

OPERATORLARNI QAYTA YUKLASH(__ADD__, __STR__, __LEN__)

Maxmudjonova Shoxistaxon Shuxratjon qizi

Farg'ona davlat universiteti talabasi

maxmudjonovashoxista312@gmail.com

Tursunaliyeva Mohinur Zoirjon qizi

Farg'ona davlat universiteti talabasi

mohinurtursunaliyeva2907@gmail.com

Yusupov Mirsaidbek

Farg'ona davlat universiteti

Amaliy matematika va informatika

kafedra o'qituvchisi

mirsaidbeky@gmail.com

Annotatsiya: Ushbu maqolada Python dasturlash tilida operatorlarni qayta yuklash (operator overloading) tushunchasi, xususan `__add__`, `__str__`, va `__len__` maxsus metodlari orqali qanday qo'llanilishi va amaliy misoli — ikki o'lchovli Vector sinfi — batafsil tahlil qilindi. Operatorlarni qayta yuklash dasturiy ta'minotni yanada ixcham, o'qilishi oson va moslashuvchan qilishda muhim rol o'ynashi, shuningdek, matematik va ma'lumotlar strukturalari bilan ishlashda qulaylik yaratishi ko'rsatildi. Maqolada operatorlarni qayta yuklashning afzalliklari, ehtiyotkorlik bilan foydalanish lozimligi va kelajakdagi rivojlanish istiqbollari ham ko'rib chiqildi.

Kalit so'zlar: operatorlarni qayta yuklash, Python, `__add__`, `__str__`, `__len__`, obyektga yo'naltirilgan dasturlash, Vector sinfi, maxsus metodlar, operator overloading, ma'lumotlar strukturalari, dasturiy ta'minot.

Аннотация: В статье рассматривается концепция перегрузки операторов в языке программирования Python, с акцентом на специальные методы `__add__`, `__str__` и `__len__`. На примере класса двумерного вектора показано, как перегрузка операторов улучшает читаемость, гибкость и компактность кода. Обсуждаются преимущества применения перегрузки операторов при работе с математическими операциями и структурами данных, необходимость осторожного использования и перспективы дальнейшего развития данной техники.

Ключевые слова: перегрузка операторов, Python, `__add__`, `__str__`, `__len__`, объектно-ориентированное программирование, класс `Vector`, специальные методы, структуры данных, разработка программного обеспечения.

Abstract: This article explores the concept of operator overloading in Python programming language, focusing on special methods `__add__`, `__str__`, and `__len__`. The practical example of a two-dimensional `Vector` class demonstrates how operator overloading can enhance code readability, flexibility, and compactness. The study highlights the advantages of operator overloading in handling mathematical operations and data structures, discusses the need for careful implementation, and suggests future directions for expanding this technique.

Keywords: operator overloading, Python, `__add__`, `__str__`, `__len__`, object-oriented programming, `Vector` class, special methods, data structures, software development.

KIRISH

Zamonaviy dasturlash tillarida obyektga yo'naltirilgan dasturlash (OOD) konsepsiyasi muhim ahamiyatga ega bo'lib, bu uslub dasturchilarga murakkab tizimlarni yanada samarali va oson boshqariladigan tarzda yaratishga imkon beradi. Obyektlarga xos metodlar va atributlar yordamida dasturiy modellarni qurishda, operatorlarni qayta yuklash (operator overloading) tushunchasi katta o'rin tutadi. Bu imkoniyat dasturchilarga o'z sinflari uchun standart operatorlarning (masalan, `+`, `-`, `*`, `len`, `str`) qanday ishlashini o'zgartirish yoki kengaytirishga ruxsat beradi. Shunday qilib, kod ko'rinishi yanada tushunarli va tabiiyroq bo'ladi.

Operatorlarni qayta yuklash dasturlash tillarida operatorlarning standart vazifalarini yangi ma'nolar bilan boyitish imkonini beradi. Masalan, ikkita murakkab obyektни qo'shish uchun **add** metodi ishlatiladi, bu esa matematik ifodalarni sinflar kontekstida aniq va qulay ifodalashga yordam beradi. Shuningdek, **str** metodi obyektни matn ko'rinishida ifodalash imkonini yaratib, dastur ishini monitoring qilish yoki natijalarni foydalanuvchiga tushunarli tarzda ko'rsatishda zarur bo'ladi. **len** metodi esa obyektning o'lchamini yoki elementlar sonini aniqlash uchun ishlatiladi, bu esa ko'plab ma'lumotlar strukturasi sinflarida foydalidir.

Python dasturlash tilida operatorlarni qayta yuklash oson va intuitiv tarzda amalga oshiriladi. Bu tilning imkoniyatlaridan foydalanib, dasturchilar o'z sinflari uchun qulay va mos operatorlar yaratadilar. Shu bilan birga, qayta yuklangan operatorlar yordamida kodning o'qilishi va tushunilishi osonlashadi, chunki ular oddiy matematik yoki mantiqiy ifodalarga o'xshash ishlaydi. Bu xususiyat dastur tuzishda samaradorlikni oshirishga yordam beradi.

Operatorlarni qayta yuklash nafaqat matematik amallar bilan ishlashda, balki ma'lumotlar tuzilmalari, matnlar, fayllar va hatto grafik interfeys komponentlari bilan ishlashda ham keng qo'llaniladi. Masalan, ro'yxatlar yoki satrlar kabi sinflarda **len()** funksiyasi chaqirilganda **len** metodi ishlaydi, bu esa obyektning uzunligini aniq va samarali aniqlash imkonini beradi. Shu tarzda, dasturchilar o'z sinflariga mos metodlarni belgilab, kodni soddalashtirishi mumkin.

Shunday qilib, operatorlarni qayta yuklash dasturlashda kodning moslashuvchanligini oshiradi, sinflarning o'zaro aloqalarini yaxshilaydi va dasturiy mahsulotni yanada o'qilishi qulay qiladi. Ushbu mavzuni chuqurroq o'rganish orqali, dasturchilar Python tilidagi operatorlar va maxsus metodlarning imkoniyatlarini to'liq anglab, samarali va sifatli dasturlar yaratishlari mumkin bo'ladi.

ADABIYOTLAR TAHLILI

Operatorlarni qayta yuklash masalasi dasturlash metodologiyasida keng o'rganilgan va ko'plab ilmiy va amaliy manbalarda muhim mavzu sifatida ko'tarilgan. Python dasturlash tilida maxsus metodlar orqali operatorlarni qayta yuklash imkoniyati dasturchilarga sinflarni yanada moslashuvchan va tabiiy ishlashini ta'minlashga xizmat

qiladi. Misol uchun, Lutz (2013) o‘zining “Learning Python” kitobida **add**, **str**, va **len** metodlarining dasturiy modellarga qanday jozibadorlik va qulaylik olib kirishini batafsil tahlil qiladi. Unga ko‘ra, operatorlarni qayta yuklash kodning o‘qilishini osonlashtiradi va obyektlarning o‘zaro munosabatlarini yanada aniqroq ifodalashga imkon beradi.

Shuningdek, Beazley va Jones (2017) Pythonning ilg‘or xususiyatlari haqida yozgan asarlarida operatorlarni qayta yuklash texnikasining samarali qo‘llanilishi amaliy loyihalarda dasturiy interfeyslarni yaxshilashda muhim ekanligini ta’kidlaydi. Ularning fikricha, ayniqsa matematik yoki ma’lumotlar tuzilmalari bilan ishlashda **add** metodi sinflarning kengaytirilishida katta qulaylik yaratadi. Bu metod yordamida murakkab obyektlarni birlashtirish yoki qo‘shish mantiqiy va tushunarli bo‘ladi.

Bundan tashqari, Python rasmiy hujjatlarida (Python Software Foundation, 2024) operatorlarni qayta yuklashga oid maxsus metodlar keng ko‘lamda yoritilgan. **str** metodi obyektning matn ko‘rinishida ifodalashda zarur bo‘lib, ayniqsa diagnostika va foydalanuvchi interfeysi ishlab chiqishda muhim rol o‘ynaydi. **len** esa kolleksiyalar va ma’lumotlar strukturalarining o‘lchamini aniqlash uchun standart usul sifatida taqdim etilgan, bu metod orqali kod o‘qilishi va funkcionalligi sezilarli darajada yaxshilanadi.

Tadqiqotlarda, xususan Mark Lutz va David Beazley ishlarida operatorlarni qayta yuklash konsepsiyasi nafaqat kodni ixchamlashtirish, balki dasturdagi ma’no va izchillikni ta’minlash vositasi sifatida ham baholangan. Ular operatorlarni qayta yuklash dasturiy interfeyslarni yaratishda intuitiv va tabiiy muloqot imkoniyatini berishini ko‘rsatadi, bu esa dastur sifatini va uni qo‘llash qulayligini oshiradi.

Shuningdek, amaliy dasturchilar va ta’lim sohasidagi mutaxassislar (Yamamoto, 2021) operatorlarni qayta yuklash imkoniyatini o‘quv jarayonida ham qo‘llashni tavsiya qiladilar. Bu nafaqat talabalarni dasturlashning chuqur konseptlariga ega bo‘lishlariga yordam beradi, balki ularning kodni yanada o‘qilishi oson va sodda yozish ko‘nikmalarini rivojlantiradi. Bu fikrlar operatorlarni qayta yuklash konsepsiyasining ta’lim va amaliyotda qanchalik muhim ekanligini ko‘rsatadi.

ASOSIY QISM

Operatorlarni qayta yuklash dasturlashda obyektlar orasidagi muloqotni tabiiy va intuitiv qilish uchun muhim vositadir. Python dasturlash tilida bu imkoniyat maxsus

metodlar yordamida ta'minlanadi. Masalan, **add** metodi yordamida ikkita obyektning "+" operatori orqali qo'shish mumkin bo'ladi. Bu nafaqat kodni ixchamlashtiradi, balki murakkab ma'lumotlar strukturasi bilan ishlashda qulaylik yaratadi. Masalan, matnlarni, ro'yxatlarni yoki maxsus sinflarni qo'shishda **add** operatorining qayta yuklanishi kodning mantiqiy va tushunarli bo'lishini ta'minlaydi.

str metodi esa obyektning matn ko'rinishida ifodalash uchun javobgardir. Bu metod qayta yuklanganda, obyektning foydalanuvchi yoki dasturchi uchun qulay va o'qilishi oson bo'lgan ko'rinishi yaratiladi. Misol uchun, ma'lumotlarni konsolga chiqarishda yoki log fayllarga yozishda **str** metodi yordamida obyektning mazmuni tushunarli tarzda aks ettiriladi. Bu dasturdagi xatoliklarni aniqlash va dastur holatini tushunishni sezilarli darajada osonlashtiradi.

len metodi esa obyektning uzunligini yoki o'lchamini aniqlash uchun ishlatiladi. Bu metod ko'pincha kolleksiya yoki ma'lumotlar tuzilmalari bilan ishlashda qo'llaniladi. Masalan, ro'yxat, qator yoki maxsus sinfdagi elementlar sonini aniqlashda **len** metodi orqali kodni standart va oson tushuniladigan ko'rinishda yozish mumkin. Operatorni qayta yuklash bu metodning mos obyektlarda ishlashini ta'minlab, kodni yanada moslashuvchan qiladi.

Umuman olganda, operatorlarni qayta yuklash dasturiy ta'minotni yanada ixcham, tushunarli va kengaytiriladigan qiladi. Bu metodlar yordamida dasturchilar o'z sinflarini Pythonning ichki operatorlari bilan bir xil darajada ishlashini ta'minlay oladi. Natijada, kod nafaqat samarali, balki boshqa dasturchilar uchun ham oson tushunarli va qo'llanilishi qulay bo'ladi. Shu bilan birga, operatorlarni qayta yuklash dasturlashda ob'ektga yo'naltirilgan paradigmaning to'liq imkoniyatlaridan samarali foydalanishni ta'minlaydi.

Masala:

Vector sinfi ikki o'lchamli vektorlarni ifodalaydi.

- `__add__` operatori yordamida ikki vektor qo'shilishi kerak (har bir koordinata bo'yicha yig'indi).
- `__str__` metodi vektorni (x, y) ko'rinishida chiqaradi.
- `__len__` metodi vektor uzunligini (modulini) butun qism sifatida qaytaradi.

Sinfga tegishli metodlarni to'liq yozing va sinovdan o'tkazing.

Masala yechimining python dasturlash tilidagi kodi:

```
import math
```

```
class Vector:
```

```
    def __init__(self, x, y):
```

```
        self.x = x
```

```
        self.y = y
```

```
    def __add__(self, other):
```

```
        if not isinstance(other, Vector):
```

```
            raise TypeError("Faqat Vector obyektlari qo'shilishi mumkin")
```

```
        return Vector(self.x + other.x, self.y + other.y)
```

```
    def __str__(self):
```

```
        return f"({self.x}, {self.y})"
```

```
    def __len__(self):
```

```
        length = math.sqrt(self.x ** 2 + self.y ** 2)
```

```
        return int(length) # Vektor uzunligining butun qismini qaytaradi
```

```
# Sinov kodi:
```

```
v1 = Vector(3, 4)
```

```
v2 = Vector(1, 2)
```

```
print("v1:", v1)          # (3, 4)
```

```
print("v2:", v2)          # (1, 2)
```

```
v3 = v1 + v2
```

```
print("v1 + v2 =", v3)    # (4, 6)
```

```
print("v1 uzunligi:", len(v1)) # 5 (chunki  $\sqrt{3^2 + 4^2} = 5$ )
```

```
print("v3 uzunligi:", len(v3)) #  $\text{int}(\sqrt{4^2 + 6^2}) = \text{int}(\sqrt{16+36}) = \text{int}(7.21) = 7$ 
```

Kodning natijasi:

```
v1: (3, 4)
v2: (1, 2)
v1 + v2 = (4, 6)
v1 uzunligi: 5
v3 uzunligi: 7
```

Ushbu dastur Python dasturlash tilida operatorlarni qayta yuklash (operator overloading) mavzusini amalda ko‘rsatadi. Dasturda “Vector” nomli sinf yaratilgan bo‘lib, u ikki o‘lchovli vektorlarni ifodalaydi. Sinfga ikki xususiyat — x va y koordinatalari berilgan. Operatorlarni qayta yuklash yordamida vektorlar ustida oddiy matematik amallarni bajarish mumkin.

“**add**” metodi yordamida ikkita vektorning mos koordinatalari yig‘iladi va yangi vektor hosil qilinadi. Bu metod “+” operatorining sinf uchun qayta aniqlangan ko‘rinishi bo‘lib, ikki obyektни qo‘shishda avtomatik chaqiriladi. Masalan, (3,4) va (1,2) vektorlarini qo‘shganda natija (4,6) vektori olinadi.

“**str**” metodi vektor obyektini matn ko‘rinishida — x va y qiymatlari qavs ichida ajratilgan holda — chiqarishga xizmat qiladi. Bu metod `print()` funksiyasi chaqirilganda avtomatik ishlaydi va vektorlarni o‘qilishi oson formatda ko‘rsatadi.

“**len**” metodi esa vektor uzunligini (modulini) hisoblaydi. Bu yerda uzunlik formula orqali aniqlanadi: kvadrat ildiz ostida x^2 va y^2 yig‘indisi. Natija butun son ko‘rinishida qaytariladi, bu metod “`len()`” funksiyasi chaqirilganda ishlaydi.

Dasturda uchta vektor yaratilib, ularning yig‘indisi va uzunliklari hisoblanadi. Bu orqali operatorlarni qayta yuklash qanday ishlashi va sinf obyekt bilan qanday amallar bajarilishi amaliy ko‘rinishda namoyon bo‘ladi.

Umuman olganda, operatorlarni qayta yuklash sinflarga o‘ziga xos matematik va mantiqiy xatti-harakatlar berish imkonini yaratadi. Bu kodni soddalashtiradi, o‘qishni osonlashtiradi va murakkab amallarni intuitiv tarzda bajarishga yordam beradi. Ushbu usul ayniqsa matematik hisob-kitoblar, fizika modellashtirish va kompyuter grafikasi kabi sohalarda juda foydalidir.

NATIJA, MUHOKAMA, TAKLIF

Ushbu dasturiy yechimda operatorlarni qayta yuklash (operator overloading) konsepsiyasining Python tilida qanday samarali qo'llanishi amaliy tarzda ko'rsatildi. "Vector" sinfi misolida **add**, **str** va **len** maxsus metodlari yordamida ikki o'lchovli vektorlarni qo'shish, matn ko'rinishida chiqarish va uzunligini hisoblash funksiyalari to'liq amalga oshirildi. Sinf metodlari aniq, sodda va intuitiv tuzilgan bo'lib, dastur kodining o'qilishi va tushunilishini sezilarli darajada yengillashtirdi.

Natijalarga ko'ra, operatorlarni qayta yuklash yordamida obyektlar ustida murakkab matematik va mantiqiy amallarni an'anaviy operatorlar bilan bajarish mumkinligi dasturga yanada moslashuvchanlik va qulaylik beradi. Masalan, ikkita vektorni qo'shish uchun oddiygina "+" operatoridan foydalanish, sinf ichidagi murakkab kodlarni yashirish bilan birga, foydalanuvchiga tabiiy va oson tushuniladigan interfeys yaratadi. Bu kodni boshqa dasturchilar bilan bo'lishishda yoki keyinchalik qo'llashda ham katta afzallik hisoblanadi.

str metodining qayta yuklanishi foydalanuvchiga obyektning matn shaklida qulay ko'rinishda taqdim etadi, bu esa dastur natijalarini monitoring qilish va tahlil qilishda muhimdir. Ayniqsa, dasturdagi xatoliklarni aniqlash yoki foydalanuvchi interfeysini yaratishda aniq va o'qilishi oson ma'lumot taqdim etilishi katta yordam beradi.

len metodining qayta yuklanishi esa dasturdagi obyektlarning o'lchami yoki uzunligini oddiy "len()" funksiyasi orqali aniqlash imkonini beradi. Bu yondashuv ma'lumotlar strukturalari bilan ishlashda qulaylik yaratadi va kodning ixchamligini oshiradi.

Muhokama qilinadigan yana bir jihat shundaki, operatorlarni qayta yuklashdan ehtiyotkorlik bilan foydalanish kerak. Agar ular noto'g'ri yoki mantiqsiz tarzda qayta yuklansa, kod murakkab va tushunarsiz bo'lib qolishi mumkin. Shuning uchun operatorlarning asl ma'nolariga mos ravishda ishlatish va metodlarni to'g'ri loyihalash muhimdir. Masalan, qo'shish operatori faqat mantiqan qo'shilishi mumkin bo'lgan obyektlar uchun qayta yuklanishi kerak.

Taklif sifatida, operatorlarni qayta yuklash imkoniyatlarini yanada kengroq qo'llash uchun sinflarda boshqa ko'plab operatorlarni ham qayta yuklash tavsiya qilinadi. Masalan, **sub** (ayirish), **mul** (ko'paytirish), **eq** (tenglik) kabi metodlar sinf funksionalligini oshiradi va dasturiy interfeysni yanada boyitadi. Shuningdek, foydalanuvchining ehtiyojiga qarab,

maxsus metodlar yordamida murakkab matematik amallar yoki ma'lumotlar strukturalari ustida oson ishlash imkonini yaratish mumkin.

XULOSA

Operatorlarni qayta yuklash obyektga yo'naltirilgan dasturlashning ajralmas qismi bo'lib, Python tilida sinflarning funksionalligini kengaytirish va kodni yanada tabiiy, o'qilishi qulay shaklga keltirishda muhim rol o'ynaydi. **add**, **str**, va **len** kabi maxsus metodlar orqali dasturchilar murakkab amallarni sodda operatorlar yordamida amalga oshirishi mumkin, bu esa dasturiy ta'minotning samaradorligi va sifatini oshiradi.

Tajriba shuni ko'rsatadiki, operatorlarni qayta yuklash yordamida yaratilgan sinflar nafaqat matematik yoki mantiqiy operatsiyalarni to'g'ri va aniq bajaradi, balki foydalanuvchi interfeysini yaxshilash va dastur holatini tushunishni osonlashtirish imkonini beradi. Shu bilan birga, operatorlarni noto'g'ri qo'llash dastur kodining murakkablashishiga olib kelishi mumkinligi sababli, ularni ehtiyotkorlik bilan loyihalash zarur.

Kelajakda operatorlarni qayta yuklash texnikasini yanada kengaytirish va murakkab tizimlarda samarali qo'llash uchun yangi metodlarni joriy etish, shuningdek, dasturchilarning bu konsepsiyaga oid bilimlarini chuqurlashtirish muhim ahamiyat kasb etadi. Bunday yondashuv dasturiy mahsulotlarning sifatini va qo'llanilish qulayligini yanada oshiradi hamda rivojlanayotgan dasturlash paradigmalari bilan uyg'unlashishga yordam beradi.

Foydalanilgan adabiyotlar ro'yxati:

1. **Lutz, M. (2013).** Learning Python (5th Edition). O'Reilly Media.
— Python dasturlash tilining asoslari va maxsus metodlar, operatorlarni qayta yuklash haqida batafsil ma'lumot.
2. **Beazley, D., & Jones, B. (2017).** Python Cookbook (3rd Edition). O'Reilly Media.
— Amaliy Python misollari, operatorlarni qayta yuklash va murakkab dastur interfeyslarini yaratish bo'yicha tavsiyalar.
3. **Python Software Foundation. (2024).** Python 3.12 Documentation.
<https://docs.python.org/3/reference/datamodel.html#special-method-names>

— Python rasmiy hujjatlari, maxsus metodlar va operatorlarni qayta yuklash bo'yicha standartlar.

4. **Yamamoto, K. (2021).** "Teaching Object-Oriented Programming Concepts with Operator Overloading in Python." *Journal of Computer Science Education*, 14(2), 85-97.

— Operatorlarni qayta yuklashni ta'lim jarayonida qo'llash va o'quvchilar uchun samaradorligi haqida tadqiqot.

5. **Sweigart, A. (2015).** *Automate the Boring Stuff with Python*. No Starch Press.

— Operatorlarni qayta yuklashga oid amaliy misollar va Python dasturlash tilidagi oson tushunarli tushunchalar.

6. **Hetland, M. L. (2005).** *Beginning Python: From Novice to Professional*. Apress.

— Python dasturlash tilida obyektga yo'naltirilgan dasturlash va maxsus metodlar haqida asosiy tushunchalar.

7. **Summerfield, M. (2010).** *Programming in Python 3*. Addison-Wesley.— Python 3 ning obyektga yo'naltirilgan imkoniyatlari, operatorlarni qayta yuklash metodlari haqida batafsil tavsif.

8. **Van Rossum, G., & Drake, F. L. (2009).** *The Python Language Reference Manual*. Python Software Foundation.— Python tilining rasmiy til tavsifi, operatorlarni qayta yuklash mexanizmi haqida ma'lumotlar.

9. **Downey, A. (2015).** *Think Python: How to Think Like a Computer Scientist* (2nd Edition). Green Tea Press.— Obyektga yo'naltirilgan dasturlash asoslari va operatorlarni qayta yuklashga kirish.

10. **Gosling, J., Joy, B., Steele, G., & Bracha, G. (2014).** *The Java Language Specification* (Java bilan solishtirish uchun). Addison-Wesley.
— Operatorlarni qayta yuklash konsepsiyasi va obyektga yo'naltirilgan dasturlash paradigmasini boshqa tillarda ham ko'rib chiqish uchun.