La surcharge des opérateurs est un type de polymorphisme qui permet de redéfinir le comportement des opérateurs pour des objets de types différents. Cela permet d'écrire un code plus générique et plus lisible.

En Python, la surcharge des opérateurs est réalisée en utilisant des méthodes spéciales commençant par un double tiret bas (\_\_) et le nom de l'opérateur. Par exemple, la surcharge de l'opérateur d'addition (+) est réalisée à l'aide de la méthode spéciale \_\_add\_\_().

Voici un exemple de surcharge de l'opérateur d'addition pour la classe Point :

Python

class Point:

def \_\_init\_\_(self, x, y):

self.x = x

self.y = y

def \_\_add\_\_(self, other):

return Point(self.x + other.x, self.y + other.y)

point1 = Point(1, 2)

point2 = Point(3, 4)

print(point1 + point2) # Point(4, 6)

Dans cet exemple, la méthode spéciale \_\_add\_\_() est utilisée pour définir le comportement de l'opérateur d'addition (+) pour les objets de type Point. La méthode \_\_add\_\_() renvoie un nouvel objet Point dont les coordonnées sont la somme des coordonnées des deux points d'origine.

La surcharge des opérateurs peut être utilisée pour une grande variété de tâches. Par exemple, elle peut être utilisée pour :

* Ajouter deux objets de types différents
* Comparer deux objets de types différents
* Multiplier deux objets de types différents
* Etc.

Conclusion

La surcharge des opérateurs est un outil puissant qui peut être utilisé pour rendre le code plus générique et plus lisible. Elle permet aux programmeurs de définir un comportement spécifique pour les opérateurs pour des objets de types différents.

**Sources**

1.<https://github.com/hibana2077/Notebook> soumis à licence (MIT)