

05-POO-classes-e-objetos-dinamicos

March 3, 2020

1 Programação Orientada aos Objetos (POO) - parte V

Pedro Cardoso

(ISE/UAlg - pcardoso@ualg.pt)

1.1 Classe são “abertas”

- Em Python as classes que não são *builtins* podem ser alteradas em tempo de execução, devido a natureza dinâmica da linguagem, i.e., por exemplo, é possível acrescentar métodos e atributos novos. A mesma lógica se aplica aos objetos.

```
In [1]: class Carro:
        def __init__(self, cor, marca):
            self.cor = cor
            self.marca = marca

In [2]: Carro.ano = 1970  # variavel da classe!

In [3]: carro_1 = Carro('Vermelha', 'Fiat')
        print(dir(carro_1))  # tem o atributo ano

['__class__', '__delattr__', '__dict__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__'

novo metodo (print) para a classe

In [4]: def my_repr(self):
        return '{} de cor {} do ano {}'.format(
            self.marca, self.cor, self.ano)

        Carro.print = my_repr

        print(dir(Carro))  # tem o metodo print

['__class__', '__delattr__', '__dict__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__'

In [5]: carro_2 = Carro('vermelha', 'Fiat')
        carro_2.print()
```

```
Out[5]: 'Fiat de cor vermelha do ano 1970.'
```

```
In [6]: carro_2.dono = 'Margarida'    # atributo só do objeto carro_2
```

```
print(dir(carro_2))
```

```
['__class__', '__delattr__', '__dict__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__
```

```
In [7]: print(carro_2.dono)
```

Margarida

```
In [8]: print(carro_1.dono) # preparem-se para o erro!!!
```

```
-----
```

AttributeError

Traceback (most recent call last)

<ipython-input-8-fb5ddb5e3f65> in <module>()

```
----> 1 print(carro_1.dono) # preparem-se para o erro!!!
```

AttributeError: 'Carro' object has no attribute 'dono'