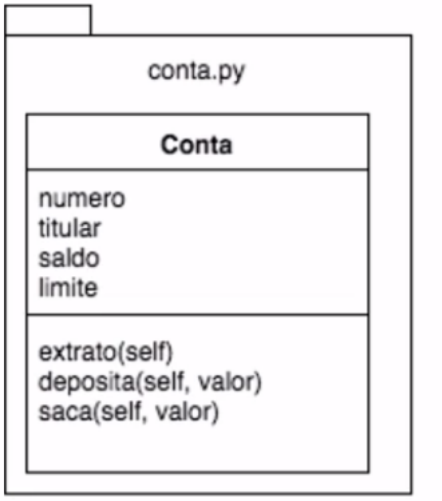
04

Revisão do conteúdo

Transcrição

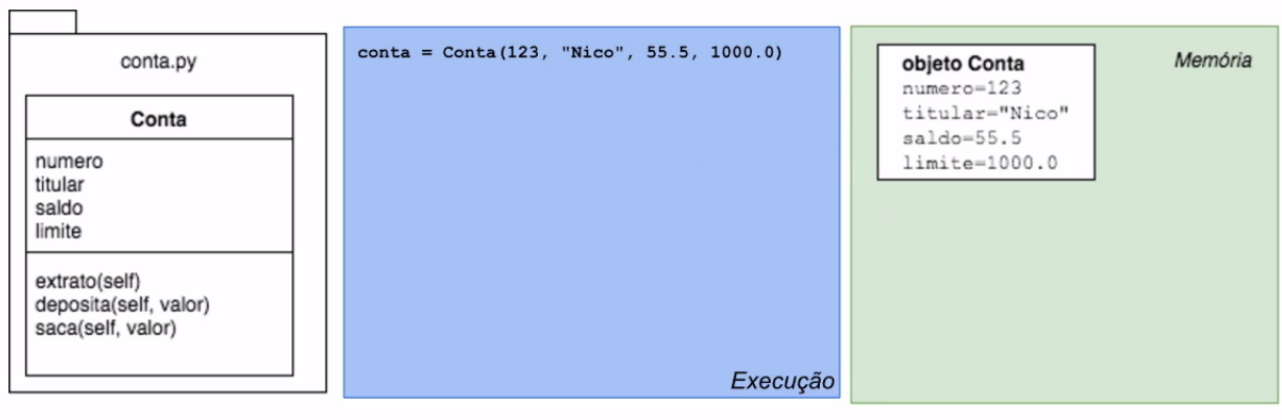
A seguir, temos uma breve revisão. Por isso, se você considera que aprendeu todo o conteúdo visto até agora, pode ir para o próximo vídeo.

O conteúdo visto neste curso, pode ser aplicado para Java, C#, PHP, Ruby ou qualquer outra linguagem Orientada a Objeto.



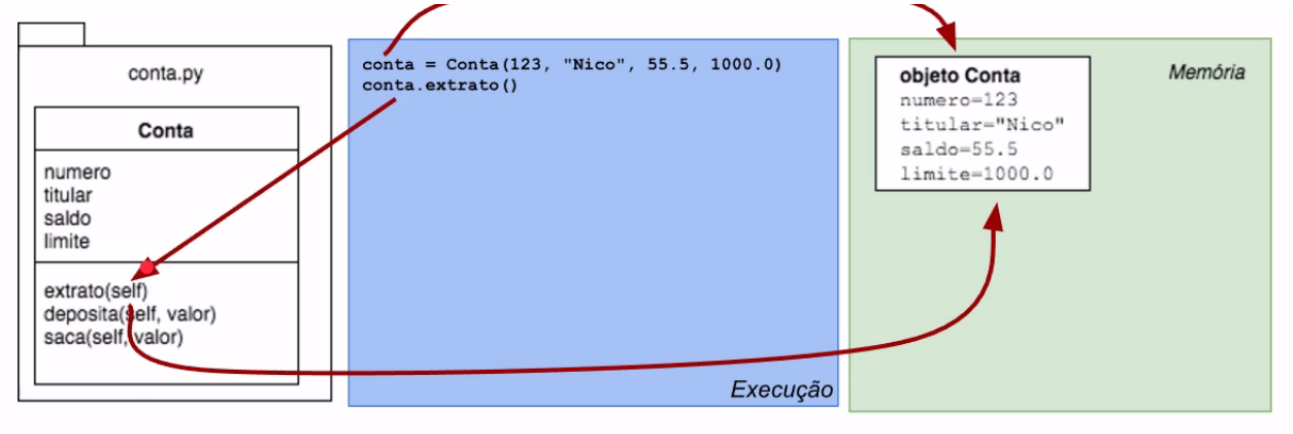
O diagrama da classe `Conta` contém novos dados: os métodos `extrato()`, `deposita()`, `saca()` foram incluídos juntos com seus parâmetros. Normalmente, não incluímos a função construtora `__init__`, considerando que ela é chamada de forma implícita.

Depois, criamos a classe `Conta`, usamos parâmetros, dentro da referência `conta` — como vemos na segunda parte do diagrama. O resultado é o objeto `Conta` criado em memória, com os parâmetros passados para a função construtora, como é ilustrado na parte verde do diagrama.

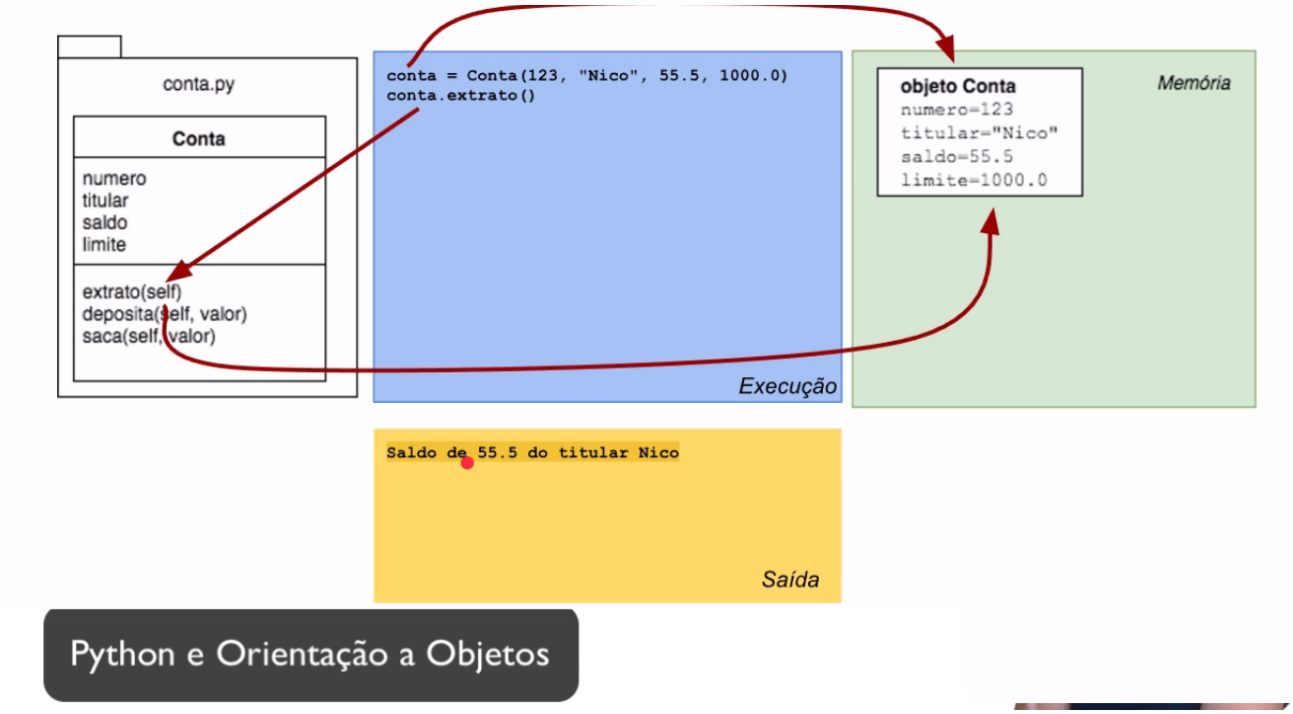


A referência `conta` é devolvida na execução, em que é retornado o endereço dentro de `conta`.

Quando falamos anteriormente sobre **atributos**, mostramos que as referências são utilizadas para acessar o objeto e imprimir um valor. Vimos que o método `extrato()` pode ser utilizado para impressão de valores, como em `conta.extrato()`. Desta vez, a referência foi usada para a chamada do método, assim o objeto será passado automaticamente.



Neste caso, a variável `self` e `conta` serão equivalentes. Ou seja, `self` também sabe onde se encontra o objeto, por isso, dentro do método `extrato()` podemos implementar a maneira de como os dados serão impressos. Quando a função for chamada no console, uma mensagem com o valor do saldo será impressa na saída.



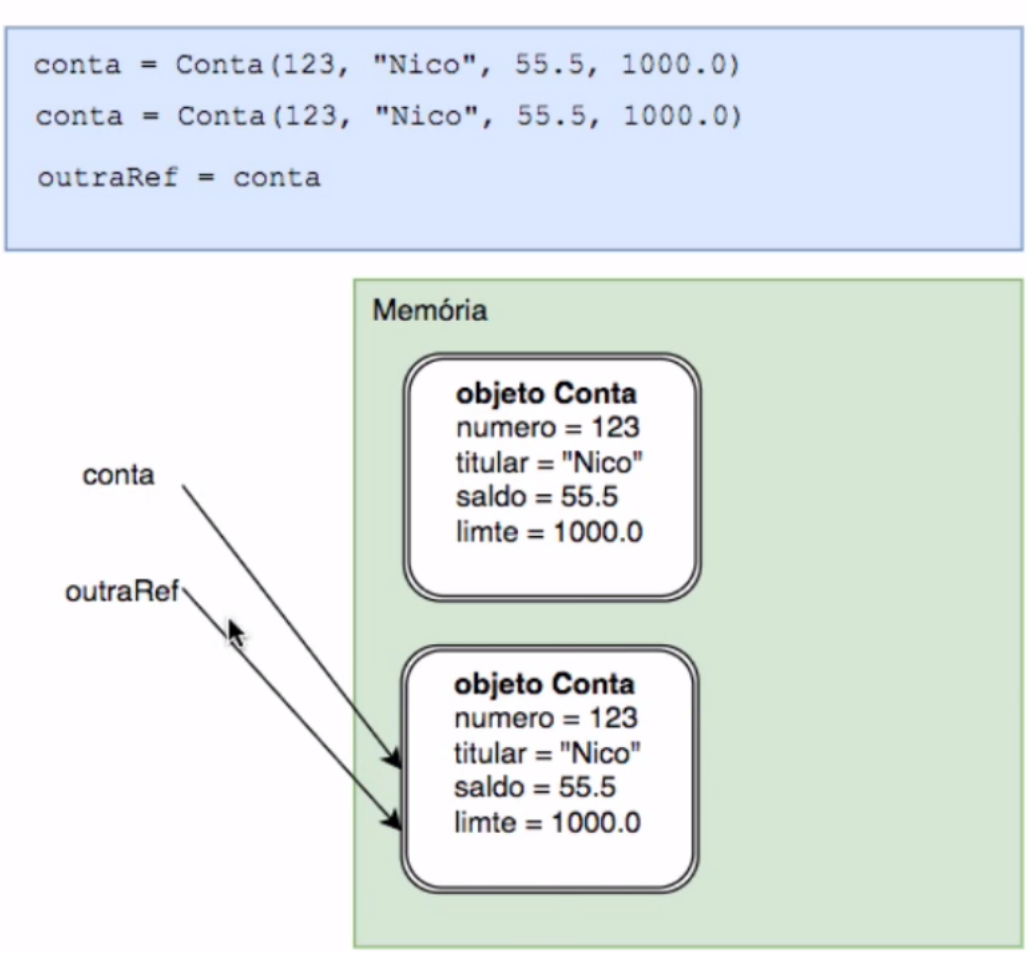
Python e Orientação a Objetos

Simularemos no diagrama a criação do segundo objeto. Novamente, o endereço será guardado dentro da referência do objeto. Agora, a variável `conta2` vai apontar para `Conta` com os dados do titular `Marco`, sendo possível invocar o método `extrato()`, que retornará os valores dos atributos relacionados a este objeto.

Novamente, teremos a variável `self`, mas dessa vez, ela apontará para o segundo objeto. Isto significa que, dependendo da referência utilizada, o `self` apontará para objetos distintos.

Na parte amarela do diagrama, em que vemos o retorno do método `extrato()`, veremos duas mensagens:

```
Saldo de 55.5 do titular Nico
Saldo de 100.0 do titular Marco
```



É útil criarmos esses desenhos incluindo as referências utilizadas no código, nós faremos mais isso adiante. Mais adiante, nos aprofundaremos no assunto encapsulamento.