

Programação Orientada a Objetos I

Aula 02 - Construtores, getters e setters

Me. Edimo Sousa Silva



Construtores

- São os métodos usados para instanciar os objetos das respectivas classes;
- Definem quais ações devem ser tomadas ao criar uma instância da classe;
- Em Java o construtor é um método que tem o mesmo nome da classe e sem indicação de tipo de retorno;
- O construtor só é chamado no momento de criação de um novo objeto;



Construtores - parte 2

- Toda classe tem pelo menos um construtor;
- Se nenhum construtor for explicitamente definido, um construtor padrão, que não recebe argumentos, é incluído na classe pelo compilador Java;
- Se o programador da classe criar pelo menos um método construtor, o construtor padrão não será criado automaticamente;
- O construtor pode ser private;



Construtores - parte 3

- O retorno do operador new é uma referência para o objeto recém-criado;
- O construtor pode receber argumentos;
- Uma classe pode ter vários construtores (sobrecarga);
- O programador define a lógica contida no construtor.



Construtor - exemplo - UML

Conta
-saldo: float -titular: String -custoPorOperacao: float
+Conta() +Conta(String titular, float saldo) +Conta(String titular, float saldo, float custoPorOperacao) +sacar(float valor) : boolean +depositar(float valor) : void +exibirInformacoes() : void



Construtor - exemplo - UML

```
public Conta(){  
}
```

```
public Conta (String titular, float saldo){  
    this.titular = titular;  
    this.saldo = saldo;  
}
```



Modificadores de acesso

- São padrões de visibilidade de acessos às classes, atributos e métodos;
- 4 tipos: public, private, protected e default;
- Associados ao encapsulamento;
- A maioria das variáveis devem ter visibilidade private, isto ajuda na segurança e no encapsulamento;
- Programação defensiva.



Modificadores de acesso - getters e setters

```
Class Pessoa{
    private String nome;
    public void setNome(String nome){
        if( nome != null ){
            this.nome = nome;
        }
        else{
            System.out.println("Nome nao pode ser nulo!");
            System.exit(0);
        }
    }
    public String getNome(){
        return nome; // ou return this.nome
    }
}
```




Exercício

- Criar a classe Conta de acordo com a UML
 1. Só é possível usar métodos sacar, depositar e exibirInformacoes quando os 3 atributos estiverem devidamente preenchidos;
 2. A cada operação de saque ou depósito o custo de operação é debitado;
 3. O titular obrigatoriamente precisa ter nome composto. ex: “Marcos Jose”;
 4. O saldo nunca pode ser negativo;
 5. O custo por operacao tem valor minimo 1;
 6. Outros métodos podem ser criados de acordo com a demanda.



Scanner

```
import java.util.Scanner;
public class Principal {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Digite um nome:");
        String a = scanner.nextLine();
        System.out.println("O nome digitado foi " + a);

        System.out.println("Digite um numero:");
        int num = scanner.nextInt();
        System.out.println("O numero digitado foi " + num);
    }
}
```