

Programação Orientada a Objetos

A dificuldade que se coloca na POO é a diferença de abordagem do problema. Enquanto a programação estruturada tem como principal foco as ações (procedimentos e funções), a POO preocupa-se com os objetos e os seus relacionamentos. Além do conceito de **objeto**, a POO tem como fundamentos os conceitos de **encapsulamento**, **classe**, **herança** e **polimorfismo**.

NOÇÃO DE OBJETO

Um objeto é uma entidade abstrata ou existente no mundo real, sobre a qual se pretende incorporar num sistema de informação. O objeto é caracterizado por um conjunto de propriedades, um comportamento e uma identidade.

- ✚ As propriedades são as características que definem o objeto - **atributos**;
- ✚ o comportamento é definido como as operações que o objeto pode efetuar sobre si próprio ou outros objetos – **métodos**;
- ✚ a identidade permite identificar um objeto em particular como único num conjunto de objetos idênticos - **valores dos atributos**

Por exemplo, considere os objetos da figura 1:



Fig. 1 - Exemplo de objetos — carros

- ✚ Apesar de serem do mesmo tipo — carros — possuem características e funcionalidades que as distinguem.
- ✚ O objeto carro possui propriedades, como velocidade, cilindrada, altura, comprimento, cor, marca e matrícula.
- ✚ O objeto carro também possui procedimentos/comportamentos, como ligar, desligar, acelerar e parar.
- ✚ Finalmente, os valores do número de série, da marca, da cor, etc. constituem a identificação entre os demais objetos do mesmo tipo.

Objeto	Atributos	Valor do atributo	Métodos
Carro	Matrícula, marca, nº de série, modelo, cor	50-AM-60, Renault, 45327/15, Clio, verde	Acelerar, travar, buzinar, arrancar
Pessoa	Nome, idade, género, nacionalidade	Ana Vaz, 43, F, Espanhola	falar, andar, pensar

NOÇÃO DE CLASSE

Uma classe define um conjunto de características que são comuns a uma série de objetos com particularidades semelhantes. Para que um objeto seja de determinada classe, terá, obrigatoriamente, de respeitar a especificação da classe – tornando-se numa **instância da classe**. Uma classe é como que um molde a partir do qual se criam objetos de um determinado tipo (**abstração**).

Cada instância de uma classe terá um valor diferente para cada um dos atributos definidos na especificação da classe.

Uma classe é, portanto, uma estrutura de dados que define, em abstrato, os atributos e funcionalidades de uma família de objetos (**instâncias da classe**).

Classes e objetos podem dizer respeito a qualquer tipo de entidades usadas em programação, tais como: janelas, menus, botões de comando, estruturas de dados, caixas de texto, imagens, etc.

- Os objetos, em programação, são unidades de código utilizadas no desenvolvimento de aplicações.
- Os atributos dos objetos são guardados em variáveis internas ao próprio objeto.
- Os valores dos atributos de um objeto podem ser colocados na altura da criação do objeto, sob forma de variáveis deste, e variar durante a execução do programa;
- Um método é semelhante a uma função – uma entidade lógica que aceita determinados parâmetros de entrada e que realiza uma determinada ação, podendo não devolver resposta) mas que é definido no interior de uma classe destinando-se a operar entre os objetos dessa classe.

Classes e objetos

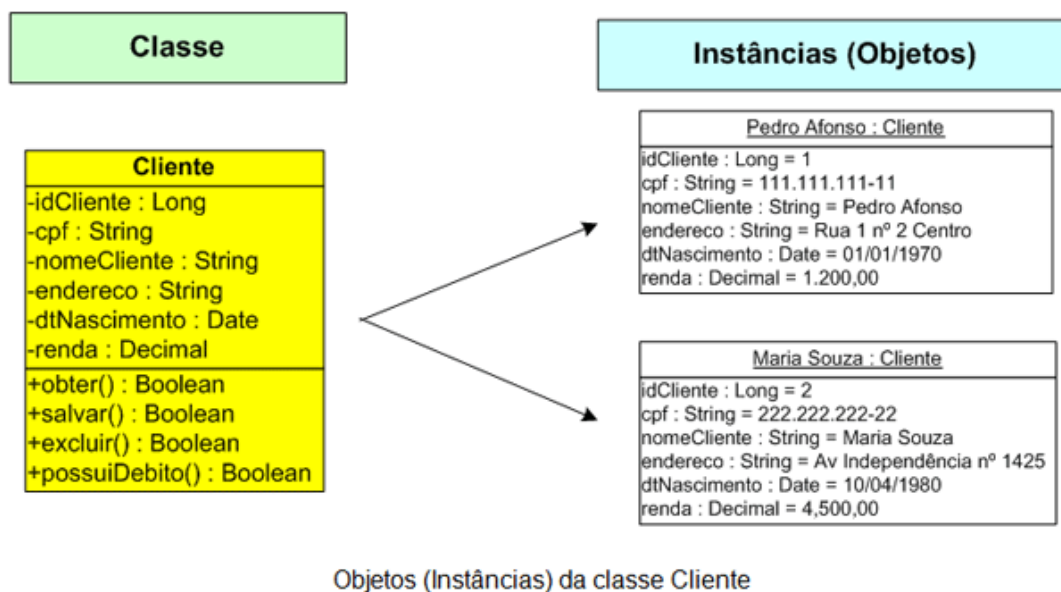
O uso do paradigma de programação orientada a objetos, permite aos programadores definir os seus próprios tipos de dados – **classes** –, que encapsulam atributos e funcionalidades (eventos) dando origem a novos tipos de variáveis. As classes são **instanciáveis** em objetos que são representados por variáveis do tipo das respetivas classes.

Um **objeto** pode ser entendido como uma unidade básica de modularização que agrega, em simultâneo, valores (ditos **atributos**) e funcionalidades (ditas **métodos**). Mais ainda, numa linguagem fortemente tipada como **Python**, o menu de atributos e métodos de um dado objeto determina precisamente o seu tipo (ou **classe**).

Assim:

- Os programas orientados a objetos são conjuntos de classes;
- Os objetos interagem, pedindo uns aos outros, para executarem as respetivas funcionalidades;
- As classes escondem a implementação dos seus atributos e funcionalidades, mas expõem a informação necessária para que terceiros possam invocar a execução das suas funcionalidades.

Classes e Objetos



Variáveis de Instância e Variáveis estáticas

Os atributos dos objetos são guardados em **variáveis de instância**. Estas variáveis são sempre atributos do objeto corrente, daí que o seu nome seja precedido de **self**.

As classes podem conter **variáveis estáticas**, isto é, variáveis que fazem parte da classe, mas que são atributos dos objetos.

Visibilidade das variáveis

A **interface** de uma classe é determinada pelo conjunto de atributos e operações (métodos) disponibilizados, também designados por **membros** da classe. A visibilidade dos membros de uma classe pode ser dividida em 3 categorias:

- Pública (**public**) – A interface da classe está visível para todos os objetos do sistema de software;
- Protegida (**protected**) - A interface da classe é visível apenas para os membros da própria classe e para as subclasses;
- Privada (**private**) – A interface só é visível para a própria classe.

Esta característica de definir a visibilidade dos objetos é chamada de **encapsulamento**, e é um processo pelo qual se “escondem” os detalhes de conceção e implementação do objeto.

As operações associadas aos objetos podem ser classificadas nas seguintes categorias:

Modificador (**modifier**) – Operação que altera o estado de um objeto;

Seleto (bselector) – Operação que acede ao estado de um objeto;

Construtor (bconstructor) – Operação que cria um objeto ou inicializa o seu estado;

Destrutor (bdestructor) - Operação que elimina um objeto.

Carro
- placa : String - numChassi : int
+ acelerar() : void + frear() : boolean

Exercícios (consulta o NB10. 10794 POO em Python):

Para cada alínea grava um programa com o nome: **classe1_a.py** , em que a é o nº da alínea)

1. Estudante de Informática

1.1. Define a classe **EstudanteInf** com as seguintes Variáveis de instância:

Variável	Designação	Tipo	Significado
instância	Nome	alfanumérica	Nome do aluno
	Teste1	Real	Nota do 1.º teste
	Teste2	Real	Nota do 2.º teste

A classe deverá conter:

- Um construtor que receba as 3 variáveis de instância;
- O método **ClassFinal**, que calculará a média aritmética das notas dos 2 testes;

Elabora um programa que defina a classe **EstudanteINF**, instancie a aluna Ana Paiva, que teve 14 e 16, respetivamente, nos testes 1 e 2, e calcule a sua classificação final.

Output esperado:

Ana Paiva teve a classif. final de 15 valores

1.2 . Estudante de Informática com *getters*

Limita a visibilidade das variáveis de instância da classe **EstudanteInf** do exercício anterior- Nome, Teste1 e Teste2- e acrescenta a esta classe três acessos – *getters* - para ler estas variáveis. O programa principal deve imprimir o nome do aluno, as notas dos dois testes e a classificação final. Repete os alinhamentos de saída.

Output esperado:

Nome	Teste 1	Teste 2	Classif. Final
Ana Paiva	14	16	15

1.3. Estudante de Informática com *getters* e *setters*

Adiciona à classe **EstudanteInf()** do exercício 1.2, três acessos- *setters*- para alterar os valores das variáveis de instante. O programa principal deve alterar o nome do aluno, as notas dos dois testes e imprimir a classificação final atualizada. O olutput deverá ser semelhante ao anterior.

1.4. Estudante de informática com propriedades de acesso

Substitui os métodos de leitura e atribuição de valores às variáveis de instância da classe `EstudanteInf()` do exercício 1.3 por propriedades com as mesmas funcionalidades. Testa o programa alterando o nome e as notas da aluna ANA Paiva para Ana Maria Paiva, 16 e 18.

Output esperado:

```
Digite o nome correto da aluna: carla
Digite as nota do teste 1 da aluna carla: 12
Digite a nota do teste 2 da aluna carla: 13
      Nome      Teste 1   Teste 2   Classif. Final
carla              12        13        13
```

1.5. Estudante de informática com o método de impressão

Adiciona à classe estudante do exercício 3.4 um método para imprimir o nome, as notas dos dois testes e a classificação final de um aluno.

Output esperado:

```
Digite o nome correto da aluna: maria
Digite a nota do teste 1 da aluna maria: 17
Digite a nota do teste 2 da aluna maria: 18
      Nome      Teste 1   Teste 2   Classif. Final
maria              17        18        18
```

1.6. Estudante de informática em módulo separado

Relativamente ao exercício 1.5, guarda a classe `EstudanteInf` em `classe1_6.py` e procede à sua importação no módulo do programa principal.

1.7. Pauta de alunos de informática

Elabore um programa que, recorrendo à classe `EstudanteInf` contida em `classe7_7.py`, instancie os alunos da seguinte lista e crie um dicionário com os nomes e as classificações finais desses alunos:

```
Inf=[["Ana pinto", 14, 15], ["Rui Pinto", 17,18], ["cala silva", 14,10], ["Telmo gomes", 10,12]]
```

Código:

```
from Classe1_7 import *
Inf=[['Ana pinto', 14, 15], ['Rui Pinto', 17,18], ['Carla Silva', 14,10], ['Telmo Gomes', 10,12]]
Pauta={}
|
```

Output esperado:

```
{'Ana Pinto': 15, 'Rui Pinto': 18, 'Carla Silva': 12, 'Telmo Gomes': 11}
```

1.8. Situação dos alunos

Acrescenta à classe `EstudanteInf` contida em `classe1_8.py` um método estático que, recebendo a classificação final do aluno (`cf`), determina a sua situação:

Se `cf` for menor que oito, Escreve “Reprovado”
Se `cf` for inferior a 10, escreve “Admitido à Oral”
Outros valores para `cf`, escreve “aprovado”

Elabora um programa que, instanciando a classe `EstudanteInf`, imprima uma pauta com o nome de cada aluno de informática, a classificação final e a situação final.

Output esperado:

```
{'Ana Pinto': (7, 'Reprovado'), 'Rui Pinto': (18, 'Aprovado'), 'Carla Silva': (9, 'Admitido à Oral'), 'Telmo Gomes': (11, 'Aprovado')}
```

Bibliografia:

<https://www.usandoaccess.com.br/tutoriais/classe-no-access-orientacao-a-objetos.asp>

Carvalho, Adelaide. (2021). Práticas de Python - Algoritmia e programação. Lisboa: FCA

Vasconcelos, J. (2015). Python - Algoritmia e Programação Web. Lisboa: FCA