

Sobrecarga de Operadores

quinta-feira, 27 de outubro de 2022 14:59

O comportamento de operadores em python é definido por métodos especiais, vamos alterar o comportamento de operadores com classes definidas pelo usuário.

os operadores são os + * - e os demais.

podemos então modificar o comportamento deles na classe.

Ex:

```
class retangulo():  
  
    def __init__(self,x,y):  
        self.x = x  
        self.y = y  
  
    def __add__(self, other):  
        novo_x = self.x + other.x  
        novo_y = self.y + other.y  
        return retangulo(novo_x,novo_y)  
  
r1 = retangulo(10,20)  
r2 = retangulo(10,20)  
print(r1+r2)
```

```

class retangulo():

    def __init__(self,x,y):
        self.x = x
        self.y = y

    def get_area(self):
        return self.x * self.y

    def __repr__(self):
        #pega a mensagem que seria o resultado da operação <main> (local do objeto)
        return f"<class 'Retangulo ({self.x}, {self.y})>"

    def __add__(self, other):
        novo_x = self.x + other.x
        novo_y = self.y + other.y
        return retangulo(novo_x,novo_y)

    def __lt__(self, other):
        a1 = self.get_area()
        print(a1)
        a2 = other.get_area()
        print(a2)

```

```

    def __gt__(self, other):
        a1 = self.get_area()
        a2 = other.get_area()

r1 = retangulo(10,20)
r2 = retangulo(50,20)
r3 = retangulo(20,30)
print(r1+r2)

```

self.x = recebe o primeiro parâmetro, e o other x é o segundo parâmetro.

quando usamos novo_x = está recebendo a soma dos parâmetros da primeira instância

no novo_y funciona da mesma forma.

quando usamos o repr ele vai pegar a mensagem que seria como resultado do + e transforma em uma mensagem personalizada, o __add__ está somando as duas instância com o comando print logo o resultado será a mensagem do repr com a soma dos parâmetros.

Ex:

```
C:\Users\geron\PycharmProjects\pythonProject\venv\Scripts\pyth
<class 'Retangulo (60, 40) '>

Process finished with exit code 0
```