

## Guião 3 – Herança e Polimorfismo

### Objectivos:

- Herança
  - Override – redefinição de métodos
- Polimorfismo
  - Funções
  - Classes Abstratas

1. Considere as seguinte função e classes:

```
def cozinha_no_forno( item ):  
    tempo_de_cozedura = item.getMinutosDeCozedura()  
    print("O forno está a cozinhar durante",  
          tempo_de_cozedura,"minutos.")  
  
class Pizza():  
    pass  
  
class Broa():  
    pass
```

Complete as classes de forma a que se possa passar um objeto de `Pizza` ou `Broa` na função `cozinha_no_forno`.

Obs: Este é um exemplo de polimorfismo de funções, onde a função `cozinha_no_forno` exige que o parametro passado em `item` implemente o método `getMinutosDeCozedura`. No ponto 5 voltaremos a este exemplo.

2. Considere a classe

```
class Pessoa:  
    def __init__(self, nome, sobrenome):  
        self.nome = nome  
        self.sobrenome = sobrenome  
  
    def getNome (self):  
        return self.nome + " " + self.sobrenome
```

Crie uma classe `Aluno` que derive de `Pessoa` e inclua um atributo com o número mecanográfico. Inclua um método `getAluno` que retorne uma string com o nome completo e o mecanográfico separados por vírgula.

3. Os métodos `getNome` e `getAluno` estão a retornar uma descrição escrita dos respectivos objetos. Faça o override ao `__str__` por forma a que essa descrição possa ser obtida a partir do `print`.

Ex: `print( Aluno("João", "Silva", 1234) )`

4. Recorrendo à classe `Celsius` do último guião. Vamos refazer a função `soma(lista)`, que devolvia um objeto `Celsius` com a soma das temperaturas na lista. Na classe `Celsius`, implemente o método `__add__` de forma a que seja possível somar dois objetos `Celsius`, diretamente, recorrendo ao operador `+`.

Ex: `Celsius(5) + Celsius(10)`

5. Voltando ao ponto 1: As classes `Pizza` e `Broa` derivam da classe `object` (implicitamente). Não há nada que obrigue ou indique a necessidade de implementarem o método `getMinutosDeCozedura`.

Crie uma classe abstrata `VaiAoForno` que descreve a necessidade (contrato) de se implementar o referido método. As classes `Pizza` e `Broa` devem agora derivar de `VaiAoForno`.

6. Com os conhecimentos que já adquiriu sobre herança, reveja o ponto 15 do guião 1.

7. Finalize o guião e envie via moodle.