

Nº Mecanográfico: _____

Nome: _____

Duração: 2 horas

- O exame é individual. Não são permitidos quaisquer equipamentos de comunicação.
- O exame deve ser respondido no computador. As respostas ao exame têm de ficar **todas** dentro de uma pasta com o formato: “/home/user/PycharmProjects/<num_mecanografico>”.
 - Para quem utilizar o PyCharm apenas tem de nomear um projeto com o seu número mecanográfico.
- Cada ponto (retângulo) do exame deve ficar num ficheiro próprio (dentro do projeto). O nome desse ficheiro deve indicar a questão a que se refere. Ex: ponto1.py
- Quando terminar o exame **não desligue** o computador e informe o docente para proceder à recolha.
- Leia atentamente as questões antes de responder, se tiver alguma dúvida de interpretação esclareça-a com o docente.

1. Considere a seguinte classe vazia:

```
class Veiculo:  
    pass
```

- Adicione os seguintes atributos públicos (de instância): **nome** , **cor** e **num_rodas**. Serão respetivamente do tipo str, str e int.
- Proceda à inicialização desses atributos.
- Instancie um objeto com esta classe e atribua a instância à variável **bicicleta**.
Nome=”Pasteleira” Cor=”Vermelho” Rodas=2
- Crie um método (instância - self) **descricao** que imprima no terminal as três características do veículo, concatenadas.
Teste com `print(bicicleta.descricao())`
- Documente a classe com a indicação do que faz e que parâmetros armazena.

2. Num novo ficheiro:

- a) Altere os atributos anteriores para privados.
- b) O atributo `num_rodas` deve poder ser lido e modificado publicamente. Crie os métodos `setter` e `getter` para o efeito.
- c) Garanta que `num_rodas` é sempre positivo. Se o valor for menor ou igual a zero, a variável toma o valor 1 e deve aparecer uma mensagem de aviso no terminal.

3. Num novo ficheiro, repita o ponto anterior utilizando o property: `num_rodas`.

4. Crie uma classe `CorRGB`. Esta classe deve armazenar o código de uma cor e uma string com o nome dado a essa cor.

O código da cor respeita o formato RGB (Red, Green, Blue) e pode ser armazenado em três variáveis separadas, cada uma respeitante a uma componente da cor, vermelho, verde e azul. Se achar mais prático, também pode ser armazenado numa única variável do tipo tuple: (vermelho, verde, azul).

Exemplo de código: o branco seria armazenado com o tuple (255, 255, 255).

Cada componente da cor pode variar de 0 a 255. Não pode conter valores negativos nem valores \geq a 256.

A inicialização do objeto deve ter a assinatura:

```
def __init__(red, green, blue, nome=None):
```

- a) Crie a classe conforme descrito.
- b) Crie um método `getHTML`, que devolva a cor armazenada no formato de cor HTML (“#RRGGBB”). O formato tem as três componentes RGB representadas individualmente em notação hexadecimal (HEX). Ex: Branco = “#FFFFFF”

Dica para a conversão: `print('#%02X%02X%02X' % (255, 255, 255))`

- c) Crie um método dentro da classe que faça o contrário, obter o objeto RGB a partir de uma string em formato HTML. Avalie bem que tipo de método deve criar (de Instância | de Classe | Estático). Utilize um comentário no código para justificar a opção.

Dica:

```
string_html = "#FFFFFF"
h = string_html.lstrip('#')
rgb = tuple(int(h[i:i+2], 16) for i in (0, 2, 4))
print('RGB =', rgb)
```

5. Num novo ficheiro, crie uma função que calcule a média de duas cores. Trata-se da média aritmética de cada uma das três componentes da cor. A função tem dois argumentos, `cor1` e `cor2`, que devem ser instâncias de `CorRGB`. A função retorna um novo objeto `CorRGB`.

```
def media(cor1, cor2):
```

Se não concluiu a questão anterior pode utilizar esta classe para referência:

```
class CorRGB:
    def __init__(self):
        self.red = 20
        self.green = 40
        self.blue = 100
```

6. Num novo ficheiro: A classe `Veiculo` do exercício 1 utiliza uma string como representação da cor do veículo. Implemente uma solução que utilize um objeto `CorRGB` em vez da string.

Lembre-se: A classe `CorRGB` armazena uma string que descreve a cor e pode sempre fazer override ao `__str__` para obter o texto da classe na imprimir.

7. Num novo ficheiro: Defina e teste uma classe `Carro`, **baseada** na classe `Veiculo` do exercício 2, mas com o número de rodas mínimo igual a 3.

Considere ainda que o `Carro` tem de armazenar informação sobre o número de lugares/pessoas que consegue transportar e que este valor tem de ser superior a 1 e inferior a 100.