaneamente todos os dados do modelo, usando o nome
  // passado como argumento e os outros valores com valores default
  operação abreContaSimples(nome) // argumento para esta operação
    início
      nomeDoCorrentista = nome;
      saldo = 0.00;
      contaÉEspecial = falso;
    fim

  // Deposita um valor na conta
  operação deposita(valor)
    início
      saldo = saldo + valor;
    fim
```

```
28
29    // Retira um valor da conta
30    operação retira(valor)
31      início
32        se (contaÉEspecial == falso) // A conta não é especial !
33          início
34            se (valor <= saldo) // se existe saldo suficiente...
35              saldo = saldo - valor; // faz a retirada.
36          fim
37        senão // A conta é especial, pode retirar à vontade !
38          saldo = saldo - valor;
39      fim
40
41    operação mostraDados() // mostra os dados da conta, imprimindo os seus valores
42      início
43        imprime "O nome do correntista é ";
44        imprime nomeDoCorrentista;
45        imprime "O saldo é ";
46        imprime saldo;
47        se (contaÉEspecial) imprime "A conta é especial.";
48        senão imprime "A conta é comum.";
49      fim
50
51  fim do modelo
```



Uma Data

Data
- dia - mês - ano
- inicializaData(d,m,a) - dataÉVálida(d,m,a) - mostraData()

Figura 1.4: O modelo `Data`, seus dados e atributos

Listagem 1.3: O modelo `Data`, em pseudo-código.

```
1  modelo Data
2  início do modelo
3    dado dia,mês,ano; // componentes da data
4
5    // Inicializa simultaneamente todos os dados do modelo
6    operação inicializaData(umDia,umMês,umAno) // argumentos para esta operação
7      início
8        // Somente muda os valores do dia, mês e ano se a data passada for válida
9        se dataÉVálida(umDia,umMês,umAno) // Repassa os argumentos para a operação
10         início
11           dia = umDia;
12           mês = umMês;
13           ano = umAno;
14         fim
15       // Se a data passada não for válida, considera os valores sendo zero
16       senão
17         início
18           dia = 0;
19           mês = 0;
20           ano = 0;
21         fim
22     fim
```

```
24    operação dataÉVálida(umDia,umMês,umAno) // argumentos para esta operação
25      início
26        // Se a data passada for válida, retorna verdadeiro
27        se ((dia >= 1) e (dia <= 31) e (mês >= 1) e (mês <= 12))
28          retorna verdadeiro;
29        // Senão, retorna falso
30        senão
31          retorna falso;
32      fim
33
34    operação mostraData() // mostra a data imprimindo valores de seus dados
35      início
36        imprime dia;
37        imprime "/";
38        imprime mês;
39        imprime "/";
40        imprime ano;
41      fim
42
43  fim do modelo
```

Um registro acadêmico de aluno

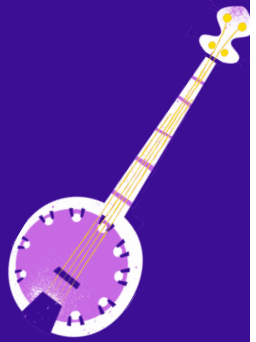


Figura 1.5: O modelo `RegistroAcademico`, seus dados e atributos

Listagem 1.4: O modelo `RegistroAcademico`, em pseudo-código.

```
1  modelo RegistroAcademico
2  início do modelo
3    // Dados do registro acadêmico
4    dado nomeDoAluno,númeroDeMatrícula;
5    dado dataDeNascimento,éBolsista,anoDeMatrícula;
6
7    // Inicializa simultaneamente todos os dados do modelo, passando argumentos
8    operação inicializaRegistro(oNome,aMatrícula,aData,temBolsa,qualAno)
9      início
10       // Usa os argumentos para inicializar os valores no modelo
11       nomeDoAluno = oNome;
12       númeroDeMatrícula = aMatrícula;
13       dataDeNascimento = aData;
14       éBolsista = temBolsa;
15       anoDeMatrícula = qualAno;
16     fim
17
18   operação calculaMensalidade() // calcula e retorna a mensalidade
19      início
20        mensalidade = 400;
21        se (éBolsista) mensalidade = mensalidade / 2;
22        retorna mensalidade;
23      fim
```

```
25    operação mostraRegistro() // mostra os dados do registro acadêmico
26      início
27        imprime "Nome do aluno:";
28        imprime nomeDoAluno;
29        imprime "Número de Matrícula:";
30        imprime númeroDeMatrícula;
31        imprime "Data de Nascimento:";
32        dataDeNascimento.mostraData(); // pede à data que se imprima !
33         se (éBolsista == verdadeiro) imprime "O aluno é bolsista.";
34         senão imprime "O aluno não é bolsista.";
35         imprime "Ano de Matrícula:";
36         imprime anoDeMatrícula;
37       fim
38
39  fim do modelo
```

Classes, Objetos, Instâncias e Referências

- **Classes** são estruturas das linguagens de programação orientadas a objetos para conter, para determinado **modelo**, os **dados** que devem ser representados e as **operações** que devem ser efetuadas com estes dados.

- **Classes** são somente **moldes** ou formas que representam os **modelos** abstratamente.
- Um **objeto** ou **instância** é uma **materialização** da classe, e assim pode ser usado para representar dados e executar operações.
- Para que os **objetos** possam ser **manipulados**, é necessária a criação de **referências** a estes objetos, que são basicamente variáveis do "tipo" da classe

OBRIGADA!

Perguntas?

Template: SlidesCarnival

Livro: Introdução à POO usando Java (Rafael Santos)

Bom Trabalho!