

## Prática 2 Herança e Polimorfismo

### Tópicos

- Herança
- Polimorfismo
- `__repr__` e `__str__`

### Exercício 2.1 - Veículos

Com base na classe **Vehicle** do Guião anterior crie classes para veículos a motor de combustão e veículos elétricos, tendo em conta o seguinte:

- Todos os veículos têm informação sobre velocidade máxima, quilómetros percorridos, cor, marca e matrícula;
- Os veículos com motor de combustão têm cilindrada do motor (em cm<sup>3</sup>);
- Os veículos elétricos têm potência total e número de motores;
- Todas as classes devem gerar representações adequadas a humanos (com `__str__`).

Ajuste o seu código para obter o resultado abaixo com o seguinte código de teste:

```
v1 = ElectricVehicle(190,0,'black','Tesla',"DZ-59-27",100,1)
v2 = CombustionVehicle(310,0,'red','Ferrari',"OF-00-00",3000)
print(v1)
print(v2)
```

Resultado:

```
DZ-59-27 : Tesla, black, 190 km/h, 0 km, 1 motor(s), 100 KW
OF-00-00 : Ferrari, red, 310 km/h, 0 km, 3000 cm3
```

### Exercício 2.2- Figuras Geométricas

Retome o exercício das figuras geométricas do guião anterior e desenvolva novas versões das classes fazendo agora uso de herança. Deve ser possível determinar a área e perímetro de qualquer das figuras.

Teste com uma adaptação do seguinte código:

```
class Figures:
    def randShape():
        rn = random.randrange(0,2)
        if rn == 0:
            return Circle()
        elif rn == 1:
            return Square()
    def main():
        figures = []
        for i in range(9):
            figures.append(Figures.randShape())
        for fig in figures:
            fig.area()
```

## Exercício 2.3- Loja online

Considere as seguintes entidades que fazem parte de uma Loja online:

- Produto, caracterizados por nome, código identificativo (inteiro) e quantidade disponível em stock, preço base sem impostos em Euros (float). O código é automático, começa em 1, e vai sendo incrementado sempre que se cria uma instância de um produto (qualquer que seja).

Existem os seguintes tipos de produtos:

- Livro, caracterizado pelo título (string) e um conjunto de autores, não repetidos;
- Telemóvel, caracterizado por nome contendo marca e modelo;
- [TPC] Televisor, caracterizados pelo nome contendo marca e modelo, diagonal do ecrã em polegadas (int);
- [TPC] Electrodoméstico, tendo por nome a concatenação de marca e modelo, código, quantidade disponível em stock e preço base sem impostos

Deve ser possível calcular o preço de venda ao público de qualquer destes produtos, tendo em conta que aos livros se aplica IVA a 6% e que para os restantes o IVA é de 23%.

Represente adequadamente estas entidades. Crie construtores, os métodos set/get que lhe pareçam adequados, bem como métodos que sejam necessários para cada classe.

Teste as classes desenvolvidas usando o programa disponibilizado no elearning<sup>1</sup> ([loja\\_teste.py](#)) e procurando aproximar o seguinte output:

Livro 1 tem 2 autores

Livro 2 tem 3 autores

Lista de Todos os Produtos:

1 Livro 1	100	15.00	15.90
2 Livro 2	450	23.00	24.38
5 Frigorífico	2	780.00	959.40
6 Fogão	1	423.00	520.29
4 XWZModel A	13	560.00	688.80
3 Sony KX1188	3	1000.00	1230.00

## Exercício 2.4- Alunos e Professores [TPC]

Crie classes Student e Teacher tendo a base da classe Person disponibilizada no elearning (ficheiro [Person.py](#)). Estudantes têm nome, cartão de cidadão, data de nascimento, curso e número mecanográfico. Os professores têm nome, cartão de cidadão, data de nascimento, lista com UCs que leccionam. Altere a classe Person se necessário.

Para testar crie uma lista com todas as pessoas associadas (alunos e professores) a uma turma de 10 alunos e mostre essa lista.

---

<sup>1</sup> O código de apoio a este guião encontra-se no diretório guiao2 do material partilhado para as aulas práticas.

## Exercício 2.5– Agência de viagens [TPC]

Pretende-se desenvolver um programa que possibilite a gestão de alguns produtos numa agência de viagens. As entidades principais neste sistema de informação são alojamentos (apartamentos e quartos em hotel). Devem ser suportadas as características seguintes:

- Um alojamento tem um código (string), nome (string), local (string), preço por noite (float), disponibilidade (booleano) e avaliação (float, entre 1.0 e 5.0). Deve permitir as operações de check-in e check-out.
- Um apartamento é um alojamento mas tem mais informação sobre o número de quartos.
- Um quarto de hotel é um alojamento mas tem mais um campo que indica o tipo (single, double, twin, triple).

Represente adequadamente todas estas entidades. Crie construtores, métodos set/get que lhe pareçam adequados, bem como métodos que sejam fundamentais para cada classe. Crie um programa de teste para criar várias instâncias de cada uma das classes.