

Programação Orientada a Objetos

Sobrecarga de operadores

Sobrecarga de operadores

Sobrecarga de operador é a capacidade de estender a funcionalidade dos operadores nativos (+, -, *, >, <, etc.) de forma que os mesmos funcionem também com tipos definidos pelo usuário.

O *Python* permite que as classes definidas pelo usuário implementem métodos para os operadores definidos na linguagem.

Por exemplo, para personalizar o operador de soma “+”, podemos implementar um método chamado **`__add__`**.

Sobrecarga de operadores

Operador	Método	Operação	Operador	Método	Operação
+	<code>__add__</code>	Adição	<code><=</code>	<code>__le__</code>	Menor ou igual
-	<code>__sub__</code>	Subtração	<code>>=</code>	<code>__ge__</code>	Maior ou igual
*	<code>__mul__</code>	Multiplicação	<code>==</code>	<code>__eq__</code>	Igual a
/	<code>__div__</code>	Divisão	<code>!=</code>	<code>__ne__</code>	Diferente de
//	<code>__floordiv__</code>	Divisão inteira	<code><<</code>	<code>__lshift__</code>	Deslocamento para esquerda
%	<code>__mod__</code>	Módulo	<code>>></code>	<code>__rshift__</code>	Deslocamento para direita
**	<code>__pow__</code>	Potência	<code>&</code>	<code>__and__</code>	E bit-a-bit
+	<code>__pos__</code>	Positivo	<code> </code>	<code>__or__</code>	Ou bit-a-bit
-	<code>__neg__</code>	Negativo	<code>^</code>	<code>__xor__</code>	Ou exclusivo bit-a-bit
<	<code>__lt__</code>	Menor que	<code>~</code>	<code>__inv__</code>	Inversão
>	<code>__gt__</code>	Maior que			

Sobrecarga de operadores

Se você reescrever o método “`__add__`” por exemplo, em sua classe, ao executar uma operação de adição utilizando o operador “+” com um objeto desta classe, será executado o conteúdo do método “`__add__`” que você implementou. Vamos exemplificar isto de uma forma bem simples com dois exemplos.

Sobrecarga de operadores

Exemplo 1:

```
class MinhaString(str):
    def __sub__(self, other):
        print("self:", self)
        print("other:", other)
        subtracao = self.replace(other, '')
        return f"Resultado de '{self}' - '{other}': {subtracao}"

s1 = MinhaString("Olá mundo cruel")
s2 = "cruel"

# Ao realizar a subtração s1 - s2, será removida a string s2 de dentro da string s1,
# utilizando o self.replace(s2, '')
s3 = s1 - s2
print(s3)
```

Sobrecarga de operadores

Exemplo 2:

```
class MinhaString(str):  
    def __add__(self, other):  
        print("self:", self)  
        print("other:", other)  
        return f"{self} + {other}"  
  
s1 = MinhaString("Curso Python para")  
s2 = "Todos"  
  
# Ao realizar a soma s1 + s2, será retornado o texto "s1 + s2" literalmente.  
s4 = s1 + s2  
print(s4)
```

CONTINUA...