



unopar

UNIVERSIDADE PITÁGORAS UNOPAR ANHANGUERA - MARAVILHA
ENGENHARIA DE SOFTWARE

NATAN OGLIARI - 34466876

LINGUAGEM ORIENTADA A OBJETOS

Maravilha/SC

2023

NATAN OGLIARI - 34466876

LINGUAGEM ORIENTADA A OBJETOS

Trabalho de portfólio apresentado como requisito parcial
para a obtenção de pontos para a média semestral.

Orientador: Leonardo Santiago Sidon da Rocha.

Maravilha/SC
2023

Sumário

	Páginas
1 Introdução	4
2 Métodos	4
3 Resultados	6
3.1 Métodos das classes	6
4 Conclusões	7
5 Anexos	9

1 Introdução

Esta presente aula prática tem por fim a aplicação dos paradigmas da linguagem orientada a objetos com a linguagem de programação Java1.

Para os procedimentos práticos foi sugerido o uso de nome *gerenciaBanco*, o qual será implementado. A finalidade desta aula prática, visa a implementação de um sistema de gerenciamento de banco, aonde o cliente deste banco irá acendero sistema através de uma interface gráfica através da biblioteca *java.swing.**, o menu deverá ser do tipo *loop* o usuário deverá escolher a opção de finalizar a operação.

As funções para este programa são:

- Incerção das credenciais (Nome, Sobrenome e cpf);
- Consulta de saldo;
- Depósito;
- Saque;
- Finaliza a operação com uma mensagem de despedida;

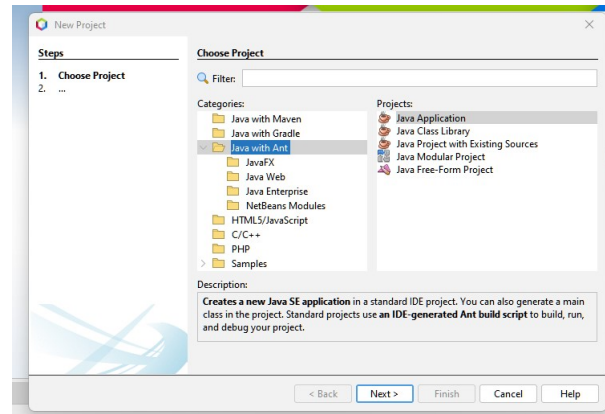
Segundo freecodecamp (2023), a linguagem orientata a objetos se define em quatro pilares, sendo elas: herança, encapsulamento, abstração e poliformismo. Isso permite definir, neste caso, umas características de **cliente**, e utiliza-lós inúmeras vezes sem a necessidade de rescreve-lá.

2 Métodos

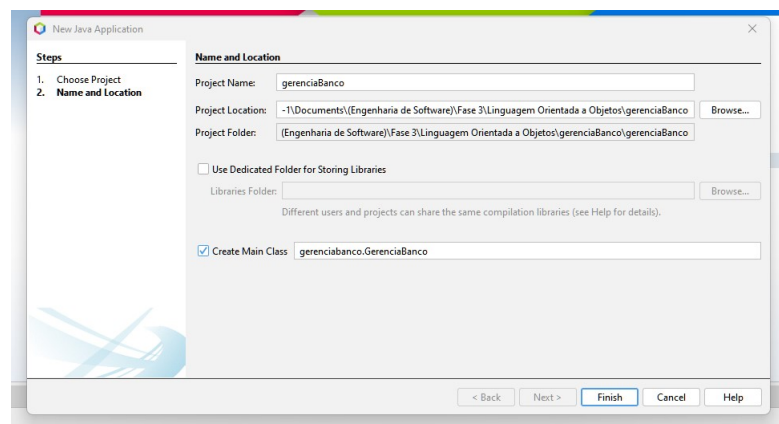
De forma continua as instruções do roteiro da aula prática, é criado o projeto em java conforme figura 2. E denominado de *gerenciaBanco*.



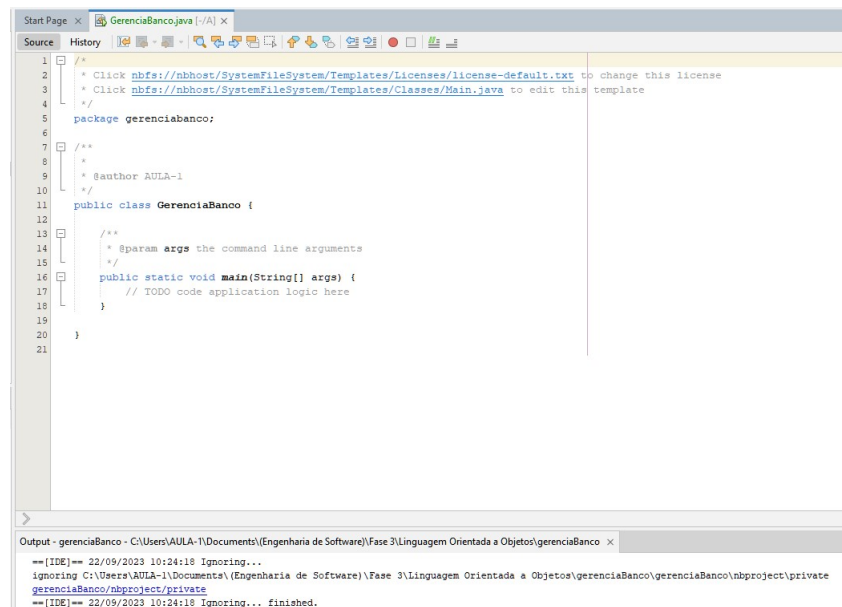
Figura 1. Logo Java, Stackify (2023)



(a) Início projeto.



(b) Nome projeto.



(c) Projeto em branco.

Figura 2. Projeto java, O autor (2023)

Para a organização desta aula prática foi confeccionado um diagrama UML para o programa, conforme demonstra a figura 3

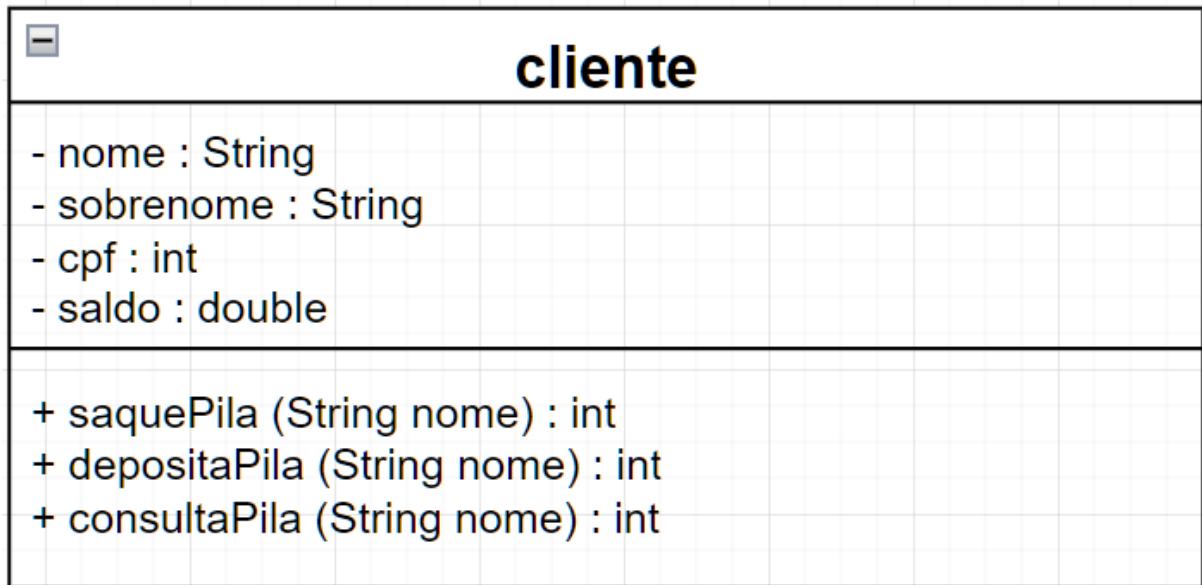


Figura 3. Diagrama UML, O autor (2023)

Os atributos definidos são:

- name : String
- sobrenome : String
- cpf : String
- saldo : double

Seguindo boas práticas de programação, define-se variáveis de controle de erros, e para tal, a criação de constante de verificação em vez de zeros e uns (0,1), do tipo *private static final int STATUS_OK = 1*; e *private static final int STATUS_FAIL = 0*;

De forma sequente foi prosseguido com as instruções do roteiro da aula prática.

3 Resultados

3.1 Métodos das classes

Os métodos definem as ações da classe, eu acesso é *minhaClasse.meuMetodo(args)*

```

1  /**
2   * Realiza a consulta do saldo de uma conta
3   * @param nome Passa o nome da conta a ser consultada e informa o
      saldo ao cliente.
4   */
5   public double consultaPilas(String nome) {
  
```

```

6      JOptionPane.showMessageDialog(null,"Seu saldo atual e de: R$" +
      this.saldo, this.nome+" "+this.sobrenome, JOptionPane.
      INFORMATION_MESSAGE, icon);
7  }

```

Listing 1. *consultaPilas*

```

1  /**
2   * Realiza um depósito em uma conta
3   * @param nome Informa o nome da conta a ser depositado
4   * @return retorna STATUS_OK se a operacao ocorreu com sucesso e
5   *         retorna STATUS_FAIL se ocorrer um erro
6   */
7  public void depositaPila( String nome){
8      try { // verifica se a entrada e do tipo numeral
9          double pilaDeposito = Double.parseDouble(JOptionPane.
10             showInputDialog(null,"Informe a quantidade em Reais
11             (R$) a ser depositado na conta do "+this.nome+" "+
12             this.sobrenome));
13             this.saldo += pilaDeposito;
14             //JOptionPane.showMessageDialog(null,"Seu Saldo e R$
15             :"+this.consultaPilas(this.nome)+" Reais", this.
16             nome, JOptionPane.INFORMATION_MESSAGE, icon);
17             return STATUS_OK;
18         }
19
20         catch (NumberFormatException e) { // imprime o erro na tela
21             e informa o que foi digitado.
22             JOptionPane.showMessageDialog(null,"Entre com valor
23             valido, do tipo numeral.\n Use (.) ponto em vez de
24             (,) virgula\n ERRO: " + e.getMessage() , "ERRO",
25             JOptionPane.ERROR_MESSAGE);
26             return STATUS_FAIL;
27         }
28     }

```

Listing 2. *depositoPilas*

4 Conclusões

I fsfsdf

II kugfhiuh

1. Anterior ... (NINGUEM, 2022)
2. Próximo ...

Referências

FREECODECAMP. **Programação orientada a objetos e programação estruturada**. 2023. Acessado em: 06 nov. 2023. Disponível em: <<https://www.freecodecamp.org/portuguese/news/os-quatro-pilares-da-programacao-orientada-a-objetos-com-javascript/>>.

NINGUEM, S. **O curioso caso do livro que ninguém escreveu**. Terra do Nunca: Editora Fantasia, 2022. Acessado em: 09 jan. 2022.

STACKIFY. **Application Logging e Exception Handling**. 2023. Acessado em: 09 set. 2023. Disponível em: <<https://stackify.com/java-logging-best-practices/>>.

5 Anexos

```
1 package gerenciabanco;
2
3 import java.util.*;
4 //import javax.swing.JOptionPane;
5 import javax.swing.*;
6 import java.awt.*;
7 import java.lang.Integer;
8 import java.lang.Exception;
9 import java.lang.Error;
10 import java.lang.reflect.Method;
11 /**
12  * @author Natan Ogliari
13  * @version 0.1
14  */
15 public class GerenciaBanco {
16
17     public static class cliente{
18         String nome;
19         String sobrenome;
20         String cpf;
21         double saldo;
22
23         private static final int STATUS_OK = 1;///define um retorno
24         caso de sucesso
25         private static final int STATUS_FAIL = 0;///define um
26         retorno caso de falha
27
28         ImageIcon icon = new ImageIcon("C:\\Users\\AULA-1\\Documents
29             \\(Engenharia de Software)\\Fase 3\\Linguagem Orientada a
```

```

Objetos\\gerenciaBanco\\gerenciaBanco\\src\\gerenciabanco
\\saracura.jpg");

27
28  /**
29  * Realiza um deposito em uma conta
30  * @param nome Informa o nome da conta a ser depositado
31  * @return retorna STATUS_OK se a operação ocorreu com sucesso
32  *       e retorna STATUS_FAIL se ocorrer um erro
33  */
34  public int depositaPila(String nome){
35
36      try { // verifica se a entrada e do tipo numeral
37          double pilaDeposito = Double.parseDouble(JOptionPane.
38              showInputDialog(null, "Informe a quantidade em Reais
39              (R$) a ser depositado na conta do "+this.nome+" "+
40              this.sobrenome));
41          this.saldo += pilaDeposito;
42          //JOptionPane.showMessageDialog(null, "Seu Saldo é R$
43          :"+this.consultaPilas(this.nome)+" Reais", this.
44          nome, JOptionPane.INFORMATION_MESSAGE, icon);
45          return STATUS_OK;
46      }
47
48      catch (NumberFormatException e) { // imprime o erro na
49          tela e informa o que foi digitado.
50          JOptionPane.showMessageDialog(null, "Entre com valor
51          válido, do tipo numeral.\n Use (.) ponto em vez de
52          (,) virgula\n ERRO: " + e.getMessage() , "ERRO",
53          JOptionPane.ERROR_MESSAGE);
54          return STATUS_FAIL;
55      }
56  }
57
58  /**
59  * Realiza o saque de uma conta
60  * @param nome Informa o nome da conta a ser realizado o saque
61  * @return retorna STATUS_OK se a operação ocorreu com sucesso
62  *       e retorna STATUS_FAIL se ocorrer um erro
63  */
64  public int saquePila(String nome){

```

```

55         try { // tratamento de exception
56             double pilaSaque = Double.parseDouble(JOptionPane
                    .showInputDialog(null, "Informe a quantidade em
                        Reais (R$) a ser sacada na conta do "+this.
                            nome+" "+this.sobrenome));
57             if (this.saldo >= pilaSaque) { // verifica se tem
                    saldo
58                 this.saldo -= pilaSaque;
59                 return STATUS_OK;
60             }
61             else {
62                 JOptionPane.showMessageDialog(null, "O valor
                        informado para o saque é maior que seu
                        saldo.\nSeu saldo é de: R$ "+this.saldo, "
                        Saldo Insuficiente", JOptionPane.
                            ERROR_MESSAGE, icon);
63                 return STATUS_FAIL;
64             }
65         }
66         catch (NumberFormatException e) {
67             JOptionPane.showMessageDialog(null, "Entre com
                    valor válido, do tipo numeral.\n ERRO: " + e.
                        getMessage() , "ERRO", JOptionPane.
                            ERROR_MESSAGE);
68             return STATUS_FAIL;
69         }
70     }
71
72     /**
73     * Realiza a consulta do saldo de uma conta
74     * @param nome Passa o nome da conta a ser consultada e
        informa o saldo ao cliente.
75     */
76     public void consultaPilas(String nome) {
77         JOptionPane.showMessageDialog(null, "Seu saldo atual é de:
                R$" +this.saldo, this.nome+" "+this.sobrenome,
                    JOptionPane.INFORMATION_MESSAGE, icon);
78     }
79 }
80
81 public static void main(String[] args) {

```

```

82
83     ImageIcon icon = new ImageIcon("C:\\Users\\AULA-1\\Documents
      \\(Engenharia de Software)\\Fase 3\\Linguagem Orientada a
      Objetos\\gerenciaBanco\\gerenciaBanco\\src\\gerenciabanco
      \\saracura.jpg");
84
85     cliente clientel = new cliente();//instância o cliente
86     JOptionPane.showMessageDialog(null, "Bem vindo ao Banco
      Saracura do Banhado\\n", "INÍCIO", JOptionPane.
      INFORMATION_MESSAGE, icon);//add custom icon
87     try {
88         clientel.nome = JOptionPane.showInputDialog(null, "
      Informe seu Nome.", "Nome");
89         if (clientel.nome == null){//caso o usuario cancele a
      opção
90             JOptionPane.showMessageDialog(null, "Você cancelou a
      operação");
91         }
92     }
93     catch (NullPointerException e){
94         JOptionPane.showMessageDialog(null, "Entre com um Nome
      válido.", "Erro" , JOptionPane.ERROR_MESSAGE);
95     }
96
97     try {
98         clientel.sobrenome = JOptionPane.showInputDialog(null, "
      Informe seu Sobrenome.", "Sobrenome");
99     }
100    catch(NullPointerException e){
101        JOptionPane.showMessageDialog(null, "Entre com um
      sobrenome válido.", "Erro" , JOptionPane.ERROR_MESSAGE)
      ;
102    }
103
104    try{
105        clientel.cpf = JOptionPane.showInputDialog(null, "Informe
      o número do CPF.", "000.000.000-00");
106    }
107    catch (NullPointerException e){
108        JOptionPane.showMessageDialog(null, "Entre com um CPF
      válido.", "Erro" , JOptionPane.ERROR_MESSAGE);

```

```

109     }
110     //cliente1.saldo = 445;//remover
111     while(true){ //InterruptedException
112
113         String opcao = JOptionPane.showInputDialog(null,"Opção 1
            - Consulta saldo\n Opção 2 - Realizar um deposito\n
            Opção 3 - Realizar um saque\n Opção 4 - Sair \n",4);//
            deixa a opçã4 4 como deful
114         //conversão de String para int
115         int control = 0;
116         if ("opcao" == null){
117
118             control = 0;
119         }
120         else {
121             control = Integer.parseInt(opcao);
122         }
123
124         //int control = Integer.parseInt(opcao);
125
126         if ("opcao" == null){//caso o usuario cancele a opção
127             JOptionPane.showMessageDialog(null, "Você cancelou a
                operação");
128             break;
129         }
130
131         if (control == 0){
132             JOptionPane.showMessageDialog(null, "Você cancelou a
                operação");
133             break;
134         }
135
136         if (control == 4){//para sair da operação
137             JOptionPane.showMessageDialog(null,"Volte sempre "+
                cliente1.nome+" "+cliente1.sobrenome+"\nTenha um
                bom dia!", "LOGOUT", JOptionPane.
                INFORMATION_MESSAGE, icon);//add custon icon
138             break;
139         }
140
141         switch (control){

```

```

142         case 0:
143             JOptionPane.showMessageDialog(null, "Você
144                 cancelou a operação");
145             break;
146
147         case 1://consulta de saldo     consultaPilas(nome)
148             JOptionPane.showMessageDialog(null,"Consulta
149                 saldo.", clientel.nome+" "+clientel.sobrenome,
150                 JOptionPane.INFORMATION_MESSAGE, icon);//add
151                 custom icon
152             clientel.consultaPilas(clientel.nome);
153             break;
154
155         case 2://realiza deposito
156             JOptionPane.showMessageDialog(null,"Realizar um
157                 deposito.", clientel.nome+" "+clientel.
158                 sobrenome, JOptionPane.INFORMATION_MESSAGE,
159                 icon);
160             int verifica = clientel.depositaPila(clientel.
161                 nome);
162             if (verifica == 1){
163                 JOptionPane.showMessageDialog(null,"Depósito
164                     realizado com sucesso.", clientel.nome+" "
165                     +clientel.sobrenome, JOptionPane.
166                     INFORMATION_MESSAGE, icon);
167             }else{
168                 JOptionPane.showMessageDialog(null,"Falha no
169                     deposito.\n Tente novamente.", clientel.
170                     nome+" "+clientel.sobrenome, JOptionPane.
171                     ERROR_MESSAGE, icon);
172             }
173             break;
174
175         case 3:// realiza saque
176             JOptionPane.showMessageDialog(null,"Realiza um
177                 saque.", clientel.nome+" "+clientel.sobrenome,
178                 JOptionPane.INFORMATION_MESSAGE, icon);
179             verifica = clientel.saquePila(clientel.nome);
180
181             if (verifica == 1){
182                 JOptionPane.showMessageDialog(null,"Saque

```

```

167         realizado com sucesso.", clientel.nome+" "
168         +clientel.sobrenome, JOptionPane.
            INFORMATION_MESSAGE, icon);
169     }else{
170         JOptionPane.showMessageDialog(null,"Falha no
171         saque.\n Tente novamente.", clientel.nome+
172         " "+clientel.sobrenome, JOptionPane.
            ERROR_MESSAGE, icon);
173     }
174     break;
175
176     default:
177         JOptionPane.showMessageDialog(null,"Opção
178         inválida.", clientel.nome+" "+clientel.
            sobrenome, JOptionPane.INFORMATION_MESSAGE,
            icon);
179     break;
180 }
181 }
182 }
183 }

```