

PROJETO INTEGRADOR III-A

Projeto: Gerenciamento de Contas Bancárias

Este projeto envolve a criação de uma aplicação básica de gerenciamento de contas bancárias.

Você pode criar uma aplicação web para o gerenciamento de contas bancárias utilizando Python e Django. Django é um framework web popular para desenvolvimento rápido de aplicações web em Python.

Classes:

Aqui está uma abordagem geral de como você pode estruturar sua aplicação web com Django para o gerenciamento de contas bancárias:

ContaBancaria: Defina seus modelos: No arquivo `models.py`, defina os modelos `ContaBancaria`, `ContaCorrente` e `ContaPoupanca` com os atributos e métodos necessários. Certifique-se de usar encapsulamento adequado para proteger os atributos das classes, conforme mencionado nas instruções.

Atributos: número da conta, saldo.

Métodos: `depositar(valor)`, `sacar(valor)`, `obterSaldo()`.

ContaCorrente:

Crie suas views: No diretório `views.py`, crie as views para lidar com as operações de depósito, saque, cálculo de juros etc. Aqui você irá definir as funções que serão chamadas quando determinadas URLs forem acessadas.

Herda de `ContaBancaria`.

Defina suas URLs: No arquivo `urls.py`, defina as URLs que correspondem às views que você criou anteriormente.

Atributo adicional: taxa de manutenção.

Crie seus templates HTML: No diretório `templates`, crie os arquivos HTML que serão usados para renderizar as páginas da sua aplicação. Por exemplo, você pode ter um template para exibir o saldo da conta, outro para realizar um depósito etc.

Método: `aplicarTaxaManutencao()`.

Crie seus formulários: No diretório `forms.py`, crie os formulários Django que serão usados para coletar informações dos usuários, como o valor do depósito ou saque.

ContaPoupanca:

Configure suas URLs: No arquivo `urls.py` do seu projeto Django, configure as URLs para apontar para as views corretas.

Herda de `ContaBancaria`.

Implemente a lógica de negócios: No arquivo `views.py`, implemente a lógica de negócios para realizar as operações desejadas, como depósito, saque, cálculo de juros etc.

Atributo adicional: taxa de juros.

Método: `calcularJuros()`.

Teste sua aplicação: Crie um programa principal (Main) que cria objetos das classes de modelo e realiza operações como depósito, saque, cálculo de juros poupança.

Instruções:

1. Crie a classe `ContaBancaria` com os atributos e métodos mencionados.
2. Crie as classes `ContaCorrente` e `ContaPoupanca` que herdam de `ContaBancaria`.
3. Implemente os métodos específicos para cada tipo de conta.
4. Teste as funcionalidades em um programa principal (Main) que cria objetos das classes e realiza operações como depósito, saque, cálculo de juros poupança.
5. Certifique-se de usar encapsulamento adequado para proteger os atributos das classes.

Critérios de Avaliação:

1- Correção Funcional – Valor: 2,0 pontos.

- As classes foram corretamente implementadas, seguindo as especificações fornecidas.
- As operações de depósito, saque, aplicação de taxas, etc., são realizadas corretamente para cada tipo de conta.

2- Herança e Encapsulamento – Valor: 2,0 pontos.

- A herança está sendo usada de maneira adequada para compartilhar características comuns entre as classes.
- O encapsulamento está implementado corretamente, protegendo os atributos e promovendo o uso de métodos para interações.

3- Clareza e Estrutura do Código – Valor: 2,0 pontos.

- O código é claro, bem estruturado e segue as melhores práticas de programação em Java.
- Nomes de classes, métodos e variáveis são descritivos e seguem as convenções de nomenclatura.

4- Testes e Demonstrações – Valor: 2,0 pontos.

- O programa principal (Main) contém testes abrangentes para demonstrar o funcionamento do sistema.
- Os testes incluem diferentes cenários, como depósitos, saques e aplicação de taxas.

5- Comentários e Documentação – Valor: 2,0 pontos.

- Existem comentários adequados no código, explicando partes complexas ou fornecendo contexto adicional.
- O código está adequadamente documentado, destacando o propósito das classes e métodos.

Forma de envio: Envie o vídeo em MP4 ou link do vídeo apresentando as telas do seu projeto e execução do programa.

Valor: 10 pontos.

Observação: O trabalho poderá ser realizado em dupla.

Bom trabalho!