


| | | | |
|---|--|----------------|------------------|
|  | UNESP – CENTRO UNIVERSITÁRIO | | |
| | CURSO: Sistemas para Internet / Sistemas de Informação | | TURNO: Noite |
| | UNIDADE CURRICULAR: Linguagem de Programação I | | |
| | PERÍODO: 2 | C.H.: 60h/aula | SEMESTRE: 2023.1 |
| | PROFESSOR: MESSIAS RAFAEL BATISTA | | |
| EQUIPE | ALUNO 1 : LUIZ GONZAGA GONDIM RIBEIRO NETO | | |
| | ALUNO 2 : GONÇALO GABRIEL DE HOLANDA SANTOS | | |
| | ALUNO 3 : DANILO MONTEIRO DE MEDEIROS | | |
| | ALUNO 4 : SERGIO FERNANDES TAVARES | | |
| | ALUNO 5 : EWERTON MONTEIRO | | |

Documentação do Projeto: Banco Virtual

1. Introdução

O Banco Virtual é um sistema desenvolvido em Java que simula as operações de um banco, permitindo a criação e gerenciamento de contas correntes, contas poupança e contas de investimentos. O objetivo do projeto é fornecer uma plataforma onde os clientes possam realizar transações bancárias básicas, como depósitos e saques, além de calcular rendimentos específicos para contas poupança e contas de investimentos.

2. Tecnologias Utilizadas

Linguagem de Programação: Java

IDE: IntelliJ IDEA

O Java foi escolhido como a linguagem de programação principal devido à sua portabilidade, segurança e ampla adoção na indústria. A IDE IntelliJ IDEA foi utilizada para desenvolver, depurar e executar o código do projeto, oferecendo recursos avançados de desenvolvimento.

3. Estrutura do Projeto

O projeto foi organizado em várias classes que representam as entidades e funcionalidades do sistema. As principais classes são:

Classe **Conta**: Representa uma conta bancária e contém atributos como número da conta, nome do titular, saldo e CPF. Possui métodos para realizar depósitos e saques, bem como para calcular rendimentos específicos para contas poupança e contas de investimentos.

Classe **ContaCorrente**: Derivada da classe **Conta**, representa uma conta corrente específica com características adicionais, como limite de cheque especial.

Classe **ContaPoupanca**: Derivada da classe **Conta**, representa uma conta poupança específica que recebe um acréscimo de 2% a cada chamada do método **rendimentoPoupanca**.

Classe **ContaInvestimento**: Derivada da classe **Conta**, representa uma conta de investimento específica que recebe um acréscimo de 10% a cada chamada do método **rendimentoInvestimento**.

Classe **App**: Classe principal que contém o método **main** e implementa o menu interativo para acessar as funcionalidades do banco.

4. Funcionalidades

O Banco Virtual oferece as seguintes funcionalidades:

Criação de novas contas correntes, contas poupança e contas de investimentos.
Acesso a contas existentes através do número da conta.

Realização de depósitos em contas.

Realização de saques em contas, considerando o saldo disponível.

Cálculo de rendimento de contas poupança e contas de investimentos.

5. Recursos Utilizados

5.1. Encapsulamento

Os atributos das classes foram definidos como privados (**private**) e acessados por meio de métodos getters e setters públicos (**get** e **set**), seguindo o princípio de encapsulamento. Isso permite controlar o acesso aos atributos e manter a integridade dos dados.

5.2. Composição

A classe **Conta** utiliza a composição para criar objetos de outras classes, como **ContaCorrente**, **ContaPoupanca** e **ContaInvestimento**. Essa abordagem permite a reutilização de código e a composição de funcionalidades específicas nas classes derivadas.

5.3. Herança

As classes **ContaCorrente**, **ContaPoupanca** e **ContaInvestimento** herdam da classe base **Conta**, aproveitando os atributos e métodos comuns. A herança possibilita a especialização das classes e a criação de comportamentos específicos para cada tipo de conta.

5.4. Polimorfismo

O polimorfismo é aplicado na manipulação de objetos do tipo **Conta** em conjunto com as classes derivadas (**ContaCorrente**, **ContaPoupanca** e **ContaInvestimento**). Isso permite tratar objetos de diferentes classes como objetos de um tipo comum, facilitando a execução de métodos específicos de cada classe através de uma referência genérica.

5.5. Classes e Métodos Abstratos

A classe abstrata **Conta** possui métodos abstratos, como **rendimentoPoupanca** e **rendimentoInvestimento**, que são implementados nas classes derivadas. Isso permite definir comportamentos comuns nas classes abstratas e implementações específicas nas subclasses.

5.6. Métodos Construtores

Foram definidos métodos construtores nas classes **Conta**, **ContaCorrente**, **ContaPoupanca** e **ContaInvestimento** para inicializar os atributos das instâncias. Os construtores são responsáveis por garantir que as contas sejam criadas corretamente com seus respectivos valores.

6. Conclusão

O projeto do Banco Virtual foi desenvolvido utilizando a linguagem de programação Java e a IDE IntelliJ IDEA. Foram aplicados princípios de orientação a objetos, como encapsulamento, composição, herança, polimorfismo, classes e métodos abstratos, e métodos construtores. O resultado é um sistema que permite criar e gerenciar diferentes tipos de contas bancárias, realizar transações e calcular rendimentos. O uso das tecnologias e recursos mencionados proporcionou uma estrutura organizada, flexível e de fácil manutenção.