

## Material de Apoio 2

### Súmary

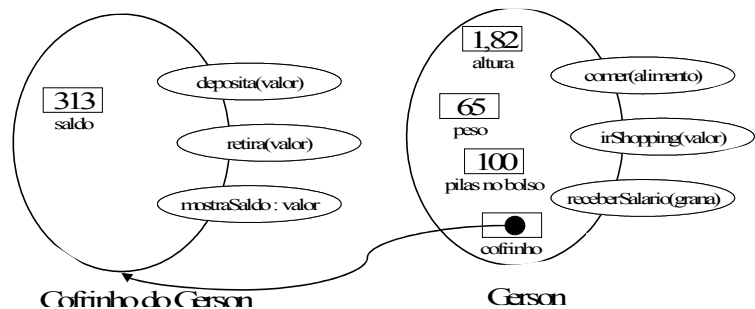
Neste material encontra-se introduzida a noção de objeto e classe no paradigma de programação orientado a objetos. Destaque também para os conceitos de referência e variável.

### Noção de objeto

O princípio da orientação a objetos está associado ao *encapsulamento* de comportamento: objetos podem ser considerados uma metáfora de comportamento intrínseco de entidades reais. Tal como em sistemas reais, em um programa OO não existem meios de abrir um objeto e olhar no seu interior, tão pouco alterar seu estado. Neste paradigma, a única forma de fazer evoluir um programa é permitir que objetos compartilhem dados entre si através de trocas explícitas de mensagens. Uma *mensagem* consiste em uma requisição, onde o objeto requisitante (sender) envia uma solicitação de serviço ao objeto requisitado (receiver). Este serviço, necessariamente, precisa pertencer à *interface de serviços* oferecida pelo objeto requisitado. Na nomenclatura OO, um serviço é especificado para um objeto como um *método*. A natureza de uma mensagem depende do tipo de serviço fornecido pelo método ao qual ela foi enviada. Um método pode ser de *acesso*, o qual permite a leitura de informações que estão localizadas no contexto do objeto requisitado, ou de *modificação*, que possui a capacidade de alterar alguma informação armazenada no contexto do objeto. A estas informações, representadas por dados armazenados no contexto do objeto, dá-se a denominação de *atributos* – objetos possuem atributos representando seu *estado interno*. O estado interno de um objeto pode conter tanto valores como referenciar outros objetos.

### Exercícios

1. Identifique, na figura ao lado, os seguintes itens:
  - a. Interface dos objetos
  - b. Métodos de acesso
  - c. Métodos modificadores
  - d. Estado interno



2. Utilizando a mesma figura do exercício anterior, proponha uma mensagem invocando um serviço no objeto Cofrinho do Gerson pelo objeto Gerson. Reapresente a figura, mostrando os estados alterados.
3. Identifique outros possíveis objetos interagindo com os objetos apresentados na figura do Exercício 1.
4. Tomando como base o objeto Cofrinho do Gerson defina a estrutura de um novo objeto: conta em banco do Gerson.

### Noção de referência

Na programação OO, a evolução da execução se dá através da interação entre objetos. Para tanto, os objetos trocam *mensagens* entre si. Como na vida real, um objeto que deseja mandar uma mensagem a um outro objeto necessita alguma referência sobre o objeto destinatário. Esta referência tem a função de endereçar o destinatário. No exemplo apresentado no Exercício 1, o atributo *cofrinho* do objeto Gerson é, na verdade, uma referência ao objeto Cofrinho do Gerson. Note que um objeto é apenas referenciado, ou seja, ele não se encontra contido no estado interno do objeto que o referencia. Desta forma, é possível que um mesmo objeto seja referenciado por dois ou mais objetos ao mesmo tempo – basta para isto que existam duas, ou mais, referências a este objeto. Note, no entanto, que, caso um novo objeto seja atribuído a uma referência já existente, a referência passará a referenciar-se a este novo objeto.

5. Faz sentido, em um programa orientado a objetos, a existência de um objeto que não possua nenhuma referência a ele?
6. Dois objetos da classe Cofrinho podem possuir o mesmo valor em saldo? Justifique.
7. Imagine a situação em que objetos representando os professores Vera e Gerson possuam, cada um, um cofrinho em seu estado interno. O objeto Gerson poderá obter informações sobre o cofrinho do objeto Vera?
8. Considere uma situação onde um cofrinho pode ser de uso compartilhado (situação semelhante a uma conta conjunta, onde dois “professores” podem compartilhar o mesmo cofrinho). Como isto poderia ser representado durante a execução de um programa OO? (Utilize como base a figura do Exercício 1).

### Noção de variável

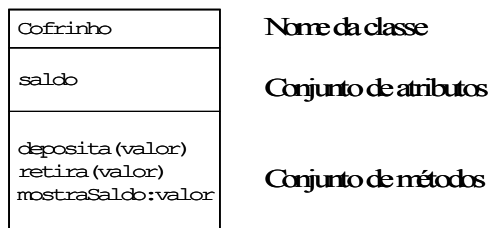
Uma variável consiste em um nome lógico atribuído a uma posição de memória. Nesta posição de memória é possível armazenar informações. Como seu nome já indica, o valor de uma variável “vária” durante a execução do programa. Uma variável difere de uma referência a um objeto uma vez que ela “contém” um valor, enquanto que a referência “referencia” um objeto. Observe que uma variável pode conter apenas um valor a cada vez.

9. Identifique, no exemplo dado na figura do Exercício 1, onde encontram-se as variáveis?
10. Estas variáveis também compõem o estado interno de algum objeto?
11. Um determinado valor associado a uma determinada variável pode ser compartilhado entre outras variáveis?
12. Faça um algoritmo (seqüência de instruções) que troque o conteúdo de duas variáveis: A e B, conforme o esquema abaixo.  
Antes: A = 313; B = 613;  
Depois: A = 613; B = 313;

### Noção de classe

Ainda considerando sistemas reais, é fácil notar que muitos objetos possuem características estruturais semelhantes, embora todo objeto seja único. Um bom exemplo disto são os objetos que recebem a denominação genérica de *professor*. Todos os professores possuem uma “mesma estrutura”, ou seja, todos os objetos professor implementam os mesmos métodos (comer, irShopping, receberSalario, darAula, atenderAluno etc) e mantêm informações sobre os mesmos atributos (peso, altura, pilas no bolso, cofrinho, número funcional, nome etc). O fato de que cada objeto mantém seus próprios atributos de forma encapsulada confere uma identidade única a cada objeto – e também permite que dois objetos possam, eventualmente, possuir valores iguais para seus atributos. Estes grupos de objetos com estruturas semelhantes são definidos em termos de *classes*. Classes consistem na unidade básica da construção de um programa OO. Uma classe define o conjunto de atributos mantidos por um conjunto de objetos e o comportamento que estes objetos devem respeitar. Nesta nomenclatura, um objeto é dito *instância* de uma classe, ou seja, um objeto materializa, em um programa em execução, uma unidade de execução que reflete a definição apresentada em uma classe.

13. A figura abaixo representa a classe Cofrinho. Considerando a simbologia utilizada nesta representação, defina a classe Professor e a classe ContaEmBanco.



14. Descreva, textualmente, os serviços prestados pelos métodos da interface de cada classe definida.
15. Identifique no programa Java abaixo onde se encontram as classes e os objetos.

```
public class teste {  
    public static void main( String args [] ) {  
        String nome;  
        Entrada teclado;  
        Saida tela;  
  
        nome = new String();  
        tela = new Saida();  
        teclado = new Entrada();  
        tela.print("Digite seu nome: ");  
        nome = teclado.gets();  
        tela.print("Ola " + nome + ", benvindo a Laboratorio 1!!!");  
    }  
}
```

16. Comente, linha a linha, o programa apresentado na figura do Exercício 15.
17. O que é um arquivo .jar utilizado em Java?

### Noção de tipo de dado

Um tipo de dado define o espaço de memória requerido para armazenar dados de uma variável. Em Java, os tipos primitivos são utilizados para representar valores [booleanos](#) (verdadeiro ou falso), [caracteres](#), valores [numéricos inteiros](#) e valores [numéricos em ponto flutuante](#). Este tipos são: bool, char, int, double.

18. Qual tipo de dado é adequado ao atributo altura na classe Professor?
19. E para um eventual atributo sexo?
20. Quais são os valores que um atributo do tipo int pode assumir?