

**PYTHON DASTURLASH TILIDA LAMBDA IFODALARI,
KONVERTATSIYA VA O'ZGARUVCHILAR DOIRASI**

Tojimamatov Israiljon Nurmamatovich

Farg'ona davlat universiteti

israiltojimatov@gmail.com

Muxsinova Sevinchxon Ikromjon qizi

Farg'ona davlat universiteti

akramovasevinchxon08@gmail.com

Annotatsiya: Maqolada Python dasturlash tilidagi lambda ifodalari, ma'lumot turlarini o'zgartirish (konvertatsiya) va o'zgaruvchilar doirasi kabi tushunchalarning nazariy asoslari tahlil qilingan. Ushbu tushunchalar dastur funkcionalligini oshirishda, kodni qisqartirishda va xatolarning oldini olishda muhim ahamiyatga ega. Maqolada ularning o'zaro bog'liqligi va dastur loyihalashdagi roli ochib beriladi.

Kalit so'zlar: Python, lambda ifodasi, konvertatsiya, o'zgaruvchilar doirasi, dasturlash tushunchalari.

Аннотация: В статье рассматриваются теоретические основы лямбда-выражений, преобразования типов и области видимости переменных в языке программирования Python. Эти понятия играют важную роль в улучшении функциональности программ, сокращении кода и предотвращении ошибок. В работе также раскрываются взаимосвязи между этими концепциями и их значение при проектировании программ.

Ключевые слова: Python, лямбда-выражения, преобразование типов, область видимости переменных, программные концепции.

Abstract: This article analyzes the theoretical foundations of lambda expressions, type conversion, and variable scope in the Python programming language. These concepts are essential for improving program functionality,



reducing code length, and minimizing errors. The paper also explores their interrelation and importance in software design.

Keywords: *Python, lambda expressions, type conversion, variable scope, programming concepts.*

Kirish

Python dasturlash tili o'zining soddaligi, tushunarli sintaksisi va keng imkoniyatlari bilan zamonaviy dasturlash tillari orasida yetakchi o'rinlardan birini egallagan. U nafaqat dasturlash asoslarini o'rgatuvchi vosita, balki sun'iy intellekt, veb-ishlanmalar, ilmiy hisob-kitoblar va ma'lumotlar tahlili kabi yo'nalishlarda keng qo'llanilmoqda.

Ushbu maqola Python dasturlash tilidagi muhim tushunchalar – lambda ifodalari, ma'lumot turlarini o'zgartirish (konvertatsiya) va o'zgaruvchilar doirasi (scope) nazariy asosda o'rganilib, ular dastur ishlashiga qanday ta'sir qilishi va ulardan qanday foydalanish lozimligi haqida tahliliy ma'lumot beradi.

Metodologiya

Ushbu ilmiy ish nazariy tahlil uslubida bajarildi. Python dasturlash tilining rasmiy hujjatlari, yetakchi mualliflarning darsliklari va o'quv adabiyotlari asosida kontseptual yondashuv ishlab chiqildi. Har bir tushuncha haqida ilmiy-texnik manbalardan olingan asosiy ta'riflar, ularning o'zaro aloqasi va dasturga ta'siri nazariy jihatdan o'rganildi.

Analitik tahlil qilish uchun quyidagi manbalar asos qilib olindi:

- Python rasmiy hujjatlari (<https://docs.python.org/3>),
- Mark Lutzning "Learning Python",
- Allen Downey'ning "Think Python",
- va boshqa metodik qo'llanmalar.

Natijalar

Tahlillar shuni ko'rsatdiki:

1. Lambda ifodalari — Python dasturlash tilida nomlanmagan (anonim) funksiyalarni yaratish uchun ishlatiladi. Ular ayniqsa filtratsiya, saralash va xaritalash



jarayonlarida foydalidir. Lambda funksiyalar funksional dasturlashga oid yondashuvlar bilan yaqin bog'liq.

2. Konvertatsiya (tip o'zgarishi) — bu Python dasturida bir turdagi ma'lumotni boshqasiga o'tkazish vositasi hisoblanadi. U ikki turga bo'linadi: aniq (foydalanuvchi tomonidan bajariladigan) va noaniq (Python tomonidan avtomatik bajariladigan). Bu jarayon ma'lumotlarni qayta ishlashda muhim o'rin tutadi.

3. O'zgaruvchilar doirasi (scope) — bu o'zgaruvchining dasturdagi mavjudlik maydonini bildiradi. Python tilida LEGB printsipli asosida lokal, enclosing, global va built-in doiralar aniqlanadi. Har bir daraja o'ziga xos qoidalarga ega bo'lib, noto'g'ri ishlatilganda xatolik yuzaga kelishi mumkin.

Bu natijalar Python dasturlash tilidagi mazkur tushunchalar bir-biri bilan uzviy bog'liqligini ko'rsatdi hamda ularni chuqur tushunish dasturchining samaradorligiga bevosita ta'sir qilishini isbotladi.

Maqolada ko'rib chiqilgan lambda ifodalari, konvertatsiya va o'zgaruvchilar doirasi tushunchalarini amaliy misollar yordamida yanada aniqroq tushunish mumkin.

Birinchidan, lambda ifodasi yordamida sonning kvadratini hisoblovchi kichik anonim funksiya yaratildi. Bu funksiya argument sifatida qabul qilgan sonni o'zining kvadratiga aylantiradi. Misol uchun, son 5 bo'lsa, natija 25 ga teng bo'ladi.

Ikkinchidan, konvertatsiya jarayoni ko'rsatildi. Matn ko'rinishida berilgan raqamli qiymat ("10") butun son turiga o'tkazilib, dasturda matematik amallarda qo'llanishi ta'minlandi.

Uchinchidan, o'zgaruvchilar doirasi tushunchasiga oid misol orqali global va lokal o'zgaruvchilar qanday farq qilishi va ular dasturda qanday ishlashi ifodalandi. Dasturda global o'zgaruvchi qiymati 10 bo'lsa, funksiyaning ichida e'lon qilingan lokal o'zgaruvchi 5 qiymatiga ega bo'lib, ular bir-biridan mustaqil bo'lishi namoyon bo'ldi.

Quyida ushbu amaliy tushunchalarning Python kodi keltirilgan:

```
# Lambda ifodasi: sonning kvadratini hisoblash  
kvadrat = lambda x: x ** 2
```



```
natija = kvadrat(5) # natija = 25
# Konvertatsiya: matn ko'rinishidagi sonni butun songa o'tkazish
matn_son = "10"
raqam = int(matn_son) # '10' matnini 10 soniga aylantirish
# O'zgaruvchilar doirasi misoli
x = 10 # global o'zgaruvchi
def func():
    x = 5 # lokal o'zgaruvchi
    return x
natija_func = func() # natija_func = 5
natija_global = x # global o'zgaruvchi qiymati 10
```

Ushbu misollar bo'yicha **tushuntirish:**

- Lambda ifodasi yordamida kvadrat funksiyasi qisqa va ixcham yozildi.
- int() funksiyasi matnni raqamga (konvertatsiya) aylantirishda ishlatildi.
- O'zgaruvchilar doirasi misolida x nomli o'zgaruvchi dasturda ikki xil doirada mavjudligi ko'rsatildi: global va lokal.

Ushbu misollar mavzuni amalda tushunish va dasturlashda qo'llashda yordam beradi hamda Python dasturlash tilining asosiy tushunchalarini mustahkamlaydi.

Munozara

Yuqoridagi natijalar asosida quyidagi xulosalarni chiqarish mumkin:

- Lambda ifodalari funksiyalarni qisqa yozishda juda foydali bo'lishiga qaramay, ularni murakkab mantiqli kodlarda ishlatish tavsiya etilmaydi. Bu holatda kodning tushunarliyligi pasayishi mumkin.
- Konvertatsiya jarayonlarini to'g'ri tushunmaslik, ayniqsa turli tipdagi ma'lumotlar bilan ishlayotganda, noto'g'ri hisob-kitoblarga olib keladi. Shuning uchun Python dasturida ma'lumot turlariga e'tiborli bo'lish muhim.
- O'zgaruvchilar doirasi bilan bog'liq xatolar dasturda kutilmagan natijalarni yuzaga keltiradi. Ayniqsa global va lokal o'zgaruvchilar bilan ishlayotganda aniqlik zarur.



Shuningdek, dastur samaradorligi va ishonchliligi bevosita bu tushunchalarning to'g'ri qo'llanilishiga bog'liqdir. Dasturlashni endi o'rganayotganlar bu nazariy tushunchalarni mukammal o'zlashtirish orqali amaliyotda xatolardan holi, tushunarli va samarali kodlar yaratishlari mumkin.

Xulosa

Python dasturlash tilida lambda ifodalari, konvertatsiya va o'zgaruvchilar doirasi kabi konseptlar dasturchilar uchun nazariy va amaliy jihatdan muhim ahamiyatga ega. Mazkur maqolada bu tushunchalar IMRAD uslubida nazariy jihatdan tahlil qilinib, ularning mohiyati, qo'llanishi va dastur samaradorligiga ta'siri yoritildi.

Kelgusidagi izlanishlarda bu tushunchalarning murakkab holatlarda (masalan, klasslar, dekoratorlar, yoki modul tizimida) qanday ishlashini chuqur o'rganish tavsiya etiladi.

ADABIYOTLAR RO'YXATI

1. Sobirov A., Tursunov M. «Python dasturlash tilida funksiyalar va ularning xususiyatlari». — Samarqand: Samarqand Davlat Universiteti Nashriyoti, 2022. — 180 bet.
2. Ахмадалиев Ш., Каримова Н. «Python dasturlash tili asoslari». — Toshkent: O'zbekiston Milliy Universiteti nashriyoti, 2023. — 220 bet. (Shoxrux Ahmadaliyev, Nigora Karimova. *Python dasturlash tili asoslari*. Toshkent, 2023)
3. Python rasmiy hujjatlari. <https://docs.python.org/3/>
4. Lutz M. *Learning Python*. O'Reilly Media, 2013.
5. Downey A. *Think Python: How to Think Like a Computer Scientist*.
6. Sweigart A. *Automate the Boring Stuff with Python*.
7. Zelle J. *Python Programming: An Introduction to Computer Science*.