

Hortolândia, 04 de setembro de 2013.

Aulas 21 e 22 – Programação Orientada a Objetos

Introdução

A presente aula reforça os conceitos de orientação a objetos, apresentados na aula anterior - classes, objetos, métodos e atributos, bem como se aprofunda na escrita e utilização dos **métodos das classes**.

Em particular, será apresentado o método **construtor**, o qual é utilizado externamente à classe para se instanciá-la.

1. Métodos de uma classe

Um **método** é um bloco de construção de programa, no interior de uma classe, que tem a função de realizar tarefas - por meio de suas instruções - e retornar um valor ao final da execução.

O **formato** genérico de uma definição de método é o seguinte:

```
tipo_valor_retorno nome_método (lista_parâmetros) {  
    declarações e instruções  
}
```

O **nome do método** é qualquer identificador válido, de acordo com a sintaxe da linguagem Java.

O tipo do valor de retorno é o **tipo de dados** do **resultado** retornado do método para o chamador.

A Figura 1 representa dois métodos de uma classe, com os respectivos elementos.

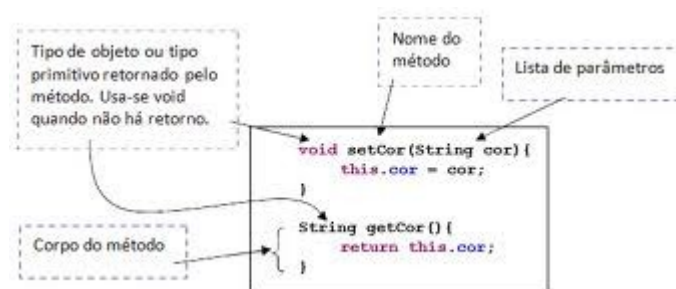


Figura 1: representação de dois métodos de uma classe, denominados `setCor()` e `getCor()`.

O tipo do valor de retorno **void** indica que o método não retorna nenhum valor. Os métodos podem retornar no máximo um valor (DEITEL et al, 2001).

A **lista de parâmetros** é uma lista separada por vírgulas na qual o método declara o nome e o tipo de cada parâmetro. Deve haver um argumento na chamada de método para cada parâmetro na definição do método.

Cada argumento também deve ser compatível com o tipo do parâmetro correspondente na definição do método. Por exemplo, um parâmetro do tipo **double** pode receber valores como 7.53, 8.39, mas não um valor tal como "Fulano de Tal", pois este último é um valor do tipo **String**.

Se um método não recebe nenhum valor, a lista de parâmetros estará vazia.

Após a primeira linha da definição do método – também conhecida como **cabeçalho do método** – existe o **corpo do método**, formado por declarações e instruções. O corpo do método também é conhecido como bloco.

As variáveis podem ser declaradas em qualquer bloco e os blocos podem ser aninhados. Um método não pode ser definido no interior de outro método.

A palavra-chave **return** é utilizada ao final do método para retornar um valor (do mesmo tipo de dados indicado no cabeçalho do método) para a instrução que invocou o método.

Por exemplo, suponha que uma determinada classe possua os métodos **getCor()** e **setCor()**, responsáveis pela leitura e escrita dos valores de uma variável de instância, do tipo String, denominada **cor**.

O código-fonte desses dois métodos está descrito na Figura 1, bem como na Listagem 1 a seguir.

```
public class Exemplo1 {  
    public String cor;  
    public static void main(String[] args){  
        Exemplo1 ex1 = new Exemplo1();  
    }  
    public Exemplo1(){  
        // Comandos e declarações do Construtor.  
        this.cor = "";  
    }  
    public void setCor(String cor){  
        this.cor = cor;  
    }  
    public String getCor(){  
        return cor;  
    }  
}
```

Listagem 1: classe de exemplo contendo os métodos setCor() e getCor().

2. O Método Construtor

Existe um método especial em cada classe, denominado **construtor**.

O **método construtor** possui as seguintes características:

- Ele possui o mesmo nome da classe;
- Serve para inicializar os valores dos atributos da classe;

A linguagem Java chama o construtor de uma classe quando um objeto desta classe é instanciado, por meio de uma instrução **new**.

Por exemplo, para instanciar um objeto da **ClasseB**, no interior da **ClasseA**, pode-se ter o seguinte trecho de código:

```
public class ClasseA {  
    public static void main(String[] args){  
        ClasseB cb = new ClasseB( );  
    }  
}  
  
public class ClasseB {  
    public ClasseB(){  
        // Corpo do método construtor da ClasseB.  
    }  
}
```

Obs.: No trecho de código acima, estão representadas duas classes, ClasseA e ClasseB. As mesmas devem ser escritas cada qual em um arquivo de mesmo nome da respectiva classe.

3. Utilização dos métodos de uma classe

A fim de se utilizar um método de uma classe, faz-se necessário que o objeto desta referida classe esteja instanciado.

Além disso, utiliza-se o caractere “.” (ponto) para separar a variável de objeto da classe do método que estiver sendo chamado.

Por exemplo, para o código da classe Exemplo1, mostrada na Listagem1, poderia ter-se uma outra classe que invocasse os métodos getCor() e setCor() da classe Exemplo1.

Considere o código-fonte da classe Exemplo2 da Listagem 2 a seguir.

```
public class Exemplo2{  
    public static void main(String[] args){  
        Exemplo1 ex1 = new Exemplo1();  
        ex1.setCor("Branca");  
        System.out.println("Cor: " + ex1.getCor());  
    }  
}
```

Listagem 2: Classe Exemplo2.

5. Conclusões

A presente aula ilustrou os conceitos de **método**, **chamadas de método** e o **método construtor**.

Tais conceitos são essenciais para a **programação orientada a objetos**.

6. Referências Bibliográficas

BORATTI, Isaias Camilo. **Programação Orientada a Objetos usando Delphi**. Quarta Edição. Editora Visual Books. Florianópolis, 2007.

DEITEL, H.M., DEITEL, P.J. **Java – Como programar**. Terceira edição. Porto Alegre: Bookman Editora, 2001.

SANTOS, Rafael. **Introdução à Programação orientada a objetos usando Java**. 8ª Reimpressão. Rio de Janeiro: Campus - Elsevier, 2003.