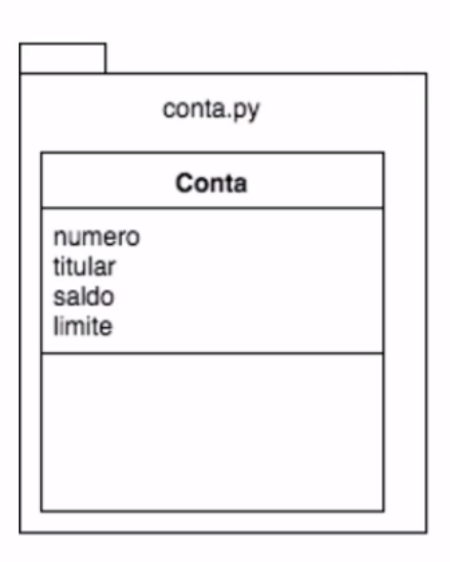
07

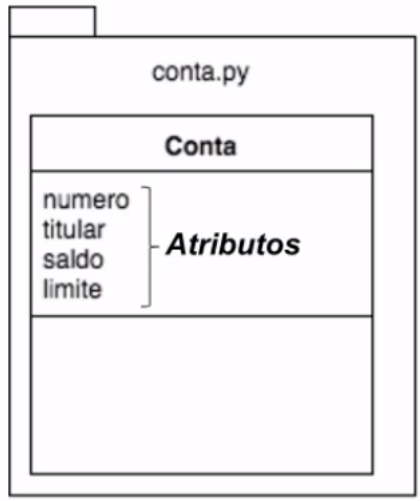
Acessando atributos

Transcrição

Vamos começar a utilizar os atributos da conta. Anteriormente, criamos a classe `Conta()` dentro de `conta.py`. Usaremos UML (Linguagem de Modelagem Unificada), para explicar a estrutura de `Conta`:



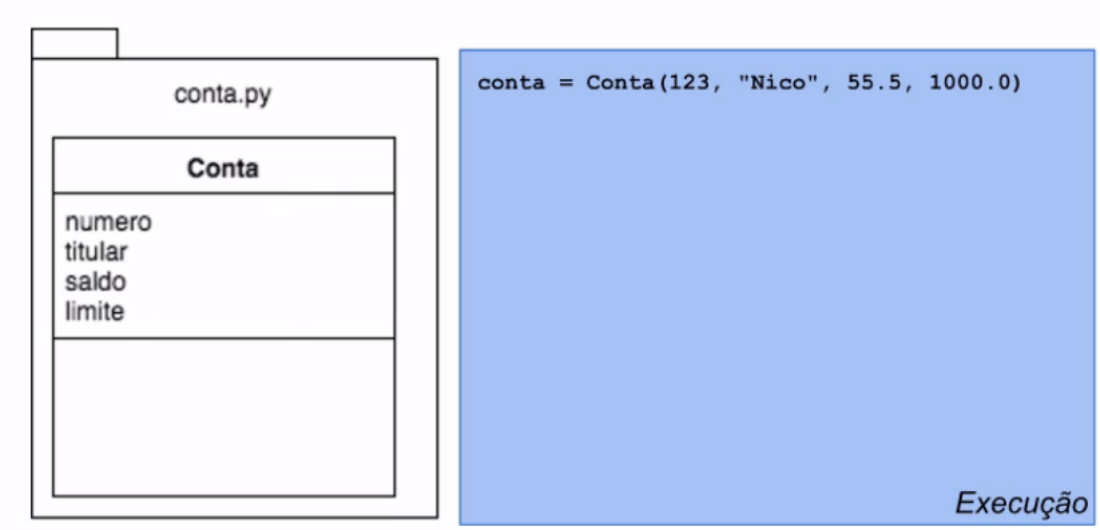
Logo acima temos um diagrama de classes, que mostra os atributos: "numero", "titular", "saldo" e "limite". A linguagem UML é uma anotação visual, usada para descrever o nosso sistema por meio de gráficos e desenhos.



A `Conta()` está dentro de um módulo — que em algumas linguagens receberá o nome de `package` ou `namespace` —, sendo que um módulo poderia ter uma ou mais sintaxes. Por enquanto, usamos a classe para criar uma conta, passando alguns valores como parâmetro para a função construtora.

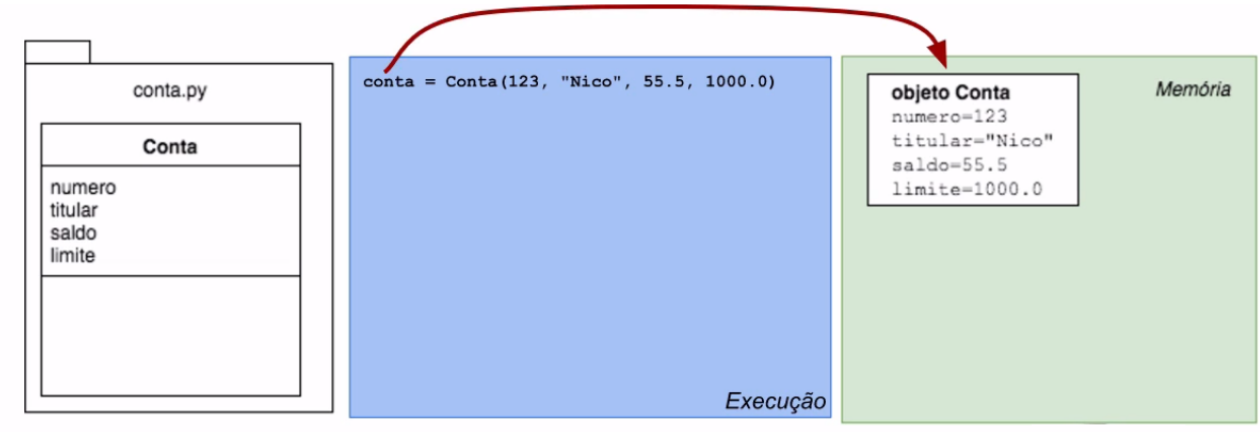
```
>>> conta = Conta(123, "Nico", 55.5, 1000.0)
```

O diagrama ficará da seguinte maneira:



Observe que `__init__` não é exibido no diagrama de classes, porque se trata de uma função implícita, que é chamada automaticamente. Nós chamamos a classe `Conta` seguida de parênteses e, por baixo dos panos, o Python passará os valores para a função construtora. Quando executamos a linha com a referência `conta`, em memória, o Python vai gerar o objeto em que serão guardados os valores.

A referência `conta` sabe onde se encontra o objeto em memória. Mesmo sem sabermos como, o Python aloca isso e encontra espaço; nós não temos controle sobre essa parte do processo.



Falta ainda utilizarmos os atributos. De volta ao PyCharm, no console, vamos importar `Conta` e criar a primeira conta.

```
>>> from conta import Conta
>>> conta = Conta(123, "Nico", 55.5, 1000.0)
Construindo objeto ... <conta.Conta object at 0x10293ae48>
```

Em seguida, criaremos a segunda conta.

```
>>> conta2 = conta(321, "Marco", 100.0, 1000.0)
Construindo objeto ... <conta.Conta object at 0x10293acf8>
```

Temos dois objetos criados, agora, falta fazer alguns atributos. Nós chegaremos ao objeto por meio da referência `conta`, responsável por indicar onde se encontra o objeto. Precisamos dizer usando a linguagem Python "vai para esse objeto e acessa aquele atributo". O pedido de "vai" nas linguagens Orientadas a Objeto é indicado com `.`, e o console compreenderá que queremos fazer algo com esse objeto. No caso, nós queremos mostrar o saldo, logo, executaremos `conta.saldo` no console, e o objeto será impresso.

```
>>> conta.saldo
55.5
>>> conta2.saldo
100.0
```

Ele imprimiu o objeto guardado dentro do atributo `saldo`. Então, para imprimirmos a referência, podemos utilizar os outros atributos, como `conta.titular`, `conta.limite`.

Agora, você pode praticar o conteúdo apresentado com os exercícios, e mais adiante, a classe `Conta` receberá novos métodos.