Conceitos de Programação Orientada a Objetos

Prof. Edwar Saliba Júnior
Fevereiro de 2011

Conceitos Fundamentais

- Classe;
- Ocultamento de informação;
- Encapsulamento;
- Atributo;
- Método;
- Mensagem;
- Objeto.

É bom saber!

- Palavra-chave no paradigma de orientação a objetos:

ORGANIZAÇÃO

do código-fonte!

Carro X OO

- Suponha que você queira guiar um carro e em determinados momentos fazê-lo andar mais rápido pisando no acelerador;
- O que deve acontecer antes que você possa fazer isto?
  - Alguém tem que projetar e construir o carro!

Classe

- Em geral, os carros 'nascem' em um desenho de engenharia semelhante às plantas utilizadas nos projetos de casas;
- Nesta planta, procura-se detalhar o máximo possível, de forma genérica, todos os atributos (características) e funcionalidades do que está sendo projetado;
- Neste desenho de engenharia (o do carro) está o desenho do pedal do acelerador, do freio, da direção e etc.

Ocultamento de Informações

- O pedal do acelerador, o pedal do freio e a direção, **ocultam** de seus usuários os complexos mecanismos que os fazem funcionar;
- Este processo de ocultamento permite que pessoas com pouco ou nenhum conhecimento de como os motores e as engrenagens funcionam, dirijam um carro facilmente.

Encapsulamento

- Junto ao mecanismo do acelerador, neste mesmo carro, estão encapsulados outros mecanismos fundamentais para o seu funcionamento, tais como:
  - o câmbio de marchas para aumentar ou diminuir, juntamente com o acelerador, a velocidade do carro;
  - o freio para diminuir a velocidade ou mesmo parar o carro;
  - o velocímetro para medir a velocidade que o carro está fazendo;
  - o marcador de combustível para medir o que tem de gasolina no tanque e
  - o volante para mudar o carro de direção e etc.

Atributos

- Ainda no projeto diversos atributos do carro são definidos pelos engenheiros. Tais como:
  - quantidade de portas;
  - tamanho e largura dos pneus;
  - motor;
  - combustível e
  - etc.

Método

- Para realizar uma tarefa em um programa é necessário um:
  - **Método**: Os métodos descrevem mecanismos que realizam tarefas;
  - O método oculta de seus usuários as tarefas complexas que ele realiza. Assim como o pedal do acelerador, o pedal de freio, o câmbio de marchas, o velocímetro e o marcador de combustível; ocultam do motorista os complexos mecanismos que os fazem funcionar.

Mensagem

- Nem todos os mecanismos (**métodos**) citados anteriormente funcionam sozinhos. Ou seja, para o carro começar a andar ou mesmo para andar mais rápido, uma **mensagem** deverá ser enviada ao motor. Esta mensagem é enviada pisando-se no acelerador;
- Para parar o carro envia-se uma mensagem pisando no freio;
- Para mudar de direção envia-se uma mensagem virando o volante;
- E assim por diante.

Objeto

- Depois que todos estes mecanismos foram projetados e no projeto estão todos perfeitamente calculados e elaborados, então, podemos criar e usufruir do nosso carro ou melhor, do nosso **OBJETO**.

Programação Orientada a Objetos

- Análogo a todos os passos descritos nos *slides* anteriores, se dá a criação de um programa orientado a objetos;

- Vejamos nos *slides* a seguir:

TAD

- TAD (Tipo Abstrato de Dados) é a representação **encapsulada** de um tipo definido pelos seus atributos e suas operações;
- É uma estrutura de programação, na qual uma determinada estrutura de dados, é conhecida somente via as operações realizadas sobre os seus elementos de dados, sem que se identifique como a estrutura é codificada". (Staa (2000) *apud* Ferreira(2009))
- A programação orientada a objetos, é o resultado do uso da abstração de dados no desenvolvimento de *softwares*.

Classe

- Em POO é uma unidade de programação criada para abrigar os **atributos** e os **métodos** de um objeto;
- “Uma classe é um conceito de OO que encapsula as abstrações de dados e procedimentos necessários para descrever o conteúdo e o comportamento de alguma entidade do mundo real”. (Pressman (2002) *apud* Ferreira (2009));
- Uma classe é o “projeto de engenharia”, do que mais tarde será chamado de objeto, com seus atributos e métodos;
- Uma classe corresponde a um **TAD** (Tipo Abstrato de Dados).

Atributo

- Também denominado como: campo, propriedade ou variável de instância;
- Representam as características que os objetos da classe possuem;
- Um objeto possui atributos que são portados consigo, quando ele é utilizado em um programa;
- Ex.: Um objeto conta bancária tem um atributo saldo que representa a quantidade de dinheiro na conta. Cada objeto conta bancária sabe o saldo da conta que representa, mas não sabe o saldo de outras contas.

Método

- Também denominado “membro de função” ou “operação”;
- Os métodos representam o comportamento que o os objetos da classe possuem;
- Análogo às “funções” e “procedimentos” da Programação Estruturada, porém aqui, estes estão encapsulados dentro de um determinado contexto, ou seja, a classe;
- Em POO um método pertence a uma determinada **classe**.

Método Construtor

- O método construtor é um método especial que serve para inicializar um objeto e seus atributos;
- O método construtor pode ou não ser declarado explicitamente. Mesmo que não seja, toda classe possui um método construtor implícito que é disparado na criação do objeto;
- O método construtor possui o mesmo nome da classe, porém, não possui tipo de retorno;
- Podem existir diversos métodos construtores em uma mesma classe, desde que a assinatura de cada qual seja diferente uma da outra.

Objeto

- Uma classe descreve uma categoria genérica, ou seja, um tipo genérico de dados;
- Um objeto é uma instância de uma classe, ou seja, é uma estrutura de dados que representa um membro específico da categoria;
- Exemplo:
  - classe: Aluno
  - objetos da classe Aluno: João, Paulo, Sílvia, Marina, Jurema e Marcos
- Objeto é uma estrutura computacional que representa um objeto do mundo real;
- Um objeto e suas funcionalidades, só podem ser usados depois de devidamente instanciados. Salvo os casos específicos.

Partes Básicas de Um Programa

```
class NomeDaClasse{
    public static void main(String[] args){
        …
        comandos
        …
    }
}
```

Exemplo Criação de Programa

- Se você abrir o aplicativo “Bloco de Notas” (no Windows) e digitar o programa:

```
class Hello{
   public static void main(String[] args){
      System.out.println(“Hello world!”);
   }
}
```

- Ele funcionará perfeitamente. Assim que você compilá-lo e executá-lo.

Comandos: Compilação e Execução

- Vamos supor que o arquivo anterior tenha sido salvo com o nome: “hello.java”
- Compilação:

```
javac hello.java
```

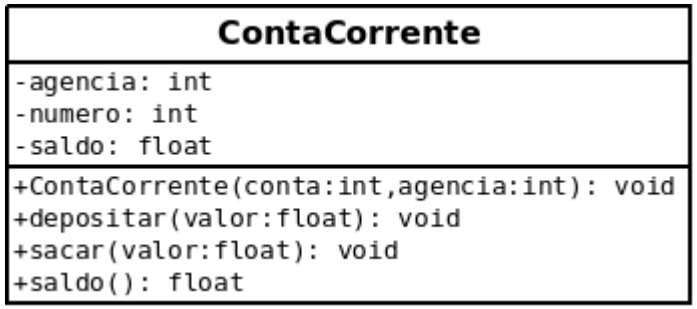
  - será criado o arquivo “hello.class”;
- Execução:

```
java hello.class
```

  - será executado o programa.

Outro Exemplo

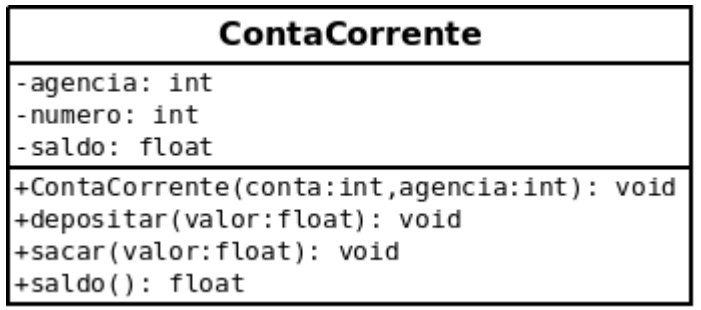
- No contexto de automação bancária, identifica-se, dentre outras, a seguinte classe:
  - Conta Corrente
- Estrutura da classe Conta Corrente (representação UML):



Código da Classe Conta Corrente

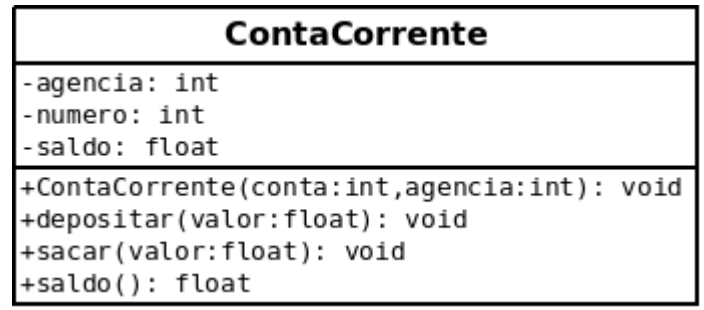
```
public class ContaCorrente {
    private int agencia;
    private int numero;
    private float saldo;

    public ContaCorrente(int conta, int agencia) {
        numero = conta;
        agencia = agencia;
        saldo = 0;
    }
```



Código da Classe Conta Corrente

```
    public void sacar(float valor){
        if((saldo - valor) >= 0)
            saldo = saldo – valor;
        else
            System.out.println(“Saldo insuficiente!”);
    }
    public void depositar(float valor){
        if(valor > 0)
            saldo = saldo + valor;
    }
    public float saldo(){
        return(saldo);
    }
}
```



Observações

- No exemplo anterior, o método ContaCorrente (de nome idêntico ao da classe) é denominado **construtor** da classe;
- O construtor é executado toda vez que um objeto desta classe é criado. Por ser um método especial, ele não especifica um tipo de retorno, pois não há;
- Um método construtor é utilizado para determinar o estado inicial do objeto;
- Em Java, objetos são criados utilizando-se o comando ***new***.

Criação de um Objeto

Exemplo:

Tipo da Variável (classe)

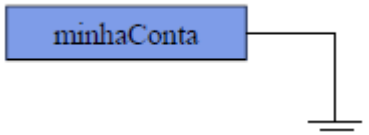
Variável

Método Construtor

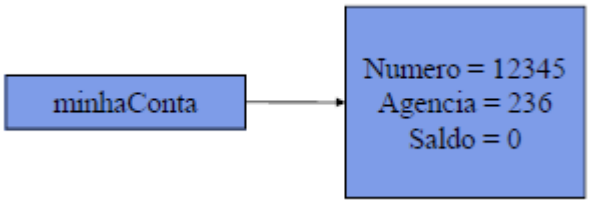
```
ContaCorrente minhaConta;
minhaConta = new ContaCorrente(12345, 236);
```

Comando para Criação de Objetos

- Na primeira linha do exemplo, é criada uma área na memória, que é uma referência para um objeto da classe **ContaCorrente**;



- Na segunda linha é instanciado (criado), um objeto da classe **ContaCorrente,** e este é atribuído a variável **minhaConta**;



Comunicação entre Objetos

- Programas orientados a objetos são constituídos por objetos que trocam mensagens entre si;
- O envio de uma **mensagem** a um objeto corresponde a invocar (chamar) um **método** de tal objeto;
- Em:

```
minhaConta.depositar(54.00);
```

- O método **depositar** do objeto **minhaConta** é invocado. Em outras palavras, é enviada uma **mensagem** para o objeto **minhaConta** para que este realize a operação depositar.

Exercícios

- Para os sistemas a seguir, identifique 3 classes, seus respectivos métodos e atributos:
  - Sistema de Video Locadora;
  - Sistema de Gestão de Biblioteca;
  - Agenda de Compromissos.

Bibliografia

- DEITEL, H. M.; DEITEL, P. J. **Java Como Programar**; tradução Edson Furmankiewicz; revisão técnica Fábio Lucchini. 6. ed., São Paulo: Pearson, 2005.
- FERREIRA, Kecia Aline Marques. *Slides* da disciplina de Programação de Computadores II. CEFET-MG, 2009.
- HUBBARD, John R. **Teoria e Problemas da Programação com Java**. 2. ed. Porto Alegre: Bookman, 2006.
- MIZRAHI, Victorine Viviane. **Treinamento em Linguagem C++ - Módulo 2**. 2. ed., São Paulo: Pearson, 2006.