

POO - Lista de Exercícios 2

Instruções:

- Para praticar, resolva os exercícios em Java e em Python.
- Em Java, os atributos das classes devem ser privados ou protegidos. Em Python, isso é opcional.
- Crie um pacote/diretório listaex2. Implemente cada exercício em um subpacote exZ, onde Z é o número do exercício. Ou seja, o primeiro exercício é implementado no pacote ex1, o segundo no pacote ex2 e assim por diante. Todos dentro do pacote listaex2.

1. Implemente as classes abaixo incluindo dois métodos construtores e os métodos getters e setters para os atributos:
 - a. Uma classe Funcionario com nome e email como atributos. Implemente o método exibeDados que imprime os dados do funcionário;
 - b. Crie a classe Assistente, que também é um funcionário, e que possui um número de matrícula (faça o método GET). Sobrescreva o método exibeDados();
 - c. Sabendo que os Assistentes Técnicos possuem um bônus salarial e que os Assistentes Administrativos possuem um turno (dia ou noite) e um adicional noturno, crie as classes Tecnico e Administrativo. Sobrescreva o método exibeDados() dessas classes;
 - d. Crie uma outra classe com o método main. Nesse método, crie um objeto de cada classe, preencha os atributos e chame o método exibeDados de cada um.
2. Elabore uma classe abstrata ContaBancaria, com os seguintes membros:
 - a. atributos String cliente, int numConta e float saldo;
 - b. método sacar, que recebe o valor do saque e atualiza o saldo e retorna verdadeiro (true), se foi possível sacar, ou não faz nada e retorna falso (false), caso não seja possível. O saque não pode ser realizado se o valor for maior do que o saldo;
 - c. método depositar, que rebe o valor do depósito e não tem retorno;
 - d. método abstrato getSaldo, que não tem entrada e retornaria o saldo atual.

Agora acrescente duas classes filhas de ContaBancaria: ContaPoupança e ContaEspecial, com as seguintes características a mais:

Classe ContaPoupança:

- i. atributo float taxa de rendimento;
- ii. implementação do método saldo, que utiliza o valor atual do saldo e a taxa de rendimento para atualizar o saldo e retornar o valor resultante.

Classe ContaEspecial

- i. atributo float limite;
- ii. redefinição do método sacar, permitindo saldo negativo até o valor do limite.
- iii. implementação do método saldo, que retorna o saldo atual.

Implemente a classe Contas, contendo o método main. Nesta classe, você deverá implementar:

- a. Incluir dados relativos a(s) conta(s) de um cliente;
 - b. Sacar um determinado valor da(s) sua(s) conta(s);
 - c. Depositar um determinado valor na(s) sua(s) conta(s);
 - d. Mostrar o novo saldo do cliente, a partir da taxa de rendimento, daqueles que possuem conta poupança;
 - e. Mostrar os dados da(s) conta(s) de um cliente;
3. Crie uma classe abstrata ou interface chamada Geometria que não possui atributos, mas possui os métodos abstratos área e comprimento, que retornam um double. Crie a classe Circulo que herda de Geometria possui um atributo raio. Crie uma classe Retangulo que herda de Geometria e possui dois atributos que representam os lados do retângulo. Crie uma classe com método main que cria um vetor que armazena círculos e retângulos e imprime as áreas e comprimentos deles.