

ANALISE E DESENVOLVIMENTO DE SISTEMAS

YZAU HIAGO DE ARAUJO LIMA

**LINGUAGEM ORIENTADA A OBJETOS:**

Aplicação de gerenciamento bancário.

Cidade: PATOS-PB

Ano: 2024

YZAU HIAGO DE ARAUJO LIMA

**LINGUAGEM ORIENTADA A OBJETOS:**

Aplicação de gerenciamento bancário.

Cidade: PATOS-PB

Ano: 2024

## SUMÁRIO

1 INTRODUÇÃO .....	3
2 DESENVOLVIMENTO .....	4
3 CONCLUSÃO.....	6

## 1 INTRODUÇÃO

O código apresentado é um exemplo de um simples sistema de gerenciamento de banco que permite ao usuário interagir com uma conta bancária fictícia através do terminal. O projeto inicia importando a classe Scanner para permitir a entrada de dados pelo usuário e utiliza a classe GerenciaBanco para gerenciar as operações bancárias.

Ao longo do código, são utilizados conceitos fundamentais da programação orientada a objetos (POO), como a criação de classes, encapsulamento, e a utilização de objetos para representar entidades do mundo real, como uma conta bancária. A compreensão e aplicação dos conceitos de POO, como vistos nesse projeto, são fundamentais para o desenvolvimento de software robusto e escalável.

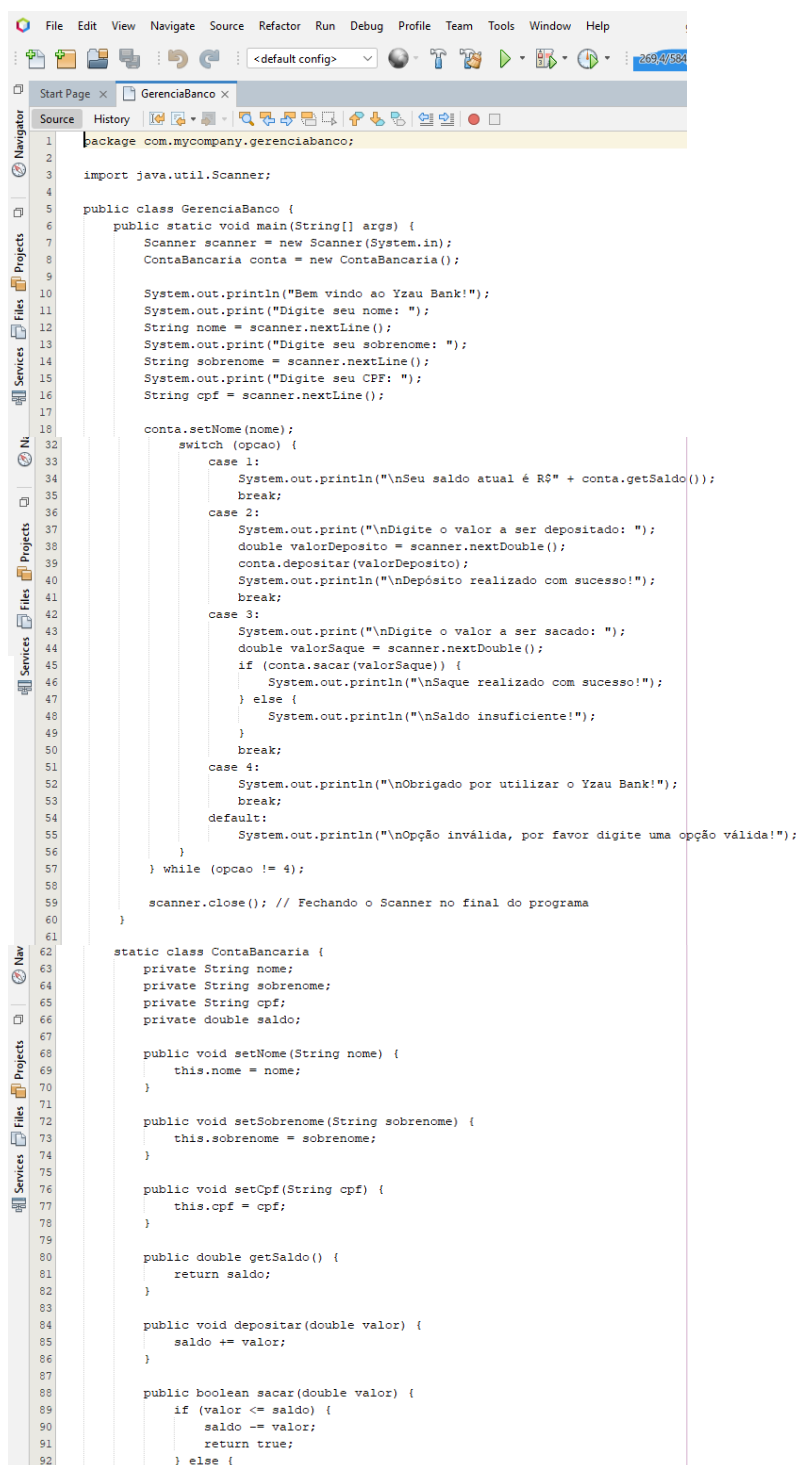
Atividade proposta: Utilizando os principais conceitos do paradigma de Orientação a Objetos, crie uma pequena aplicação de gerenciamento bancário que possibilite ao usuário informar seu nome, sobrenome e CPF. Além disso, a aplicação deverá possibilitar ao usuário consultar saldo, realizar depósitos e saques. Esses procedimentos devem se repetir até que o usuário escolha encerrar o uso da aplicação.

## 2 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

Este código é uma aplicação simples de um sistema de gerenciamento de banco, onde o usuário pode criar uma conta bancária fictícia e realizar operações como consultar saldo, realizar depósitos, realizar saques e encerrar a sessão.

### Ilustração do código:



```

1 package com.mycompany.gerenciaBanco;
2
3 import java.util.Scanner;
4
5 public class GerenciaBanco {
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8         ContaBancaria conta = new ContaBancaria();
9
10        System.out.println("Bem vindo ao Yzau Bank!");
11        System.out.print("Digite seu nome: ");
12        String nome = scanner.nextLine();
13        System.out.print("Digite seu sobrenome: ");
14        String sobrenome = scanner.nextLine();
15        System.out.print("Digite seu CPF: ");
16        String cpf = scanner.nextLine();
17
18        conta.setNome(nome);
19
20        switch (opcao) {
21            case 1:
22                System.out.println("\nSeu saldo atual é R$ " + conta.getSaldo());
23                break;
24            case 2:
25                System.out.print("\nDigite o valor a ser depositado: ");
26                double valorDeposito = scanner.nextDouble();
27                conta.depositar(valorDeposito);
28                System.out.println("\nDepósito realizado com sucesso!");
29                break;
30            case 3:
31                System.out.print("\nDigite o valor a ser sacado: ");
32                double valorSaque = scanner.nextDouble();
33                if (conta.sacar(valorSaque)) {
34                    System.out.println("\nSaque realizado com sucesso!");
35                } else {
36                    System.out.println("\nSaldo insuficiente!");
37                }
38                break;
39            case 4:
40                System.out.println("\nObrigado por utilizar o Yzau Bank!");
41                break;
42            default:
43                System.out.println("\nOpção inválida, por favor digite uma opção válida!");
44        }
45    } while (opcao != 4);
46
47    scanner.close(); // Fechando o Scanner no final do programa
48 }
49
50 static class ContaBancaria {
51     private String nome;
52     private String sobrenome;
53     private String cpf;
54     private double saldo;
55
56     public void setNome(String nome) {
57         this.nome = nome;
58     }
59
60     public void setSobrenome(String sobrenome) {
61         this.sobrenome = sobrenome;
62     }
63
64     public void setCpf(String cpf) {
65         this.cpf = cpf;
66     }
67
68     public double getSaldo() {
69         return saldo;
70     }
71
72     public void depositar(double valor) {
73         saldo += valor;
74     }
75
76     public boolean sacar(double valor) {
77         if (valor <= saldo) {
78             saldo -= valor;
79             return true;
80         } else {
81             return false;
82         }
83     }
84 }

```

```
93         return false;
94     }
95 }
96
97 }
98
```

## Ilustração do resultado:

```
File Edit View Navigate Source Refactor Run Debug Profile Te
<default config>
Output - Run (gerenciaBanco)
--- exec:3.1.0:exec (default-cli) @ gerenciaBanco ---
 Bem vindo ao Yzau Bank!
 Digite seu nome: yzau
 Digite seu sobrenome: hiago
 Digite seu CPF: 12345678900

 Escolha uma opcao:
 1. Consultar saldo
 2. Realizar deposito
 3. Realizar Saque
 4. Encerrar e sair
 Opcao escolhida: 2

 Digite o valor a ser depositado: 100

 Depoimento realizado com sucesso!

 Escolha uma opcao:
 1. Consultar saldo
 2. Realizar deposito
 3. Realizar Saque
 4. Encerrar e sair
 Opcao escolhida: 1

 Seu saldo atual ♦ R$100.0

 Escolha uma opcao:
 1. Consultar saldo
 2. Realizar deposito
 3. Realizar Saque
 4. Encerrar e sair
 Opcao escolhida: 3

 Digite o valor a ser sacado: 50

 Saque realizado com sucesso!

 Escolha uma opcao:
 1. Consultar saldo
 2. Realizar deposito
 3. Realizar Saque
 4. Encerrar e sair
 Opcao escolhida: 1

 Seu saldo atual ♦ R$50.0

 Escolha uma opcao:
 1. Consultar saldo
 2. Realizar deposito
 3. Realizar Saque
 4. Encerrar e sair
 Opcao escolhida: 4

 Obrigado por utilizar o Yzau Bank!

 BUILD SUCCESS

 Total time: 38.495 s
 Finished at: 2024-04-22T21:52:25-03:00
|
```

### 3. CONCLUSÃO

O código apresentado é uma aplicação simples de um sistema de gerenciamento de banco em Java, que permite ao usuário criar uma conta bancária fictícia e realizar operações como consultar saldo, realizar depósitos, realizar saques e encerrar a sessão.

O programa utiliza a classe Scanner para permitir a entrada de dados pelo usuário e a classe ContaBancaria para representar a conta do usuário. O menu de opções é exibido em um loop do-while e o código utiliza um switch-case para lidar com cada opção escolhida pelo usuário.

A classe ContaBancaria contém os detalhes da conta bancária, como nome, sobrenome, CPF e saldo, e métodos como depositar e sacar permitem ao usuário adicionar e retirar dinheiro da conta.

## REFERÊNCIA

COORDENAÇÃO DE APERFEIÇOAMENTO DE PESSOAL DE NÍVEL SUPERIOR(CAPES). Orientação Capes: Combate ao plágio 2011. Disponível em [https://www.capes.gov.br/images/stories/download/diversos/OrientacoesCapes\\_CombateAoPlagio.pdf](https://www.capes.gov.br/images/stories/download/diversos/OrientacoesCapes_CombateAoPlagio.pdf) - Acesso em 22/08/2019

UNIVERSIDADE FEDERAL DO PARANÁ. Biblioteca Central. **Normas para apresentação de trabalhos**. 2. ed. Curitiba: UFPR, 1992. v. 2.