

PROGRAMAÇÃO I - CCT0827

Semana Aula: 3

Unidade 2 - Conceitos de orientação a objetos

Tema

Classes e Objetos

Palavras-chave

Classes, Atributos, Métodos, Objetos, Métodos Setters, Métodos Getters.

Objetivos

O aluno deverá ser capaz de:

- Compreender os conceitos de classes, objetos, atributos e métodos.
- Compreender os conceitos de métodos de acessos.
- Desenvolver aplicações orientadas a objetos.

Estrutura de Conteúdo

Orientação a Objetos

A programação orientada a objetos tem como principal conceito representar em um sistema computacional um objeto da vida real. Esta representação deve descrever o objeto quanto às suas características e a ações que o objeto poderá realizar dentro do sistema. Não devemos nos preocupar com todas as características presentes no objeto, mas com aquelas que serão necessárias ao sistema (requisitos).

Classes

As classes Java são responsáveis pelo conjunto de códigos para a criação de objetos e aplicações.

Uma classe Java deve descrever as características e ações que o objeto possui ao ser representado em um sistema computacional, levando em consideração as características e ações realmente importantes para o sistema.

Atributos

Atributo é conceitualmente um descritor do objeto. Deve representar uma característica do objeto. O conjunto de atributos do objeto devem representar todas as características importantes do objeto para o sistema.

Exemplos:

```
String matricula; // atributo para armazenamento da matrícula
```

```
String nome; // atributo para armazenamento do nome
```

```
double cr; // atributo para armazenamento do cr
```

Métodos:

Método é uma ação, um conjunto de instruções a serem executadas por um objeto para realizar uma determinada tarefa.

Exemplos:

```
public int soma(int n1, int n2){
```

```
    int soma;
```

```
    soma = n1 + n2;
```

```
    return soma;
```

```
}
```

```
public void imprimeAumento(double salario, int percentual){
```

```
    double aumento;
```

```
    aumento = salario + salario * percentual / 100.0;
```

```
    System.out.println("O salário com aumento é: " + aumento);
```

```
}
```

Objetos

A classe modela o objeto de acordo com as necessidades do sistema para a sua descrição e suas ações. A partir de uma mesma classe, vários objetos diferentes, mas com características semelhantes podem ser criados em um mesmo sistema ou em diferentes sistemas.

Os objetos só existem durante a execução do sistema, pois estes só existirão como referência na memória do computador neste momento. Dizemos também que os objetos só existem “em tempo de execução”, uma vez que o sistema ao ser encerrado terá toda a

sua memória apagada e conseqüentemente todas as suas variáveis e objetos não existirão mais.

Aplicações Java

Aplicações em Java são classes especiais que possuem um método main(). O método main é responsável por criar os objetos e realizar a combinação de diferentes classes para atender as necessidades de um sistema. Em cada sistema temos apenas uma aplicação, que será a responsável pela lógica de criação e uso das classes.

A comunicação entre os objetos ocorre através de trocas de mensagens, que são expressas através do uso de métodos.

Métodos Setters e Getters

1. Métodos Setters

São métodos especiais que recebem o valor do atributo e por serem métodos, podem analisar se são válidos, sendo responsáveis pela atribuição. Quando o atributo é protegido (privado) é necessário um método para realizar a atribuição.

Características dos métodos Setters:

- São sempre do tipo void, pois métodos Setters não devem retornar nada;
- Devem ser públicos para que a aplicação tenha acesso ao método;
- Devem começar pela palavra set e o nome do atributo, como tem mais de uma palavra, cada nova palavra no nome deve começar por letra maiúscula;
- Possui sempre um parâmetro do mesmo tipo do atributo que receberá o valor, pois ambos (parâmetro e atributo) devem ser do mesmo tipo.

2. Métodos Getters

São métodos especiais que retornam o valor armazenado no atributo, evitando acesso direto ao atributo pela aplicação. Assim como visto no método Setter, a proteção do atributo (private) fará com que a aplicação não tenha acesso direto ao atributo fazendo com que seja necessário um método público para recuperar o valor atribuído ao mesmo.

Características dos métodos Getters:

- São sempre do mesmo tipo do atributo que será retornado, nunca do tipo void;
- Devem ser públicos para que a aplicação tenha acesso ao método;
- Devem começar pela palavra get e o nome do atributo, como tem mais de uma palavra, cada nova palavra no nome deve começar por letra maiúscula;

- Não possui parâmetro, esses métodos nunca receberão parâmetros uma vez que não farão atribuições ou ações com parâmetros, realizando apenas o retorno do valor armazenado no atributo.

Estratégias de Aprendizagem

Implementar programas em Java, explorando os conceitos de orientação a objetos: classes, atributos, métodos, objetos, métodos setters e métodos getters.

Exemplo:

Arquivo do projeto: **Aluno.java**

```
public class Aluno {  
  
    // Atributos devem ser identificados começando por letras minúsculas  
  
    String matricula, nome;  
  
    double cr;  
  
    // Métodos devem ser identificados começando por letras minúsculas  
  
    public void setMatricula(String m){  
  
        if(!m.isEmpty()) { // se o parâmetro m NÃO (!) estiver vazio  
  
            matricula = m; // será feita a atribuição  
  
        }  
  
    }  
  
    public String getMatricula(){  
  
        return matricula; // retorna a matrícula  
  
    }  
  
    public void setNome(String n){  
  
        if(!n.isEmpty()) { // se o parâmetro n NÃO (!) estiver vazio  
  
            nome = n; // será feita a atribuição  
  
        }  
  
    }  
  
}
```

```

}

public String getNome(){

return nome; // retorna o nome

}

public void setCr(double c){

if (c >=0 && c<=10){ // se o parâmetro c for válido

cr = c; // o valor de c será atribuído

}

}

public double getCr(){

return cr; // retorna o CR

}

public void imprimir( ){

// os métodos Getters foram usados aqui

System.out.println("Matrícula: " + getMatricula());

System.out.println("Nome : " + getNome());

System.out.println("CR : " + getCr());

}

}

```

Arquivo do projeto: **AppAluno.java**

//Aplicação para uso da Classe Aluno

```

public class AppAluno { // declaração e início da classe

public static void main (String[] args){// método inicial da App

Aluno aluno1 = new Aluno(); // Criação ou instanciação do objeto aluno1

Aluno aluno2 = new Aluno(); // Criação ou instanciação do objeto aluno2

```

```
Aluno aluno3 = new Aluno(); // Criação ou instanciación do objeto aluno3
```

```
//definindo valores para os atributos do aluno1
```

```
aluno1.setMatricula( "1001" );
```

```
aluno1.setNome( "André" );
```

```
aluno1.setCr( 6.7 );
```

```
//definindo valores para os atributos do aluno2
```

```
aluno2.setMatricula( "1002" );
```

```
aluno2.setNome( "Maria" );
```

```
aluno2.setCr( 7.5 );
```

```
//definindo valores para os atributos do aluno3
```

```
aluno3.setMatricula( "" ); // valor vazio, não será atribuído
```

```
aluno3.setNome( "" ); // valor vazio, não será atribuído
```

```
aluno3.setCr( 12 ); // valor do CR inválido, não será atribuído
```

```
//exibindo os valores dos atributos de cada aluno:
```

```
aluno1.imprimir();
```

```
aluno2.imprimir();
```

```
aluno3.imprimir();
```

```
}
```

```
}
```

Indicação de Leitura Específica

Orientação a Objetos:

FURGERI, Sérgio. Java 8 – ensino didático: desenvolvimento e implementação de aplicações. São Paulo: Érica, 2015. [Páginas: 103 – 108]

Métodos Setters e Getters:

Get e Set - Métodos Acessores em Java - <https://www.devmedia.com.br/get-e-set-metodos-acessores-em-java/29241>

Aplicação: articulação teoria e prática

1) Imagine que desejamos implementar um editor gráfico que irá manipular figuras geométricas:

Retângulo

Triângulo

Círculo

Quais as classes candidatas para esse projeto ?

2) Sabendo que todo livro possui autor, editora e preço, crie uma classe pública com construtor padrão e métodos modificadores e de acesso. Depois, crie dois livros com dados obtidos via console.

3) Utilizando como base o programa do exercício 2, altere os dados do 2º. livro da seguinte forma: autor para Harlan Coben, editora Arqueiro e preço R\$ 30,00.

4) Utilizando como base o programa do exercício 2, compare os preços dos dois livros, imprimindo o nome da editora do livro de menor preço.

Considerações Adicionais