

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE - UFRN

Instituto Metrópole Digital
IMD0040 - Linguagem de Programação 2
Atividade - Herança e Polimorfismo
Professor: Emerson Alencar

Exercício 01:

Crie duas classes: Ponto2D e Ponto3D. Ponto2D possui como atributos as coordenadas x e y, enquanto Ponto3D, além delas, também possui a coordenada z. Utilize a relação de herança para representar estas classes.

A respeito dos construtores, Ponto2D deve ter apenas um construtor, que recebe os valores de x e y como parâmetros (tipo double). Já Ponto3D também deve ter apenas um construtor, que deve receber x, y e z como parâmetros (também do tipo double).

Dica: Se a relação de herança e a declaração dos construtores foram feitas corretamente, você deverá, obrigatoriamente, chamar o construtor da superclasse explicitamente.

Ambas as classes devem sobrescrever o método toString(), que é originalmente declarado na classe Object. Este método deve retornar uma representação do objeto em forma de String, indicando qual o valor de cada coordenada. É importante que Ponto3D tire proveito do método toString() de Ponto2D para mostrar os valores das coordenadas x e y.

Crie uma Classe Principal que cria 1 objeto de cada classe utilizando os construtores e escreve seus valores na tela passando para o System.out.println() o objeto criado.

Exercício 02:

Crie a classe Figura que representa figuras geométricas, representadas pelas classes Quadrado e Retângulo. Uma figura pode ter sua área calculada a partir do método calcularArea(), que retorna a área calculada da figura em forma de um double.

Crie também a classe FiguraComplexa. Uma figura complexa é também uma figura, mas a diferença é que ela é composta por várias figuras (quadrados, retângulos ou até outras figuras complexas). Para calcular a área de uma figura complexa, basta somar a área de todas as figuras que a compõem.

Para executar a aplicação, crie a classe Calculador, que é responsável por criar uma figura complexa e calcular a sua área. Esta figura deve ser composta por:

- 1 quadrado com 3 de lado
- 1 quadrado com 10 de lado
- 1 retângulo com lados 2 e 7
- 1 retângulo com lados 5 e 3

Dica: Perceba a diferença entre uma classe ser uma figura e ter uma ou mais figuras.

Insira todas as figuras criadas num objeto da class `FiguraComplexa`, utilizando o construtor. Chame o método `calculaArea` da `FiguraComplexa` e exiba seu valor no terminal.

Exercício 3:

Crie uma classe `Veiculo` com um atributo `ligado` (privado), que indica se o carro está ligado ou não. Esta classe deve ter também os métodos `ligar()` e `desligar()`, que definem o valor para este atributo, e um método getter (`isLigado()`).

Depois crie três subclasses de `Veiculo`: `Automovel`, `Motocicleta` e `Onibus`. Cada classe destas deve sobrescrever os métodos `ligar()` e `desligar()` e deve imprimir mensagens como “Automóvel ligado”, “Motocicleta desligada”, etc. Para manter a consistência do modelo, descubra como fazer para que o atributo `ligado` de `Veiculo` tenha o valor correto quando os métodos são chamados.

Crie uma aplicação que instancia três veículos, um de cada tipo, e chama os métodos `ligar()`, `desligar()` e `isLigado()`. O resultado obtido deve ser consistente com o que o modelo representa. Por exemplo, ao chamar o método `ligar()` de um `Automovel`, é esperado que o método `isLigado()` retorne `true`.