



UNIVERSIDADE FEDERAL
DO RIO DE JANEIRO

Centro de Ciências Matemáticas e da Natureza
Instituto de Computação
Lista 2 - Computação II

Professor: Giomar Sequeiros

Período: 2022 – II

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(s) nome(s) completo(s) do(s) aluno(s) na forma de comentário.
Por exemplo: # Autor: João da Silva, Ana Maria da Silva
- O **código** deve estar devidamente **comentado** indicando os tipos de entrada e saída.

Encapsulamento, herança e sobrecarga de métodos

Q1. 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 = $2*3.14*(raio)^2 + 2*3.14*raio*altura$

Q2. 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 respectivos 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.

Q3. 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 `__rmul__`.**

Q4. 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 dever 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
```