



UNIVERSIDADE ANHANGUERA

TABOÃO DA SERRA

**PORTIFÓLIO – RELATÓRIO DE AULA PRÁTICA  
LINGUAGEM ORIENTADA A OBJETOS**

**CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE  
SISTEMAS**

**Helen Barros Lopes / RA: 3474905901**

Tutor Presencial: Welton

Tutor a Distância: Marcio Akio Shimoda

## SUMÁRIO

INTRODUÇÃO.....	3
MÉTODO E DESENVOLVIMENTO.....	4
ATIVIDADE PROPOSTA.....	4
CONCLUSÃO.....	8
REFERÊNCIAS.....	9

## 1 INTRODUÇÃO

A programação orientada a objetos é uma das metodologias mais populares e eficazes para a criação de softwares complexos e escaláveis. Com a crescente demanda por softwares mais robustos e interativos, o conhecimento em programação orientada a objetos se tornou indispensável para desenvolvedores de software.

A aula prática de Linguagem Orientada a Objetos tem como objetivo fixar os conceitos de programação orientada a objetos e empregar práticas de instanciação de classes Java, utilização de métodos e atributos em Java, além de utilizar pacotes da linguagem Java para implementação de software com paradigma de orientação a objetos.

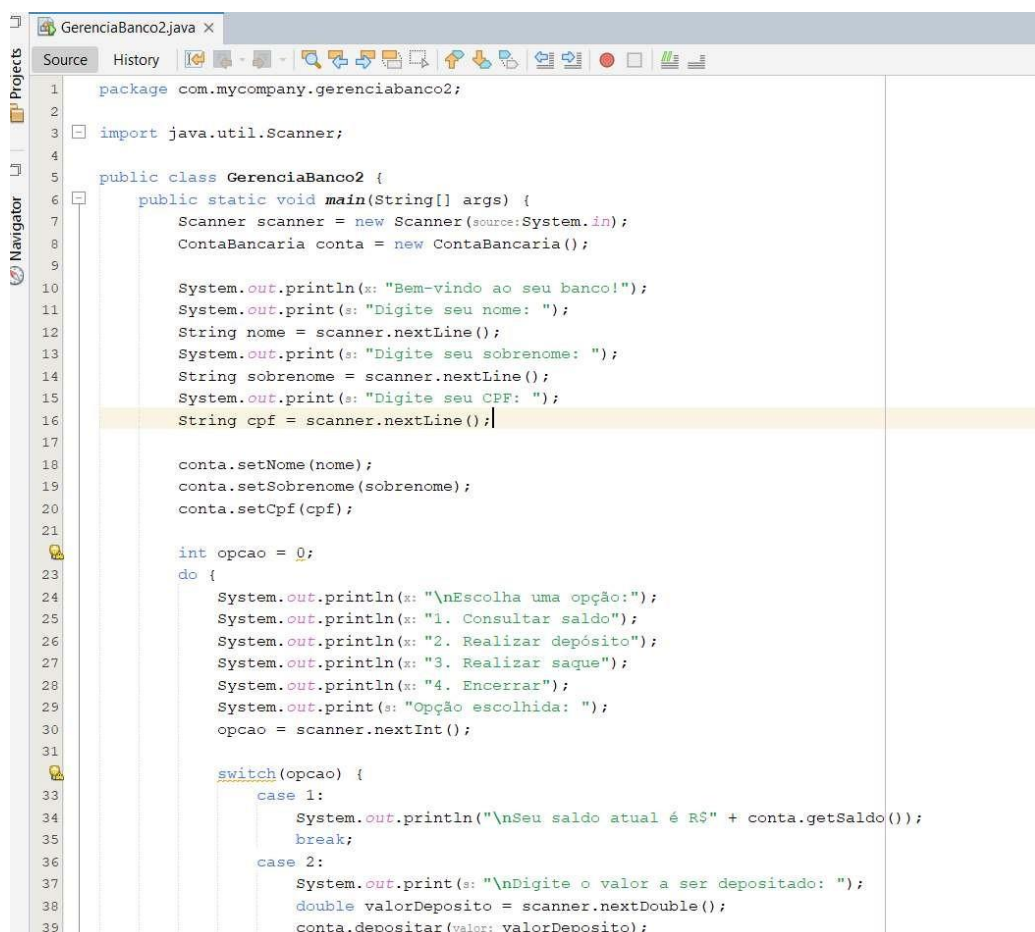
Durante a aula, será utilizado o software IDE NetBeans, uma ferramenta gratuita e de código aberto, para a criação de uma pequena aplicação de gerenciamento bancário que possibilita ao usuário informar seus dados pessoais e realizar operações bancárias. Atividades práticas como essa são fundamentais para a fixação dos conceitos e para o desenvolvimento de habilidades em programação. Além disso, permitem aplicar de forma prática os conhecimentos adquiridos em sala de aula. Com a execução deste projeto, há a oportunidade de aprimorar habilidades em programação orientada a objetos e possibilita uma visão mais clara de como aplicar esses conceitos em projetos reais.

## 2 MÉTODO E DESENVOLVIMENTO

Utilizando os principais conceitos do paradigma de Orientação a Objetos, este portfólio de aula prática apresenta a criação de uma pequena aplicação de gerenciamento bancário que possibilite ao usuário informar seu nome, sobrenome e CPF. Além disso, a aplicação possui a possibilidade de o usuário consultar saldo, realizar depósitos e saques

## 3 ATIVIDADE PROPOSTA

- O código escrito para realização do solicitado é o seguinte:



```

1 package com.mycompany.gerenciabanco2;
2
3 import java.util.Scanner;
4
5 public class GerenciaBanco2 {
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8         ContaBancaria conta = new ContaBancaria();
9
10        System.out.println(x: "Bem-vindo ao seu banco!");
11        System.out.print(s: "Digite seu nome: ");
12        String nome = scanner.nextLine();
13        System.out.print(s: "Digite seu sobrenome: ");
14        String sobrenome = scanner.nextLine();
15        System.out.print(s: "Digite seu CPF: ");
16        String cpf = scanner.nextLine();
17
18        conta.setNome(nome);
19        conta.setSobrenome(sobrenome);
20        conta.setCpf(cpf);
21
22        int opcao = 0;
23        do {
24            System.out.println(x: "\nEscolha uma opção:");
25            System.out.println(x: "1. Consultar saldo");
26            System.out.println(x: "2. Realizar depósito");
27            System.out.println(x: "3. Realizar saque");
28            System.out.println(x: "4. Encerrar");
29            System.out.print(s: "Opção escolhida: ");
30            opcao = scanner.nextInt();
31
32            switch(opcao) {
33                case 1:
34                    System.out.println("\nSeu saldo atual é R$" + conta.getSaldo());
35                    break;
36                case 2:
37                    System.out.print(s: "\nDigite o valor a ser depositado: ");
38                    double valorDeposito = scanner.nextDouble();
39                    conta.depositar(valor: valorDeposito);

```

```

GerenciaBanco2.java X
Source History
40      System.out.println(x: "\nDepósito realizado com sucesso!");
41      break;
42      case 3:
43          System.out.print(s: "\nDigite o valor a ser sacado: ");
44          double valorSaque = scanner.nextDouble();
45          if(conta.sacar(valor: valorSaque)) {
46              System.out.println(x: "\nSaque realizado com sucesso!");
47          } else {
48              System.out.println(x: "\nSaldo insuficiente!");
49          }
50          break;
51      case 4:
52          System.out.println(x: "\nObrigado por utilizar nosso banco!");
53          break;
54      default:
55          System.out.println(x: "\nOpção inválida! Digite novamente.");
56      }
57      } while (opcao != 4);
58  }
59  }
60
61  class ContaBancaria {
62      private String nome;
63      private String sobrenome;
64      private String cpf;
65      private double saldo;
66
67      public void setName(String nome) {
68          this.nome = nome;
69      }
70
71      public void setSobrenome(String sobrenome) {
72          this.sobrenome = sobrenome;
73      }
74
75      public void setCpf(String cpf) {
76          this.cpf = cpf;
77      }
78  }

```

```

GerenciaBanco2.java X
Source History
71      public void setSobrenome(String sobrenome) {
72          this.sobrenome = sobrenome;
73      }
74
75      public void setCpf(String cpf) {
76          this.cpf = cpf;
77      }
78
79      public double getSaldo() {
80          return saldo;
81      }
82
83      public void depositar(double valor) {
84          saldo += valor;
85      }
86
87      public boolean sacar(double valor) {
88          if(valor <= saldo) {
89              saldo -= valor;
90              return true;
91          } else {
92              return false;
93          }
94      }
95  }
96

```

- Obteve-se, portanto, o seguinte resultado:

```
Output - Run (gerenciaBanco2) ×
--- exec-maven-plugin:3.1.0:exec (default-cli) @ gerenciaBanco2 ---
Bem-vindo ao seu banco!
Digite seu nome: Teste
Digite seu sobrenome: Testando
Digite seu CPF: 917.589.758-49

Escolha uma opção:
1. Consultar saldo
2. Realizar depósito
3. Realizar saque
4. Encerrar
Opção escolhida: 1

Seu saldo atual é R$0.0

Escolha uma opção:
1. Consultar saldo
2. Realizar depósito
3. Realizar saque
4. Encerrar
Opção escolhida: 2

Digite o valor a ser depositado: 1000,00

Depósito realizado com sucesso!

Escolha uma opção:
1. Consultar saldo
2. Realizar depósito
3. Realizar saque
4. Encerrar
Opção escolhida: 1

Seu saldo atual é R$1000.0

Escolha uma opção:
1. Consultar saldo
2. Realizar depósito
3. Realizar saque
4. Encerrar
Opção escolhida: 3
```

Digite o valor a ser sacado: 1000,00

Saque realizado com sucesso!

Escolha uma opção:

1. Consultar saldo
2. Realizar depósito
3. Realizar saque
4. Encerrar

Opção escolhida: 1

Seu saldo atual é R\$0.0

Escolha uma opção:

1. Consultar saldo
2. Realizar depósito
3. Realizar saque
4. Encerrar

Opção escolhida: 4

Obrigado por utilizar nosso banco!

-----  
**BUILD SUCCESS**  
-----

Total time: 01:13 min

Finished at: 2023-05-05T14:38:05-03:00  
-----

## 4 CONCLUSÃO

Neste portfólio de aula prática, foram abordados conceitos fundamentais da linguagem de programação Java, como variáveis, operadores, estruturas de controle de fluxo e programação orientada a objetos.

Foi apresentado o ambiente de desenvolvimento integrado NetBeans IDE, utilizado para facilitar a criação, edição e depuração de códigos Java, bem como a sua integração com o gerenciador de dependências Maven.

Por meio de exemplos práticos, foram explorados os conceitos de herança, polimorfismo, encapsulamento e abstração, demonstrando como essas características são aplicadas em projetos Java e como podem contribuir para o desenvolvimento de aplicações robustas, flexíveis e escaláveis.

Ao final, uma aplicação de gerenciamento bancário foi criada, utilizando os conceitos e técnicas abordados ao longo do portfólio, como a criação de classes, métodos e estruturas de decisão. Essa aplicação permitiu praticar os conhecimentos adquiridos e consolidar o aprendizado de forma mais concreta.

Conclui-se que o Java é uma linguagem de programação bastante popular e utilizada em diversos setores da indústria de software, desde a criação de aplicações desktop até o desenvolvimento de sistemas web e mobile. Portanto, compreender seus conceitos fundamentais e técnicas avançadas é essencial para se tornar um programador Java competente e atualizado.



## 5 REFERÊNCIAS

CERQUEIRA, Rildo Falcão. Fundamentos de Banco de Dados. 1ª ed. São Paulo: Editora Pearson, 2013.

DATE, C. J. Introdução a Sistemas de Bancos de Dados. 8ª ed. Rio de Janeiro: Elsevier, 2004.

MACHADO, Felipe Nery Rodrigues; ABREU, Maurício Pereira de. Banco de Dados: Teoria e Prática. 1ª ed. São Paulo: Novatec Editora, 2019.

OLIVEIRA, R. T. Banco de Dados - Modelagem, Projeto e Implementação. 1ª ed. São Paulo: Editora Érica, 2014.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. S. Sistemas de Banco de Dados. 6ª ed. São Paulo: Pearson Addison Wesley, 2012.