



Análise Orientada a objetos

Aula 4

Prof. Me. Juliana Costa-Silva



Na aula de hoje...

1. [static](#)
2. [Atividade](#)
3. [Atividade](#)
4. [Leitura recomendada](#)



Java

Static

- Atributos e métodos estáticos são compartilhados por todos os objetos criados a partir daquela classe.



Static

- Atributos e métodos estáticos são compartilhados por todos os objetos criados a partir daquela classe.
- Existe apenas uma única cópia do método e/ ou atributo que esta associada com a classe e não com alguma instância criada da classe (objetos).

Atributos e métodos
estáticos

pertencem à classe

Atributos e métodos
não-estáticos

pertecem ao objeto



Restrições:

- um método estático, **NÃO** pode acessar variáveis de instância;



Restrições:

- um método estático, **NÃO** pode acessar variáveis de instância;
- um método estático, **NÃO** pode se referenciar a métodos de instância.



static

```
1 public class Estatico {  
2     static int valor = 5;  
3     public int getValor(){  
4         return valor;  
5     }  
6 }
```



static

```
1 public class TestaEstatico {
2     public static void main(String[] args) {
3         Estatico e1 = new Estatico();
4         Estatico e2 = new Estatico();
5         Estatico e3 = new Estatico();
6         System.out.println("Valor = " + e1.getValor()
7             );
8         e1.valor++;
9         System.out.println("Valor = " + e1.getValor()
10            );
11        e2.valor++;
12        System.out.println("Valor = "+ e1.getValor())
13            ;
14    }
15 }
```




Atividade

1. Modele um funcionário. Ele deve ter o nome do funcionário, o departamento onde trabalha, seu salário (double), a data de entrada no banco (String), seu RG (String) e um valor booleano que indique se o funcionário está na empresa no momento ou se já foi embora. Você deve criar alguns métodos de acordo com sua necessidade.
2. Além deles, crie um método bonifica que aumenta o salário do funcionário de acordo com o parâmetro passado como argumento. Crie, também, um método demite , que não recebe parâmetro algum, só modifica o valor booleano indicando que o funcionário não trabalha mais aqui.



Atividade

1. Transforme o modelo em uma classe Java. Teste-a, usando uma outra classe que tenha o main. Você deve criar a classe do funcionário chamada `Funcionario`, e a classe de teste você pode nomear como quiser. A de teste deve possuir o método `main`. Um esboço das classes:

```
class Funcionario {  
  
    double salario;  
    // seus outros atributos e métodos  
  
    void bonifica(double aumento) {  
        // o que fazer aqui dentro?  
    }  
  
    void demite() {  
        // o que fazer aqui dentro?  
    }  
}
```

```
1 class TestaFuncionario {  
2  
3     public static void main(String[] args) {  
4         Funcionario f1 = new Funcionario();  
5  
6         f1.nome = "Fiodor";  
7         f1.salario = 100;  
8         f1.bonifica(50);  
9  
10        System.out.println("salario atual:" + f1.salario);  
11    }  
12 }  
13 }
```



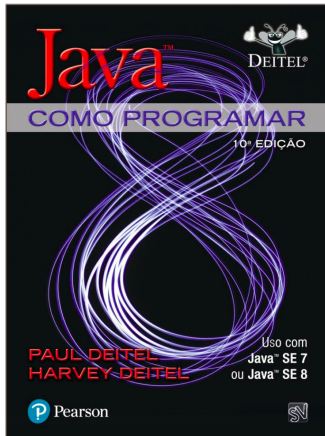
Atividade de Aula

1. Implemente o método depositar, na classe conta.
2. Lembre-se que ele deve atualizar o saldo do cliente;
3. Teste o método criado.
4. Envie o código desenvolvido em aula, como atividade de aula.



Leitura complementar

Para mais informações sobre JAVA, leia:



Capítulo 3 a 7:
[Deitel, 2016]



Referências



Deitel, Paul J.; Deitel, H. M. (2016).

Java 8: Como programar. 10ª Edição.

Pearson.