

Ifes Campus Serra

BSI – Bacharelado de Sistemas de Informação

Programação Orientada a Objetos 2022-2

Lista de Exercícios: Classes e Objetos

1. Construa a classe Java Ponto2D para representar um ponto no plano. Defina as coordenadas x e y como atributos, e os seguintes métodos: constructor, getX(): int, getY(): int, distancia(:Ponto2D): float, igual(:Ponto2D): boolean.

Os métodos representam as respectivas funcionalidades: i) definir o estado inicial do objeto; ii) retornar a coordenada x; iii) retornar a coordenada y; iv) calcular e retornar a distância do objeto Ponto2D a outro objeto Ponto2D e v) testar se um objeto Ponto2D é igual a outro objeto Ponto2D.

Além disso, implemente o método toString() que constrói e retorna uma string com o estado de um objeto Ponto2D.

Construa uma aplicação (classe) TestaPonto2D que testa a implementação e a interface da classe.

2. Construa a classe Java Eq2g para representar uma equação do segundo grau. Defina os coeficientes como o mínimo em atributos. Defina também a seguinte interface: i) constructor; ii) métodos getXX para os coeficientes; iii) quantRaizes(): int; iv) raizA(): float; v) raizB(): float; getVertice(): Ponto2D.

Os métodos representam as respectivas funcionalidades: i) definir o estado inicial do objeto; ii) retornar o valor dos coeficientes da equação; iii) retornar a quantidade de raízes reais; iv) calcular e retornar a primeira raiz real; v) calcular e retornar a segunda raiz real.

Além disso, implemente o método toString(), que constrói e retorna uma string com o estado de um objeto tipo Eq2g.

Construa uma aplicação (classe) TestaEq2g que testa a implementação e a interface da classe.

Construa a aplicação ApLucro que resolve o seguinte problema:

(Enem 2013 – PPL) Uma pequena fábrica vende seus bonés em pacotes com quantidades de unidades variáveis. O lucro obtido é dado pela expressão $L(x) = -x^2 + 12x - 20$, onde x representa a quantidade de bonés contidos no pacote. A empresa pretende fazer um único tipo de empacotamento, obtendo um lucro máximo.

Qual deve ser a quantidade de bonés a fim de se obter o lucro máximo nas vendas?

3. Construa uma classe chamada Retangulo a partir da classe Ponto2D sabendo-se que para especificar um retângulo são necessários apenas 2 vértices: o vértice esquerdo superior e o vértice direito inferior.

A partir desta informação, construa a seguinte interface: i) um constructor; ii) `getVertice1():Ponto2D`; iii) `getVertice2():Ponto2D`; iv) `getVertice3():Ponto2D`; v) `getVertice4():Ponto2D`; vi) `area():float`; vii) `perimetro():float`; viii) `pertence(:Ponto2D):boolean`.

Os métodos representam as respectivas funcionalidades: definir o estado inicial do objeto, retornar os valores (em objetos Ponto2D) dos vértices do retângulo, calcular e retornar os valores da área e perímetro do retângulo, retornar true ou false em resposta a se um objeto Ponto2D passado como parâmetro está ou não posicionado dentro do retângulo.

Além disso, implemente o método `toString()`, que constrói e retorna uma string com o estado de um objeto tipo Retangulo.

Construa uma aplicação (classe) TestaRetangulo que testa a implementação e a interface da classe.

4. Construa uma classe Java para dar suporte à seguinte narrativa:

Um automóvel de passeio é capaz de informar o seu comprimento, altura e peso, sua capacidade máxima de carga (em Kg), quantos passageiros pode levar (máximo) e quantos passageiros existem correntemente no veículo. Além disso, todo automóvel de passeio possui um tanque de combustível com uma certa capacidade máxima em litros. O automóvel pode informar, a qualquer instante, quantos litros de combustível existem no tanque. O automóvel também é capaz de dizer sua velocidade atual e se está ligado ou desligado.

Sobre o comportamento, um automóvel de passeio é capaz de acelerar para uma dada velocidade (somente se o autmóvel estiver ligado). Se a velocidade é menor do que a velocidade atual então o efeito é o de uma frenagem. Sabendo que 1 litro de combustível pesa 1 Kg, e que uma pessoa padrão pesa aproximadamente 80 Kg, o automóvel é capaz de emitir uma alerta se está ou não com sobre-peso.

Um automóvel também é capaz de abastecer o seu tanque com uma quantidade dada de combustível. A quantidade adicionada de combustível mais a quantidade corrente no tanque não deve ultrapassar a capacidade máxima do tanque. Quando isso acontece, emitir um alerta na tela.

Um automovel é capaz de embarcar e desembarcar passageiros. No caso de embarque, a quantidade total de passageiros não deve exceder a capacidade máxima do automóvel. Quando isso acontece, emitir um alerta na tela. No caso do desembarque, a velocidade do automóvel deve ser zero. Caso contrário, um alerta deve ser emitido na tela.

O automóvel, se estiver ligado, é capaz de responder se deve reabastecer ou não (alerta de abastecimento). O reabastecimento é solicitado quando o tanque estiver com uma quantidade de combustível de cerca de 25% da capacidade máxima do tanque.

O automóvel também atende ao comando “parar” reduzindo a sua velocidade corrente para zero.

Construa uma aplicação TestaAutomovel para testar os métodos da interface da classe Automovel . TestaAutomovel deve simular situações de estado a fim de testar a consistência da interface da classe.

Fim (por enquanto)