Projeto Java: Sistema Bancário Orientado a Objetos

Prof. Wilker Luz

1 de abril de 2025

Objetivos da Aula

- Compreender os conceitos de Programação Orientada a Objetos (POO).
- Aplicar os pilares da POO: encapsulamento, objetos e classes.
- Desenvolver um sistema bancário simples utilizando Java.
- Utilizar o Scanner para entrada de dados e criar menus interativos.

Estrutura do Projeto

O projeto é organizado seguindo boas práticas de separação em pacotes:

```
BancoJava/
 src/
    app/
       Principal.java
    model/
       Cliente.java
    view/
        EntradaDados.java
```

Classe Cliente.java

Local: `model/Cliente.java`

```java
package model;

public class Cliente {
    private String nome;
    private double saldo;

    public Cliente(String nome, double saldoInicial) {
        this.nome = nome;
        this.saldo = saldoInicial;
    }

```

```
12      public String getNome() {
13          return nome;
14      }
15
16      public double getSaldo() {
17          return saldo;
18      }
19
20      public void depositar(double valor) {
21          if (valor > 0) {
22              saldo += valor;
23              System.out.println("Dep sito de R$" + valor + "
                    realizado com sucesso.");
24          } else {
25              System.out.println("Valor inv lido para dep sito.")
                    ;
26          }
27      }
28
29      public void sacar(double valor) {
30          if (valor > 0 && saldo >= valor) {
31              saldo -= valor;
32              System.out.println("Saque de R$" + valor + "
                    realizado com sucesso.");
33          } else {
34              System.out.println("Saldo insuficiente ou valor
                    inv lido.");
35          }
36      }
37
38      public void transferir(Cliente destino, double valor) {
39          if (valor > 0 && saldo >= valor) {
40              saldo -= valor;
41              destino.depositar(valor);
42              System.out.println("Transfer ncia de R$" + valor + "
                     para " + destino.getNome() + " realizada.");
43          } else {
44              System.out.println("Transfer ncia  n  o realizada.
                    Verifique o saldo ou o valor.");
45          }
46      }
47
48      public void mostrarSaldo() {
49          System.out.println(nome + " - Saldo atual: R$" + saldo);
50      }
51  }
```

Listing 1: Classe Cliente

Classe EntradaDados.java

Local: `view/EntradaDados.java`

```
package view;

import java.util.Scanner;
import model.Cliente;

public class EntradaDados {
    public static Cliente criarCliente(Scanner input) {
        System.out.print("Digite o nome do cliente: ");
        String nome = input.nextLine();
        System.out.print("Digite o saldo inicial: ");
        double saldo = input.nextDouble();
        input.nextLine(); // consumir quebra de linha
        return new Cliente(nome, saldo);
    }
}
```

Listing 2: Classe EntradaDados

Classe Principal.java

Local: `app/Principal.java`

```
package app;

import java.util.Scanner;
import model.Cliente;
import view.EntradaDados;

public class Principal {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.println("Criando Cliente 1:");
        Cliente cliente1 = EntradaDados.criarCliente(input);

        System.out.println("Criando Cliente 2:");
        Cliente cliente2 = EntradaDados.criarCliente(input);

        int opcao;
        do {
            System.out.println("\n========= MENU =========");
            System.out.println("1 - Depositar");
            System.out.println("2 - Sacar");
            System.out.println("3 - Transferir");
            System.out.println("4 - Ver Saldo");
            System.out.println("0 - Sair");
            System.out.print("Escolha uma op   o: ");
            opcao = input.nextInt();
```

```
27              input.nextLine();
28
29              switch (opcao) {
30                  case 1:
31                      System.out.print("Depositar para qual cliente
                            (1 ou 2)? ");
32                      int clienteDep = input.nextInt();
33                      System.out.print("Valor a depositar: ");
34                      double valorDep = input.nextDouble();
35                      input.nextLine();
36                      if (clienteDep == 1) {
37                          cliente1.depositar(valorDep);
38                      } else if (clienteDep == 2) {
39                          cliente2.depositar(valorDep);
40                      } else {
41                          System.out.println("Cliente inv lido.");
42                      }
43                      break;
44
45                  case 2:
46                      System.out.print("Sacar de qual cliente (1 ou
                            2)? ");
47                      int clienteSaq = input.nextInt();
48                      System.out.print("Valor a sacar: ");
49                      double valorSaq = input.nextDouble();
50                      input.nextLine();
51                      if (clienteSaq == 1) {
52                          cliente1.sacar(valorSaq);
53                      } else if (clienteSaq == 2) {
54                          cliente2.sacar(valorSaq);
55                      } else {
56                          System.out.println("Cliente inv lido.");
57                      }
58                      break;
59
60                  case 3:
61                      System.out.print("Transferir de qual cliente
                           (1 ou 2)? ");
62                      int de = input.nextInt();
63                      System.out.print("Para qual cliente (1 ou 2)?
                            ");
64                      int para = input.nextInt();
65                      System.out.print("Valor a transferir: ");
66                      double valorTransf = input.nextDouble();
67                      input.nextLine();
68                      if (de == 1 && para == 2) {
69                          cliente1.transferir(cliente2, valorTransf
                               );
70                      } else if (de == 2 && para == 1) {
71                          cliente2.transferir(cliente1, valorTransf
                               );
```

```
72                      } else {
73                          System.out.println("Transfer ncia
                               inv lida.");
74                      }
75                      break;
76
77                  case 4:
78                      cliente1.mostrarSaldo();
79                      cliente2.mostrarSaldo();
80                      break;
81
82                  case 0:
83                      System.out.println("Saindo do sistema...");
84                      break;
85
86                  default:
87                      System.out.println("Op   o  inv lida.");
88              }
89
90          } while (opcao != 0);
91
92          input.close();
93      }
94  }
```

Listing 3: Classe Principal (menu interativo)

Execução no VS Code

1. Instale o Java JDK e a extensão `Extension Pack for Java`.
2. Extraia o projeto `BancoJava.zip`.
3. Abra a pasta no VS Code.
4. Execute os comandos no terminal:

```
cd src
javac app/Principal.java
java app.Principal
```

Exercício Proposto

- Adicione um terceiro cliente.
- Implemente validações para evitar valores negativos.
- Crie um relatório final com todos os saldos ao sair do programa.