



Nome: \_\_\_\_\_ Data: \_\_\_\_/\_\_\_\_/\_\_\_\_

## 2ª AVALIAÇÃO SOMATIVA

**Questão 1 – Crie uma classe** em Python que suporte a criação do objeto 'meuCarro' com seus métodos, conforme as linhas de código abaixo. ATENÇÃO: o código abaixo não poderá sofrer nenhum ajuste e/ou alteração. Sua classe é que deverá ser TOTALMENTE compatível com ele:

```
meuCarro = Carro(12);           # consumo de 12 km/l
meuCarro.abastecer(40);          # abastece com 40 litros
print(meuCarro.exibirCombustivel()) # Mostra o combustível que resta no tanque.
print(meuCarro.andar(160))        # anda 160 quilômetros.
print(meuCarro.exibirCombustivel()) # Mostra combustível restante
print(meuCarro.andar(280))        # anda 280 quilômetros.
print(meuCarro.exibirCombustivel()) # Mostra combustível restante
```

**QUESTÃO 1) 3,5 pontos**

**RA1**

Escreva aqui sua classe

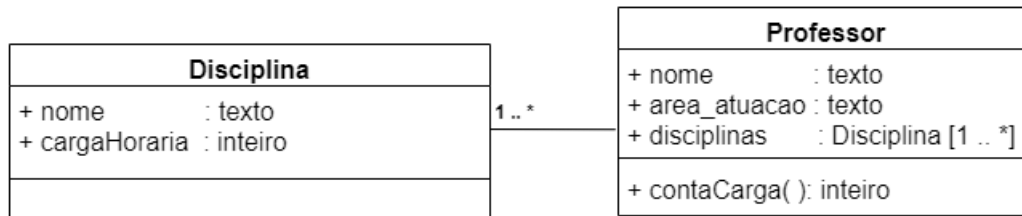

**Questão 2-** Construa seu programa orientado a objetos em Python no local indicado na prova e que implemente exatamente o seguinte Diagrama de Classes (seguir a mesma nomenclatura).

**OBS-1:** A criação e atribuição dos elementos de um atributo que é uma lista, deve ser feita no método construtor.

**OBS-2:** utilizar dicionário de dados para a passagem de parâmetros no construtor de ambas as classes.

**OBS-3:** os atributos nome (da classe Disciplina) e nome (da classe Professor) devem exigir entrada de valor obrigatória. Os demais deverão ter seus valores padrão (*default*) definidos.

**OBS-4:** o método **contaCarga()** da classe Professor serve para obter o total de carga horária atribuída ao Professor.



Após ter implementado as classes, crie no próprio código **.py** os seguintes objetos:

6 objetos disciplina

2 objetos professor

E, logo em seguida, escreva no arquivo **.py** a(s) linha(s) de código que executa(m) o que está sendo solicitado abaixo:

1. Mostre na tela o total de carga horária de cada professor (faça uso do **contaCarga()**)
2. Liste o nome de cada professor e os respectivos nomes das disciplinas associadas a eles.

#### MATERIAL DE APOIO – PRINCIPAIS OPERADORES E COMANDOS EM PYTHON

Atribuição / Entrada de dados		Operadores Lógicos e Relacionais	
Atribuição manual de valor na variável x	x = valor	E lógico	and
Leitura de informação do usuário para x	x=input('digite algo:')	OU lógico	or
Saída de dados		NEGAÇÃO lógica	not
Imprimir uma mensagem na tela	print('exemplo')	Igual a	==
Imprimir variável x na tela	print('valor:', x)	Diferente de	!=
Operadores Aritméticos		Maior que	>
Somar duas variáveis x e y	x + y	Maior ou igual a	>=
Subtrair duas variáveis x e y	x - y	Menor que	<
Multiplicar duas variáveis x e y	x * y	Menor ou igual a	<=
Divisão inteira de x por y	x // y	Funções com Listas	
Divisão em ponto flutuante de x por y	x / y	Inserir elemento	lista.append(elemento)
Resto da divisão de x por y	x % y	Ordena crescente	lista.sort()
Potência, fazendo x elevado a y	x ** y	Ordena decrescente	lista.sort(reverse=True)
Raiz, fazendo raiz y de x	x ** (1/y)	Quantidade de elementos numa lista	len(lista)
Logaritmo natural de x	math.log(x)	Funções Gerais	
Funções com Dicionário de Dados		Módulo de x	abs(x)
Usa valor padrão (default)	dicionario.pop(chave, valor)		

Escreva aqui seu programa em Python