

ESMAD | TSIW | POO
Ficha de Exercícios nº5 – Classes

Abra o Visual Studio Code e resolva os seguintes exercícios::

1. Definição de classes (ficheiros ex1.html/ex1.js)

- a. Defina uma classe **Rectangle** para representar um retângulo. Um retângulo tem uma altura, largura e uma cor de preenchimento (sempre branca).
- b. Defina dois construtores:
 - i. Um que cria um retângulo com a largura e altura especificadas.
 - ii. Um sem argumentos que cria um retângulo padrão (largura = 2 e altura =1) .
- c. Defina métodos get e set para as três propriedades da classe. A exceção é para a propriedade **color** que não deverá ser possível reescrever o seu valor
- d. Defina dois novos métodos:
 - i. Um método chamado **getArea** que retorna a área do retângulo.
 - ii. Um método chamado **getPerimeter** que retorna o perímetro do retângulo.
- e. Crie duas instâncias da classe **Rectangle**
 - i. Uma com os valores padrão
 - ii. Outra com a largura de 10 e altura de 5
- f. Imprima na consola para ambas as instâncias: as larguras, alturas, cores, áreas e perímetros.

2. Definição de classes (ficheiros ex2.html/ex2.js)

- a. Defina uma classe **Car** para representar um carro.



- b. Um carro é composto por uma marca, matrícula, cor, depósito atual (em litros) e tipo de combustível. Crie um construtor que recebe todos estes parâmetros e que inicialize as respetivas propriedade públicas
- c. Adicione dois carros:
 - i. "Ford", "91-GH-15", "verde", 40, "Gasóleo"
 - ii. "Opel", "23-AB-23", "branco", 50, "Gasolina"
 - iii. "Nissan", "12-CX-45", "preto", 22, "Gasóleo"
- d. Crie os seguintes métodos:

- i. Método para atualizar a cor de um carro
- ii. Método para abastecer o depósito. Deve receber como parâmetro o nº de litros abastecidos
- iii. Método que recebe o nº de quilómetros percorridos e que altere o nº de litros tendo em conta a seguinte consumo base: 5l/100km
- e. Adicione os objetos a um array **cars**
- f. Crie as seguintes funções:
 - i. uma função que devolva o nº de carros de uma determinada marca passada como parâmetro
 - ii. uma função que dado um tipo de combustível devolva a soma dos litros dos carros que têm esse tipo de combustível

3. Definição de classes (ficheiros **ex3.html/ex3.js**)

Os dados são pequenos poliedros gravados com determinadas instruções. O dado mais clássico é o cubo (seis faces), gravado com números de um a seis.



- a. Para este exercício deverá criar uma classe **Dice** para representar um dado e que nos permitirá rolá-lo e tirar valores que variam de 1 a 6, ou seja, um dado de seis lados.
 - i. Defina um construtor sem argumentos que cria um objeto da classe **Dice**;
 - ii. Defina os métodos set e get para a propriedade **faceValue** que guardará o valor atual da face do dado.
 - iii. Defina os métodos:
 - 1. Um método chamado **getQuantityFaces** que retorna um número representando a quantidade de faces do dado
 - 2. Um método **roll** que vai "rolar" o dado e guardar o resultado na propriedade **faceValue**. O valor sorteado deverá estar na gama de 1 até a quantidade de faces do dado;
 - iv. Arranje forma de lançar o dado 100 vezes e apresentar uma tabela de frequência similar à apresentada de seguida:

Face	1	2	3	4	5	6
Frequência	-	-	-	-	-	-

