



Prof. Dr. Fernando Almeida
proffernando.almeida@fiap.com.br





DDD (Domain Driven Design)

Java SE e Java EE



O QUE VAMOS APRENDER HOJE?

Introdução à Programação Orientada à Objetos

1

Revisão de Orientação à Objetos

4

Manipular objetos na memória

2

Criar métodos simples

5

Exemplo prático no Eclipse

3

Testar métodos simples

6

Exercícios



Introdução à Programação Orientada a Objetos

Criando métodos simples



Métodos de uma Classe



“ Métodos são conhecidos como
funções, ações de execuções ou
procedimentos (separação por blocos)
que ajudam no design do sistema



Vamos praticar?



Métodos Classe Conta

- Classe **Conta**
- Método `sacar()`
- Requisitos:
 - recebe a quantidade a ser retirada
 - faz o cálculo do novo saldo
 - informa ao usuário

· · Métodos Classe Conta

· + ·

- + ·
 - Método sacar()

```
void sacar(double quantidade) {  
    double novoSaldo = this.saldo - quantidade;  
    this.saldo = novoSaldo;  
    System.out.println("Saque realizado R$: " + quantidade);  
}
```

Métodos Classe Conta

- Classe **Conta**
- Método `depositar()`
- Requisitos:
 - recebe a quantidade a ser depositada
 - faz o cálculo do novo saldo
 - informa ao usuário o valor depositado

Métodos Classe Conta

- Método `depositar()`

```
void depositar(double quantidade) {  
    this.saldo = this.saldo + quantidade;  
    System.out.println("Você depositou R$: " + quantidade);  
}
```

Praticar



```
1 package ddd_fiap;
2
3 /**
4  * @author Fernando Almeida
5  */
6 public class Conta {
7
8     double saldo;
9     String titular;
10
11     void sacar(double quantidade) {
12         double novoSaldo = this.saldo - quantidade;
13         this.saldo = novoSaldo;
14         System.out.println("Saque realizado R$: " + quantidade);
15     }
16
17     void depositar(double quantidade) {
18         this.saldo = this.saldo + quantidade;
19         System.out.println("Você depositou R$: " + quantidade);
20     }
21 }
```



Testando a classe...



Praticar



```
12 public class TesteConta {
13     public static void main(String[] args) {
14         //instanciando uma conta
15         Conta minhaConta = new Conta();
16
17         //populando a Conta
18         minhaConta.titular = "Dono da Conta";
19         minhaConta.saldo = 1500.00;
20
21         //exibindo saldo
22         System.out.println("saldo: R$ " + minhaConta.saldo);
23
24         //sacando R$ 350.00
25         minhaConta.sacar(350);
26
27         //exibindo saldo
28         System.out.println("saldo: R$ " + minhaConta.saldo);
29
30         //depositando R$ 1500.00
31         minhaConta.depositar(1500);
32
33         //exibindo saldo
34         System.out.println("saldo: R$ " + minhaConta.saldo);
35     }
36 }
```



Adicionando novos requisitos!!!



Modelando a Classe **Funcionario**

- Diagrama UML da classe **Funcionário**
- Atributos:
 - nome do funcionário (String)
 - departamento (String)
 - salário (double)
 - data admissão (String)
 - RG (String)

Modelando a Classe **Funcionario**

- Diagrama UML da classe **Funcionário**
- Funcionalidades:
 - imprimir o nome do funcionário
 - imprimir o salário do funcionário
 - aumentar o salário do funcionário
 - calculo do ganho anual

Praticar



Escreva a classe **Funcionário** em Java

```
1 package ddd_fiap;
2
3
4 /**
5  *
6  * @author Fernando
7  */
8 public class Funcionario {
9     double salario;
10    //demais atributos e métodos da classe
11
12    void receberAumento(double aumento){
13        //implementar o método
14    }
15
16    void calcularGanhoAnual(){
17        //implementar o método
18    }
19 }
```

Praticar



Escreva a classe **TestaFuncionário**
em Java

Modelando a Classe **Funcionario**

Praticar



- Diagrama UML da classe **Funcionário**
- mais uma Funcionalidade:
 - imprimir todos os dados do funcionário

Obrigado e até a próxima aula!

FIAP

Copyright © 2022 | Professor Fernando Luiz de Almeida

Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente proibido sem consentimento formal, por escrito, do professor/autor.