

PORTIFÓLIO

DATA: 14/10/2023

CURSO: SISTEMAS DE INFORMAÇÃO – 3º SEMESTRE

MATÉRIA: LINGUAGEM ORIENTADA A OBJETOS

NOME DO PROFESSOR: LEONARDO SANTIAGO SIDON DA ROCHA

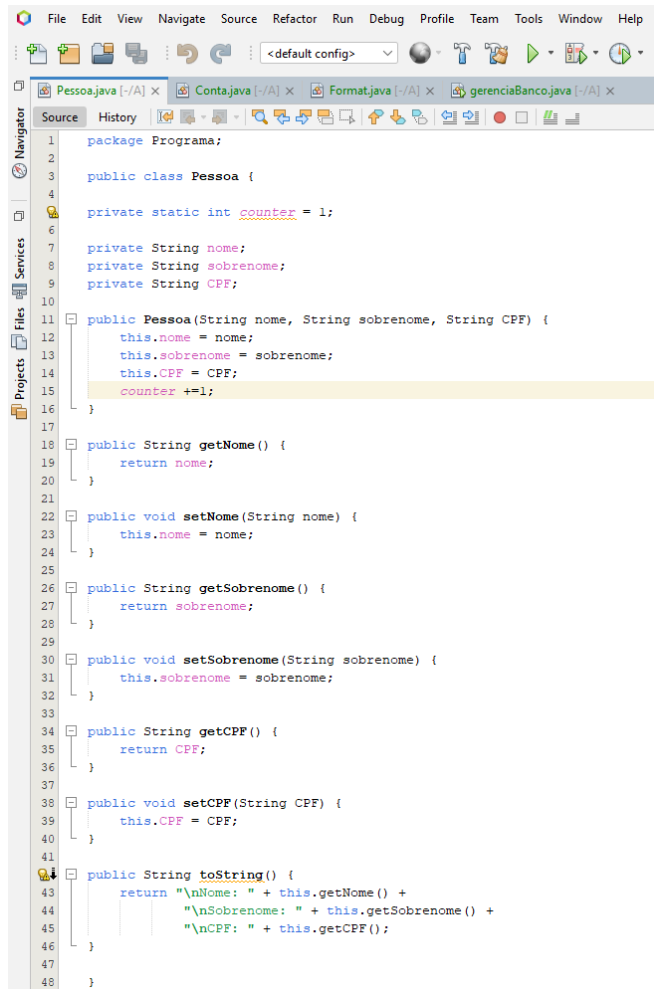
NOME DO ALUNO: FERNANDO HENRIQUE PANINI

OBJETIVO

Criar um Sistema Bancário utilizando o Programa NetBeans e a Linguagem Java, onde o Cliente possa: Abrir Conta, Verificar Saldo, Depositar, Sacar e Encerrar o Uso da Aplicação.

MÉTODOS

Classe Pessoa, com os Códigos para a Entrada de Dados Pessoais para Abertura da Conta:



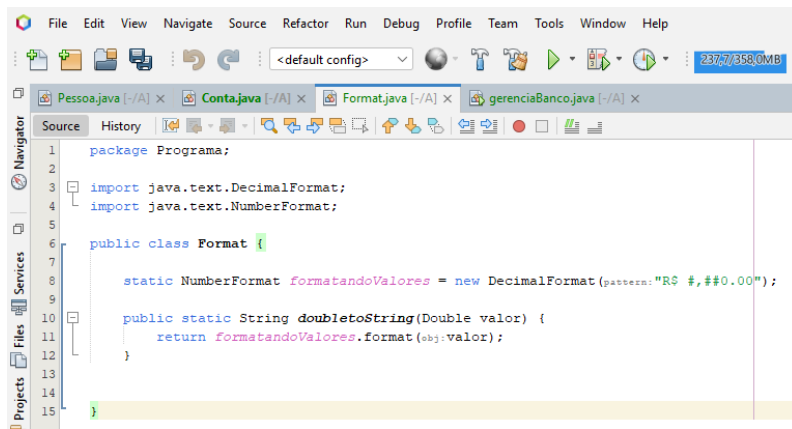
```
1 package Programa;
2
3 public class Pessoa {
4
5     private static int counter = 1;
6
7     private String nome;
8     private String sobrenome;
9     private String CPF;
10
11     public Pessoa(String nome, String sobrenome, String CPF) {
12         this.nome = nome;
13         this.sobrenome = sobrenome;
14         this.CPF = CPF;
15         counter +=1;
16     }
17
18     public String getNome() {
19         return nome;
20     }
21
22     public void setNome(String nome) {
23         this.nome = nome;
24     }
25
26     public String getSobrenome() {
27         return sobrenome;
28     }
29
30     public void setSobrenome(String sobrenome) {
31         this.sobrenome = sobrenome;
32     }
33
34     public String getCPF() {
35         return CPF;
36     }
37
38     public void setCPF(String CPF) {
39         this.CPF = CPF;
40     }
41
42     public String toString() {
43         return "\nNome: " + this.getNome() +
44             "\nSobrenome: " + this.getSobrenome() +
45             "\nCPF: " + this.getCPF();
46     }
47
48 }
```

Classe Conta, com os Códigos para Abrir Conta, Verificar Saldo, Depositar e Sacar:



```
1 package Programa;
2
3 public class Conta {
4
5     private static int contadorDeContas = 1;
6
7     private int numeroConta;
8     private Pessoa pessoa;
9     private Double saldo = 0.0;
10
11     public Conta(Pessoa pessoa) {
12         this.numeroConta = contadorDeContas;
13         this.pessoa = pessoa;
14         contadorDeContas += 1;
15     }
16
17     public int getNumeroConta() {
18         return numeroConta;
19     }
20
21     public void setNumeroConta(int numeroConta) {
22         this.numeroConta = numeroConta;
23     }
24
25     public Pessoa getPessoa() {
26         return pessoa;
27     }
28
29     public void setPessoa(Pessoa pessoa) {
30         this.pessoa = pessoa;
31     }
32
33     public Double getSaldo() {
34         return saldo;
35     }
36
37     public void setSaldo(Double saldo) {
38         this.saldo = saldo;
39     }
40
41     public String toString() {
42         return "\nNúmero da Conta: " + this.getNumeroConta() +
43             "\nNome: " + this.pessoa.getNome() +
44             "\nSobrenome: " + this.pessoa.getSobrenome() +
45             "\nCPF: " + this.pessoa.getCPF() +
46             "\nSaldo: " + Format.doubleToString(valor: this.getSaldo()) +
47             "\n";
48     }
49
50     public void depositar(Double valor) {
51         if (valor > 0) {
52             setSaldo(getSaldo() + valor);
53             System.out.println("Seu depósito foi realizado com sucesso!");
54         }
55         else {
56             System.out.println("Não foi possível realizar seu depósito");
57         }
58     }
59
60
61
62
63     public void sacar(Double valor) {
64         if(valor > 0 && this.getSaldo() >= valor){
65             setSaldo(getSaldo() - valor);
66             System.out.println("Saque foi realizado com sucesso!");
67         }
68         else {
69             System.out.println("Não foi possível realizar o saque!");
70         }
71     }
72
73
74 }
```

Código para Formatação dos Valores:



The screenshot shows an IDE window with the following code in the editor:

```
1 package Programa;
2
3 import java.text.DecimalFormat;
4 import java.text.NumberFormat;
5
6 public class Format {
7
8     static NumberFormat formatandoValores = new DecimalFormat(pattern: "R$ #,##0.00");
9
10    public static String doubletoString(Double valor) {
11        return formatandoValores.format(obj: valor);
12    }
13
14
15 }
```

The IDE interface includes a menu bar (File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help), a toolbar with icons for file operations and execution, and a sidebar with a Project Explorer and a Services view. The Project Explorer shows a project named 'Programa' with a package 'Programa' and a class 'Format.java'.

Classe Gerencia Banco, com a Criação dos Menus e os Códigos para Abrir Conta, Verificar Saldo, Depositar, Sacar e Encerrar o Programa.

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window
<default config>
Pessoa.java [~/A] x Conta.java [~/A] x Format.java [~/A] x gerenciaBanco.java [~/A] x
Source History
package Programa;

import java.util.ArrayList;
import java.util.Scanner;

public class gerenciaBanco {

    static Scanner input = new Scanner(System.in);
    static ArrayList<Conta> contasBancarias;

    public static void main(String[] args) {
        contasBancarias = new ArrayList<Conta>();
        operacoes();
    }

    public static void operacoes() {
        System.out.println("-----Bem vindo ao Banco-----");
        System.out.println("*****");
        System.out.println("-----Selecione uma opção-----");
        System.out.println("*****");
        System.out.println("Opção 1 - Abrir Conta");
        System.out.println("Opção 2 - Depositar");
        System.out.println("Opção 3 - Sacar");
        System.out.println("Opção 4 - Listar");
        System.out.println("Opção 5 - Sair");
    }
}
```

```
int operacao = input.nextInt();

switch(operacao) {
    case 1:
        abrirConta();
        break;
    case 2:
        depositar();
        break;
    case 3:
        sacar();
        break;
    case 4:
        listar();
        break;
    case 5:
        System.out.println("Obrigado por Utilizar nossa Agência");
        System.exit(status: 0);
        break;
    default:
        System.out.println("Opção Inválida!");
        operacoes();
        break;
}
}
```

```
public static void abrirConta() {
    System.out.println("Nome: ");
    String nome = input.next();

    System.out.println("Sobrenome: ");
    String sobrenome = input.next();

    System.out.println("CPF: ");
    String cpf = input.next();

    Pessoa pessoa = new Pessoa(nome, sobrenome, cpf);

    Conta conta = new Conta(pessoa);

    contasBancarias.add(conta);
    System.out.println("Sua conta foi criada com sucesso!");

    operacoes();
}

private static Conta encontrarConta(int numeroConta) {
    Conta conta = null;
    if(contasBancarias.size() > 0) {
        for(Conta c: contasBancarias) {
            if(c.getNumeroConta() == numeroConta) {
                conta = c;
            }
        }
    }
    return conta;
}
```

```

91
92
93 public static void depositar(){
94     System.out.println("Número da conta: ");
95     int numeroConta = input.nextInt();
96
97     Conta conta = encontrarConta(numeroConta);
98     if(conta != null) {
99         System.out.println("Qual valor deseja depositar? ");
100         Double valorDeposito = input.nextDouble();
101         conta.depositar(valorDeposito);
102         System.out.println("Valor depositado com sucesso! ");
103     }
104     else {
105         System.out.println("Conta não encontrada! ");
106     }
107     operacoes();
108 }
109 public static void sacar() {
110     System.out.println("Número da conta: ");
111     int numeroConta = input.nextInt();
112
113     Conta conta = encontrarConta(numeroConta);
114
115     if(conta != null) {
116         System.out.println("Qual valor deseja sacar? ");
117         Double valorSaque = input.nextDouble();
118         conta.sacar(valorSaque);
119         System.out.println("Valor sacado com sucesso! ");
120     }
121     else {
122         System.out.println("Conta não encontrada! ");
123     }
124     operacoes();
125 }
126 public static void listar() {
127     if(contasBancarias.size() > 0) {
128         for(Conta conta: contasBancarias) {
129             System.out.println(conta);
130         }
131     }
132     else {
133         System.out.println("Não há contas cadastradas!");
134     }
135     operacoes();
136 }
137
138 }

```

RESULTADOS – Programa funcionando

Abrindo a Conta:

```
Project
--- exec:3.1.0:exec (default-cli) @ gerenciaBanco ---
-----Bem vindo ao Banco-----
*****
-----Selecione uma opção-----
*****
Opção 1 - Abrir Conta
Opção 2 - Depositar
Opção 3 - Sacar
Opção 4 - Listar
Opção 5 - Sair
1

Nome:
Fernando

Sobrenome:
Panini

CPF:
352.541.214-65
Sua conta foi criada com sucesso!
```

Consultando o Saldo:

```
-----Bem vindo ao Banco-----
*****
-----Selecione uma opção-----
*****
Opção 1 - Abrir Conta
Opção 2 - Depositar
Opção 3 - Sacar
Opção 4 - Listar
Opção 5 - Sair
4

Número da Conta: 1
Nome: Fernando
Sobrenome: Panini
CPF: 352.541.214-65
Saldo: R$ 0,00
```

Depositando:

```
-----Bem vindo ao Banco-----
*****
-----Selecione uma opção-----
*****
Opção 1 - Abrir Conta
Opção 2 - Depositar
Opção 3 - Sacar
Opção 4 - Listar
Opção 5 - Sair
2
Número da conta:
1
Qual valor deseja depositar?
50
Seu depósito foi realizado com sucesso!
Valor depositado com sucesso!
```


Saldo após o Depósito:

```
-----Bem vindo ao Banco-----
*****
-----Selecione uma opção-----
*****
Opção 1 - Abrir Conta
Opção 2 - Depositar
Opção 3 - Sacar
Opção 4 - Listar
Opção 5 - Sair
4

Número da Conta: 1
Nome: Fernando
Sobrenome: Panini
CPF: 352.541.214-65
Saldo: R$ 50,00
```

Realizando o Saque:

```
-----Bem vindo ao Banco-----
*****
-----Selecione uma opção-----
*****
Opção 1 - Abrir Conta
Opção 2 - Depositar
Opção 3 - Sacar
Opção 4 - Listar
Opção 5 - Sair
3
Número da conta:
1
Qual valor deseja sacar?
25
Saque foi realizado com sucesso!
Valor sacado com sucesso!
```

Saldo Residual após o Saque:

```
-----Bem vindo ao Banco-----
*****
-----Selecione uma opção-----
*****
Opção 1 - Abrir Conta
Opção 2 - Depositar
Opção 3 - Sacar
Opção 4 - Listar
Opção 5 - Sair
4

Número da Conta: 1
Nome: Fernando
Sobrenome: Panini
CPF: 352.541.214-65
Saldo: R$ 25,00
```