PORTFÓLIO

LINGUAGEM ORIENTADA A OBJETOS

Relatório prático apresentando projeto em Linguagem Orientada a Objetos como requisito práticos para obtenção de média semestral na disciplina.

Indaiatuba / SP

2025

SUMÁRIO

1 - INTRODUÇÃO

Será apresentado a estrutura de uma aplicação estruturada de um gerenciamento bancário onde o usuário por sua vez consiga informar seu nome, sobrenome e CPF, utilizando a linguagem de programação Java, sendo abordado um desenvolvimento de software que enfatiza e organiza códigos, assim facilitando a criação e manutenção do programa.

Java e uma linguagem popular voltada para programação a objetos devido a sua portabilidade, robustez e ampla utilização no desenvolvimento de softwares comerciais, aplicativos móveis e sistemas corporativos.

2 – DESENVOLVIMENTO

A **Programação Orientada a Objetos (POO)** é um paradigma essencial no desenvolvimento de um paradigma essencial no desenvolvimento de software moderno, oferecendo modularidade, reutilização de código e facilidade de manutenção. Java, como uma linguagem fortemente baseada nesse paradigma, implementa o POO de forma rigorosa, garantindo encapsulamento, herança, polimorfismo e abstração. Vamos explorar mais adiante os principais conceitos práticos de POO em Java.

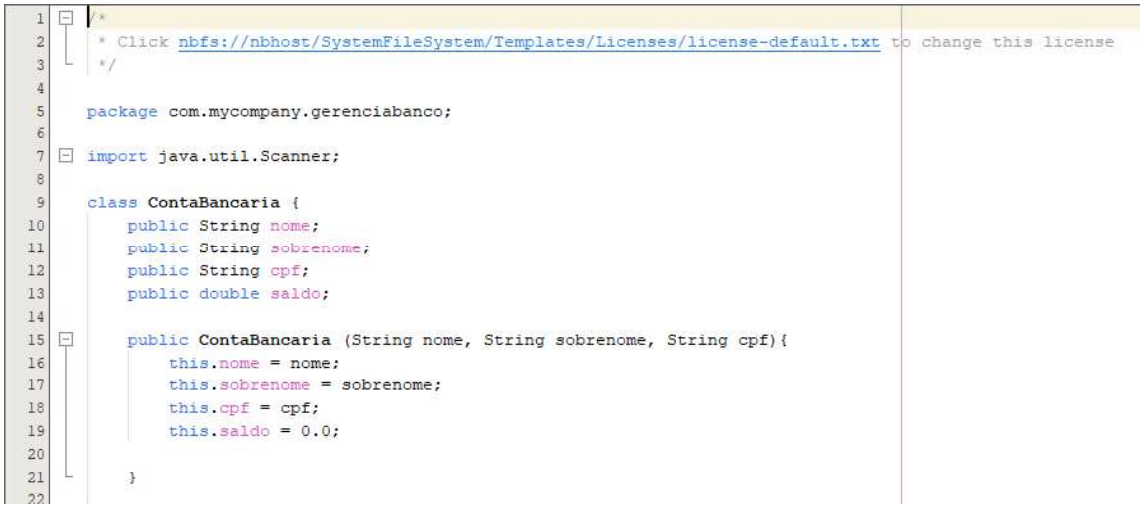
3 – ESTRUTURA E METODOS DE CODIGO

3. 1 - “CLASSES E METODOS”

.

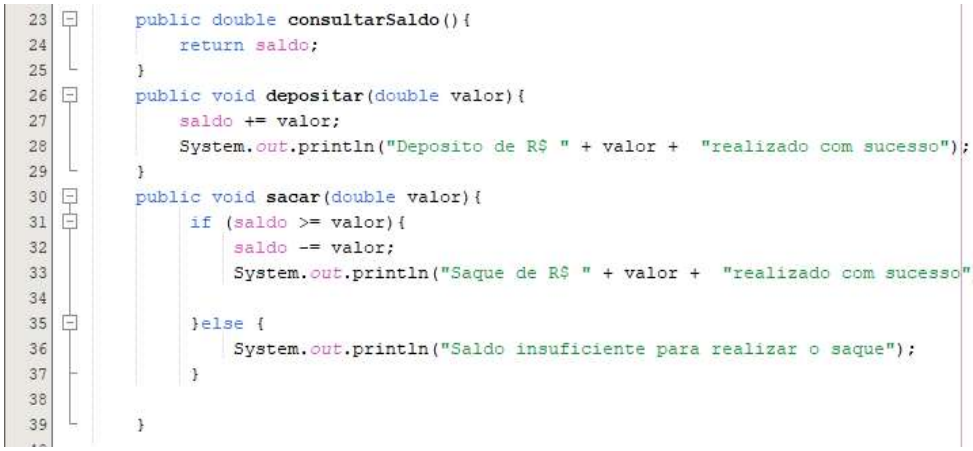
Na aula pratica, criamos um novo projeto, abrindo a pasta novo projeto Java With Maven e Java application com no nome GerenciaBanco.

Na estrutura iniciamos o código com atributos nome, sobrenome, CPF e saldo em zero usando assim código de acesso private com String que armazena caracter e double que armazena números decimais com alta precisão.



```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 */

package com.mycompany.gerenciabanco;

import java.util.Scanner;

class ContaBancaria {
    public String nome;
    public String sobrenome;
    public String cpf;
    public double saldo;

    public ContaBancaria (String nome, String sobrenome, String cpf){
        this.nome = nome;
        this.sobrenome = sobrenome;
        this.cpf = cpf;
        this.saldo = 0.0;

    }
```

Em seguida, criamos o código utilizando a estrutura de decisão if e else



```java
    public double consultarSaldo(){
        return saldo;
    }
    public void depositar(double valor){
        saldo += valor;
        System.out.println("Deposito de R$ " + valor +  "realizado com sucesso");
    }
    public void sacar(double valor){
         if (saldo >= valor){
             saldo -= valor;
             System.out.println("Saque de R$ " + valor +  "realizado com sucesso");

         }else {
             System.out.println("Saldo insuficiente para realizar o saque");
         }

    }
```

Utilizamos um método de repetição para informar um MENU onde descreve uma interação para o usuário informando a opção desejada sendo inicializado com o scanner para que possa ser digitado o que foi escolhido utilizando o scanner.nextLine() para receber o que foi digitado.

```
40
41      public void exibirMenuInfor(){
43          Scanner scanner = new Scanner(System.in);
44          int opcao;
45
46          do {
47              System.out.println("\n--------MENU-----");
48              System.out.println("1. Consultar Saldo");
49              System.out.println("2. Realizar o Deposito");
50              System.out.println("3. Realizar Saque");
51              System.out.println("4. Encerrar");
52              System.out.println("Qual opção que vc deseja escolher: ");
53              opcao = scanner.nextInt();
55              switch(opcao){
56                  case 1:
57                   System.out.println("Saldo: R$ " + consultarSaldo());
58                   break;
59                   case 2:
60                   System.out.println("Digite o valor do deposito:");
61                   double valorDeposito = scanner.nextDouble();
62                   depositar(valorDeposito);
63                   break;
64                   case 3:
65                   System.out.println("Digite o valor do saque");
66                   double ValorSaque = scanner.nextDouble();
67                   sacar(ValorSaque);
68                   break;
69                   case 4:
70                   System.out.println("Encerrando....");
71                   break;
72                   default:
73                       System.out.println("Opção invalida");
74
75
76
77              }
78          } while (opcao!=4);
79
80          scanner.close();
      }
```

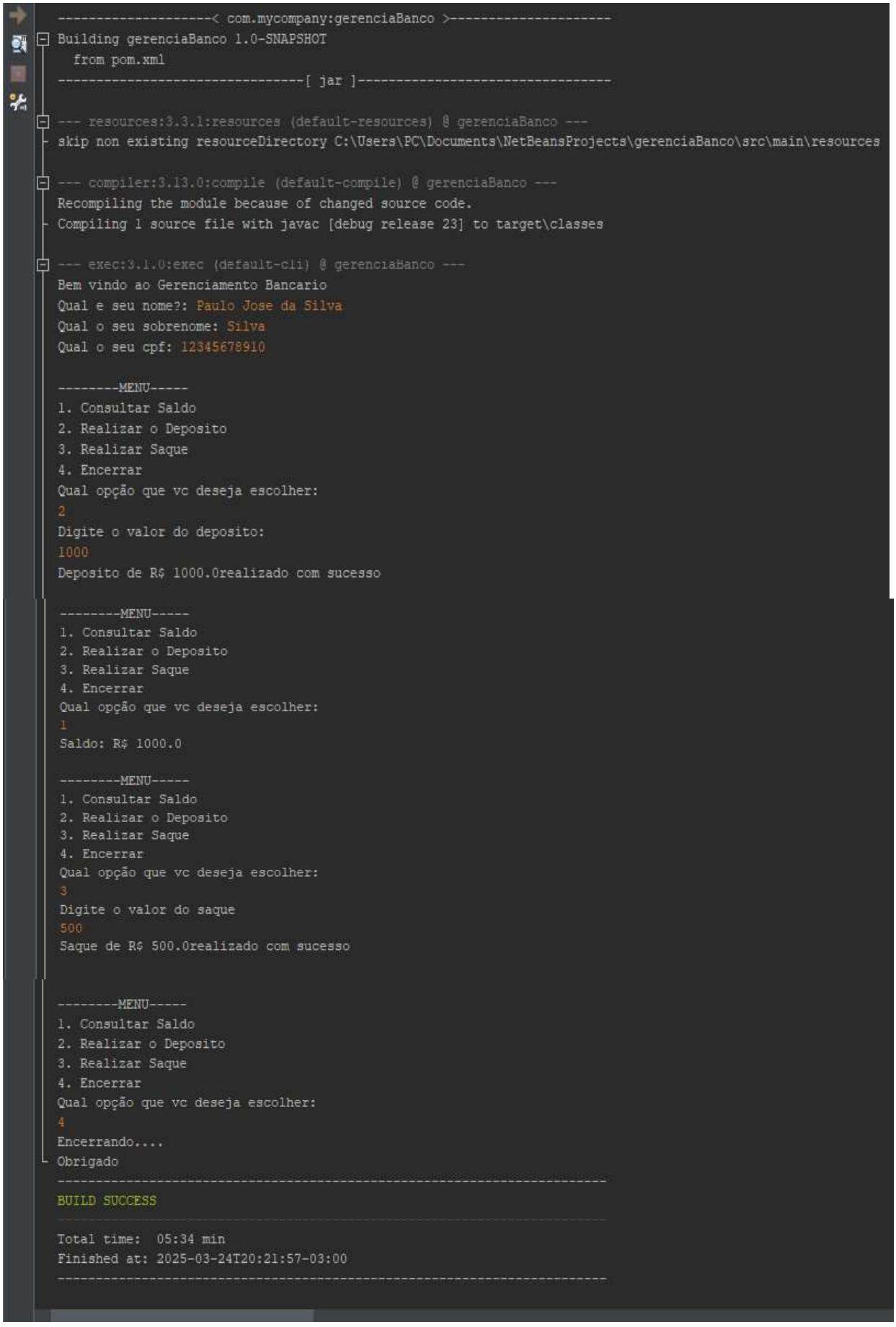
Utilizamos também o método principal main (Programa Principal) dado como o nome da classe GerenciaBanco, onde o cliente vai interagir com o sistema informando o seu nome, sobrenome e CPF. O método main em Java tem a responsabilidade e disponibiliza o ponto de entrada de uma aplicação, e responsável por iniciar a execução do programa. No contexto de um sistema de gerenciamento bancário, o método main pode ser utilizado para instanciar classes, gerenciar as transações bancárias e integrar ao usuário a utilização necessária para a sua movimentação bancária, através das classes de aplicação o método main permite a entrada do usuário que executa as operações e exibe os resultados esperados de maneira organizada. Com isso o código desenvolvido se torna modular, reutilizável e escalável para futuros aprimoramentos.

```
81
82  }
83
84  public class GerenciaBanco {
85
86      public static void main(String[] args) {
87          Scanner scanner = new Scanner(System.in);
88
89          System.out.println("Bem vindo ao Gerenciamento Bancario");
90          System.out.print("Qual e seu nome?: ");
91          String nome = scanner.nextLine();
92          System.out.print("Qual o seu sobrenome: ");
93          String sobrenome = scanner.nextLine();
94          System.out.print("Qual o seu cpf: ");
95          String cpf = scanner.nextLine();
96
97
98          ContaBancaria conta = new ContaBancaria (nome, sobrenome, cpf);
99
100         conta.exibirMenuInfor();
101
102         System.out.println("Obrigado");
103
104         scanner.close();
105
106     }
107 }
108
```

4 - RESULTADO

No resultado do sistema obtivemos a aplicação realizada com sucesso onde foi proposta e o código rodou de forma esperada. Segue abaixo os testes de forma objetiva.

Concluímos de uma forma clara que é possível realizar projetos de desenvolvimento com a linguagem Java com ótima precisão. E possível visualizar na estrutura desenvolvida que as funções e os métodos utilizados são muito importantes em todas as etapas da aplicação do código, observamos também de como algumas ferramentas e processos de um projeto devem funcionar para a obtenção do resultado esperado.



```
--------------------< com.mycompany:gerenciaBanco >---------------------
Building gerenciaBanco 1.0-SNAPSHOT
  from pom.xml
--------------------------------[ jar ]---------------------------------

--- resources:3.3.1:resources (default-resources) @ gerenciaBanco ---
skip non existing resourceDirectory C:\Users\PC\Documents\NetBeansProjects\gerenciaBanco\src\main\resources

--- compiler:3.13.0:compile (default-compile) @ gerenciaBanco ---
Recompiling the module because of changed source code.
Compiling 1 source file with javac [debug release 23] to target\classes

--- exec:3.1.0:exec (default-cli) @ gerenciaBanco ---
Bem vindo ao Gerenciamento Bancario
Qual e seu nome?: Paulo Jose da Silva
Qual o seu sobrenome: Silva
Qual o seu cpf: 12345678910

--------MENU-----
1. Consultar Saldo
2. Realizar o Deposito
3. Realizar Saque
4. Encerrar
Qual opção que vc deseja escolher:
2
Digite o valor do deposito:
1000
Deposito de R$ 1000.0realizado com sucesso

--------MENU-----
1. Consultar Saldo
2. Realizar o Deposito
3. Realizar Saque
4. Encerrar
Qual opção que vc deseja escolher:
1
Saldo: R$ 1000.0

--------MENU-----
1. Consultar Saldo
2. Realizar o Deposito
3. Realizar Saque
4. Encerrar
Qual opção que vc deseja escolher:
3
Digite o valor do saque
500
Saque de R$ 500.0realizado com sucesso


--------MENU-----
1. Consultar Saldo
2. Realizar o Deposito
3. Realizar Saque
4. Encerrar
Qual opção que vc deseja escolher:
4
Encerrando....
Obrigado
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  05:34 min
Finished at: 2025-03-24T20:21:57-03:00
------------------------------------------------------------------------
```

5 – CONCLUSÃO

Em toda a execução da aula prática, foram abordados os conceitos fundamentais da linguagem de programação Java, sendo ela Orientada a Objetos (POO), como atributos, operadores, estruturas de controle de fluxo.

Apresentamos o ambiente de desenvolvimento integrado NetBeans IDE, uma plataforma de compilação bastante conhecida que facilita a criação, o desenvolvimento, edição e depuração de códigos Java, bem como a integração com o gerenciador de dependências Maven.

Por meio dos exemplos práticos desenvolvidos foram explorados os conceitos de herança, polimorfismo e encapsulamento, onde foi demostrado as características aplicadas no projeto que contribui para o desenvolvimento de uma aplicação de códigos robustos e de fácil entendimento. Abordamos ao longo do projeto a criação de classes, métodos e estruturas de decisão, em todo o desenvolvimento aplicado nos permitiu a pratica e o conhecimento de forma bem objetiva assim consolidando o aprendizado de forma concreta.

Conclui-se que Java e uma linguagem de programação bem popular e utilizada em diversos setores industriais de software, desde a sua criação de aplicações desktop até o desenvolvimento de sistemas web e mobile.

5 – REFERÊNCIAS

LOIANE GRONER. Curso de Java 67: criando threads + métodos start, run e sleep. 2016.

LOIANE GRONER. Curso de Java 68: threads: interface runnable. 2016.

LOIANE GRONER. Curso de Java 69: criando vários threads + métodos isAlive e join.

GRONER, L. Canal Loiane Groner. Curso de Java #81.

GRONER, L. Canal Loiane Groner. Curso de Java #82.

GRONER, L. Canal Loiane Groner. Curso de Java #80.

https://www.cursoemvideo.com/curso/java-poo/