

Em Python 3, escreva a classe `GestorCodigoPostais`, no ficheiro `gestor_codigos_postais.py`:

- Para além de `self`, o construtor da classe `GestorCodigoPostais`, tem o argumento `codigos_postais`;
- Os objetos do tipo `GestorCodigoPostais` têm o atributo `codigos_postais`, que é uma lista de objetos do tipo `CodigoPostal` ;
- O atributo `codigos_postais` é inicializado, no construtor, com o valor do argumento `codigos_postais`;
- Os objetos do tipo `codigos_postais` têm os métodos `validar_codigos_postais` e `obter_codigos_postais_por_localidade`;
- O método `validar_codigos_postais` valida o atributo `codigos_postais`, de forma ao atributo `codigos_postais` conter apenas códigos postais válidos;
- Para um código postal ser válido o primeiro número tem que ter quatro dígitos, o segundo número tem que ter três dígitos e a localidade tem que estar apenas em letras capitais;
- O método `obter_codigos_postais_por_localidade`, recebe o argumento `localidade`, o método retorna uma lista de objetos do tipo `CodigoPostal` do atributo `codigos_postais` cujo a localidade é igual ao argumento `localidade` recebido;
- A função `seed`, é utilizada para inicializar um gerador de número aleatórios;

```
1 import random
2 import string
3 from gestor_codigos_postais import GestorCodigoPostais
4
5 random.seed(61526)
6
7 class CodigoPostal:
8
9     def __init__(self, digitos4, digitos3, localidade):
10
11         self.digitos4 = digitos4
12         self.digitos3 = digitos3
13         self.localidade = localidade
14         self.separador_digitos = '-'
15         self.separador_localidade = ' '
16
17     def print_codigo_postal(self):
18
19         print(f"{self.digitos4}{self.separador_digitos}{self.digitos3}{self.separador_localidade}{self.localidade}")
20
21
22
23 list_localidades = ["LOURES", "MAFRA", "OEIRAS", "CASCAIS", "lisboa", "ESPINHO", "MAIA", "Amarante", "valongo", "OVAR", "Pombal", "Batalha"]
24
25 lista_codigos_postais = []
26 for i in range(1000):
27
28     numero_4_digitos = random.randint(1, 12444)
29     numero_3_digitos = random.randint(1, 1153)
30     localidade = random.choice(list_localidades)
31     codigo_postal_i = CodigoPostal(numero_4_digitos, numero_3_digitos, localidade)
32
33     lista_codigos_postais.append(codigo_postal_i)
34
35
36 g = GestorCodigoPostais(lista_codigos_postais)
37 g.validar_codigos_postais()
38 print(len(g.codigos_postais))
39 g.codigos_postais[194].print_codigo_postal()
40 codigos_postais_localidade = g.obter_codigos_postais_por_localidade("MAFRA")
41 print(len(codigos_postais_localidade))
```

Indique se as seguintes perguntas são verdadeiras ou falsas.