

POO - Lista de Exercícios - Classes

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 listaex01. Implemente cada exercício em um subpacote exZ, onde Z é o número do exercício. Ou seja, o primeiro exercício é implementado no subpacote ex1, o segundo no subpacote ex2 e assim por diante. Todos dentro do pacote listaex1.
-
1. Um número complexo é um número z que pode ser escrito na forma $z = a + bi$, sendo a e b números reais e i denota a unidade imaginária, em que $i^2 = -1$. Crie a classe NumeroComplexo com dois atributos, a e b , do tipo double. Crie o método soma que recebe um número complexo e retorna outro número complexo que é a soma do objeto que chamou o método com o que foi recebido. Crie o método produto que recebe um número complexo e retorna outro número complexo que é o produto do objeto que chamou o método com o que foi recebido. Crie o método conjugado que não possui entrada e retorna outro número complexo que é o conjugado do objeto que chamou o método. Crie o método toString que retorna um String com o número complexo no formato $a + bi$. Crie uma classe com método main que solicita ao usuário que digite dois números complexos e depois realiza todas as operações citadas, imprimindo o resultado. Fórmulas (z e w são números complexos):
Soma: $z + w = w + z = (a + bi) + (c + di) = (a + c) + (b + d)i$
Produto: $zw = wz = (a + bi)(c + di) = (ac - bd) + (bc + ad)i$
Conjugado: $\bar{z} = a - bi$, onde \bar{z} denota o conjugado de z .
 2. Criar uma classe Conta, que possua um atributo número e o atributo saldo e os métodos para depositar, sacar conferir o saldo. Crie uma classe ContaInterface que possui apenas um método que recebe uma conta e mostra um menu para o usuário em que ele pode escolher entre sair do menu, depositar um valor, sacar um valor e conferir um saldo. Deve ser possível selecionar essas opções diversas vezes até o usuário escolher sair. Crie uma classe UsaConta com um método main que cria uma conta e chame o menu de opções.
 3. Crie uma classe Banco que contém um atributo que é um vetor do tipo Conta (Exercício 2). Crie um método que recebe uma conta e a adiciona no vetor. Crie uma classe BancoInterface que possui como atributo um objeto do tipo Banco e um construtor com argumento. Inclua também na classe BancoInterface um método com um menu que permite abrir uma nova conta, selecionar uma conta do banco ou sair do programa. Ao selecionar uma conta, invoque o método da classe ContaInterface do Exercício 2 para manipulá-la. Deve ser possível manipular diversas contas do banco até que o usuário selecione sair do programa.
 4. Imagine um programa para uma faculdade particular. Crie uma classe Aluno com registro e nome. Crie uma classe Disciplina com código, nome e valor. Crie uma classe Matricula com número, data, aluno e um vetor de disciplinas. Adicione em todas as classes o método toString, que retorna uma String com os dados da classe. Adicione na classe Matricula um método calculaTotal que retorna a soma dos valores de todas as disciplinas inseridas. Esse total deve aparecer no método toString de Matricula. Crie uma classe Main cujos atributos são vetores do tipo Aluno, Disciplina e Matricula. Crie um método para cadastrar um Aluno no vetor de alunos. Crie um método que recebe um registro de aluno e retorna o aluno no vetor que possui o número correspondente ou null se não existir. Crie também métodos equivalentes para Disciplina e Matricula. Crie um método main com um menu que permite ao usuário cadastrar diversos alunos, disciplinas e matrículas e visualizar os dados de uma matrícula a partir do número dela. O programa só encerra quando o usuário escolhe sair do programa.