



Engenharia de Sistemas e Computação

Lista 2 – Algoritmos computacionais

Professor: Giomar Sequeiros

Período: 2023 – I

Instruções:

- Criar uma pasta para cada questão (exemplo q1) contendo os arquivos fonte (extensão .py). É recomendável manter cada classe e o teste em arquivos separados (exemplo pessoa.py, cliente.py, teste.py)
- A primeira linha de cada arquivo deve conter o nome completo do aluno na forma de comentário.
Por exemplo: # Autor: João da Silva
- O código deve estar devidamente **comentado e documentado** indicando os tipos de entrada e saída.

Q1. Crie uma classe Veiculo com as seguintes características:

- 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.
- O consumo e a capacidade do tanque são especificados no **construtor** da classe.
- Crie um método mover que receba a distância em quilômetros e reduza o nível de combustível no tanque baseado no consumo do veículo. Não permitir que o veículo se mova se estiver sem combustível.
- Crie um método getCombustivel(), que retorne o nível atual de combustível.
- Crie um método abastecer, para abastecer o tanque (sem ultrapassar a capacidade máxima do tanque).
- 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 chamada **Triangulo** contendo os três lados como atributos (a, b e c) e os seguintes métodos:

- Método **construtor** para inicializar o triângulo, desde que seja válido. Obs. Use o método do item b.
- Método **ehValido** que retorna verdadeiro se o triângulo é válido (o lado maior deve ser menor que a soma dos outros lados), senão retorna falso.
- 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**.
- Método **calculaPerimetro** que retorna o perímetro do triângulo.
- 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).

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

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

Q3. Escreva uma classe **Ponto2D** que represente um ponto no plano cartesiano. Além dos atributos por você identificados, a classe deve oferecer os seguintes métodos:

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

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

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

Q4. Considere os seguintes objetos geométricos: retângulo, caixa, círculo e cilindro. Cada objeto geométrico deve possuir métodos para obter seu perímetro (figuras 2D), sua área (externa, no caso das 3D) e volume (figuras 3D). Crie um diagrama de classes que modele objetos geométricos de forma a aproveitar ao máximo suas características comuns. Depois implemente em python o modelo criado. As classes devem possuir construtores parametrizados, os quais devem ser utilizados para inicialização dos atributos dos objetos. Considere as informações abaixo para o cálculo do perímetro, área e volume dos objetos. Finalmente crie um programa de teste que instancie objetos das classes e invoque as funcionalidades implementadas.

- Retângulo:
 - Área = base*altura,
 - Perímetro = 2*base + 2*altura
- Caixa:
 - Volume = base1*base2*altura,
 - Área = 2*(base1*base2 + base1*altura + base2*altura)
- Círculo:
 - Área = $3.14*(raio)^2$,
 - Perímetro = $2*3.14*raio$
- Cilindro:
 - Volume = $3.14*(raio)^2*altura$,
 - Área = $2*3.14*(raio)^2 + 2*3.14*raio*altura$

Q5. Crie uma classe Funcionario com os atributos privados nome, cpf e salario; e um construtor que deve receber como parâmetro nome e cpf. Todos os atributos devem ter propriedades (getters) definidos.

- a) Crie uma classe TrabalhadorAssalariado e outra TrabalhadorHorista, ambas herdando de Funcionario
- b) A classe TrabalhadorAssalariado possui o método definirSalario(salario) que recebe um valor como parâmetro e o atribui ao atributo salario.
- c) A classe TrabalhadorHorista possui os atributos privados valorHora e horasTrabalhadasMes, com suas respectivas propriedades (getters e setters).
- d) A classe TrabalhadorHorista também possui o método calcularPagamento(), que ao ser invocado deve calcular o valor do salário e preencher este atributo. O salário é obtido multiplicando-se as horas trabalhadas no mês pelo valor da hora. Crie validações que verifiquem se os atributos necessários para o cálculo do salário estão preenchidos. Se não estiverem emita um aviso.
- e) Crie um programa de teste e três instâncias de cada uma das classes TrabalhadorAssalariado e TrabalhadorHorista, coloque-os em uma lista e percorra imprimindo os nomes dos funcionários com seus pagamentos do mês.

Q6. Considere a definição de um ponto $P = (x; y)$, onde x e y são as coordenadas do ponto. Implemente uma classe para definir pontos, na qual estejam definidas as seguintes operações: Sejam dois pontos $P1 = (x1; y1)$ e $P2 = (x2; y2)$

- a) Impressão do ponto $P1$ na tela: $(x1, y1)$
- b) Adição $P1 + P2$: retorna um novo objeto $P3 = (x1 + x2; y1 + y2)$
- c) Subtração $P1 - P2$: retorna um novo objeto $P3 = (x1 - x2; y1 - y2)$
- d) Multiplicação $P1 \times P2$: retorna um escalar de valor $x1 \times x2 + y1 \times y2$
- e) Multiplicação de um escalar por um ponto, como em $n \times P1$, retorna um novo objeto $P3 = (n \times x1; n \times y1)$. **Obs: Para essa operação o método a ser sobrecarregado deve ser o `__mul__`.**

Q7. Desenvolva uma classe Relogio que represente um horário no formato HH:MM:SS. Com o auxílio de prints, imprima mensagens de aviso e evite que sejam atribuídos valores inválidos para hora, minuto e segundo. Sobrecarregue os métodos `__str__`, `__add__`, `__sub__`, `__eq__`, `__gt__`, `__lt__` de forma que seu programa se comporte de acordo com as saídas abaixo

```
>>>r0 = Relogio(16 ,61 ,54)
Horario digitado invalido !
>>>r1 = Relogio(18 ,37 ,32)
>>>r2 = Relogio(20 ,0 ,30)
>>> print(r1)
18:37:32
>>> print(r2)
20:00:30
>>>r3 = r1 + r2
>>> print(r3)
14:38:02
>>>r4 = r3 - r2
O primeiro horario deve ser maior ou igual ao segundo
>>> print(r4)
None
>>>r4 = r2 - r3
>>> print(r4)
5:22:28
>>>r1 == r2
False
>>>r1 == Relogio(18 ,37 ,32)
True
>>>r3 > r3
False
>>>r3 > r2
False
>>>r2 > r3
True
```