UNIVERSIDADE FEDERAL
DO RIO DE JANEIRO

Centro de Ciências Matemáticas e da Natureza
Instituto de Computação
Lista 1 - Computação II

Professor: Giomar Sequeiros
Período: 2022 – II

Instruções:
- Cada classe com seu respectivo teste deve ser criado em um arquivo separado, nomeado adequadamente (q1.py)
- O **código** deve estar devidamente **indentado** e **comentado** indicando os tipos de entrada e saída.

Classes

Q1. Crie uma classe **Veiculo** com as seguintes características:

a) Um veículo tem um certo **consumo** de combustível (medido em km/litro), uma certa quantidade de **combustível** no tanque e uma **capacidade** máxima de combustível.
b) O consumo e a capacidade do tanque são especificados no **construtor** da classe.
c) Crie um método **mover** que receba a distância em quilômetros e reduz o nível de combustível no tanque baseado no consumo do veículo. Não permitir que o veículo se mova se estiver sem combustível.
d) Crie um método **getCombustivel**(), que retorne o nível atual de combustível.
e) Crie um método **abastecer**, para abastecer o tanque (sem ultrapassar a capacidade máxima do tanque).
f) Crie uma **função main** (fora da definição da classe) para testar a sua classe com os diferentes métodos e situações: Crie um veículo, abastece o tanque, ande, mande mostrar a quantidade de combustível e teste as validações que você criou para não permitir situações impossíveis

Q2. Crie uma classe **Funcionario** com atributos: cpf, nome, departamento, salário, data de entrada e um valor *booleano* indicando se o funcionário ainda está ativo na empresa ou se foi demitido.

a) Crie um construtor, você deve criar métodos para alterar/acessar atributos da classe.
b) Crie um método **bonificar** que aumenta o salário do funcionário ao adicionar um valor passado como parâmetro. Um funcionário inativo na empresa não pode ser bonificado
c) Crie um método **demitir**, que não recebe parâmetro algum, apenas modifica o valor *booleano* indicando que o funcionário não trabalha mais na empresa.
d) Crie uma **função main** (fora da definição da classe) para testar a sua classe com os diferentes métodos e situações:

Q3. Crie uma classe chamada **Triangulo** contendo os três lados como atributos (a, b e c) e os seguintes métodos:

a) Método **construtor** para inicializar o triângulo, desde que seja válido. Obs. Use o método do item b.
b) Método **eValido** que retorna verdadeiro se o triângulo é válido (o lado maior deve ser menor que a soma dos outros lados), senão retorna falso.
c) Método **tipoTriangulo** que retorna o tipo do triângulo. Se possuir os 3 lados iguais, é **equilátero**. Se possuir apenas 2 lados iguais, é **isósceles** e se possuir os 3 lados com valores diferentes é **escaleno**.
d) Método **calculaPerimetro** que retorna o perímetro do triângulo.

e) Método **calculaArea** que retorna a área do triângulo. A área do triângulo pode ser aproximada pela fórmula de Heron, conforme a Equação abaixo, onde *a*, *b* e *c* são os lados do triângulo e *p* é o seu semiperímetro (perímetro/2).

$$Area = \sqrt{p \times (p-a) \times (p-b) \times (p-c)}$$

f) Crie uma **função main** (fora da definição da classe) para testar a sua classe com os diferentes métodos e situações:

Q4. Escreva uma classe **Ponto2D** que represente um ponto no plano cartesiano. Além dos atributos por você identificados, a classe deve oferecer os seguintes métodos:

a) Construtores que permitam a inicialização do ponto:
   i. Por default (sem parâmetros) na origem do espaço 2D;
   ii. Num local indicado por dois parâmetros (indicando o valor de abcissa e ordenada do ponto que está sendo criado);

b) Método que permita calcular a distância euclidiana de um ponto com outro, definido pela expressão abaixo:

Sejam os pontos $P_1(x_1, y_1)$ e $P_2(x_2,y_2)$, a distância euclidiana está dado por:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

c) Crie uma **função main** (fora da definição da classe) para testar a sua classe com os diferentes métodos e situações: