

MANTIQUIY OPERATSIYALAR VA SHARTLI IFODALARDAN FOYDALANISH

Tojimamatov Israiljon Nurmamatovich

Farg'ona davlat universiteti katta o'qituvchisi

israiltojimatov@gmail.com

Abduvahobova Ozoda Begali qizi

Farg'ona davlat universiteti 3 – kurs talabasi

ozodaabduvahobova7@gmail.com

Annotatsiya: Ushbu mavzuda Python dasturlash tilida mantiqiy operatsiyalar (and, or, not) va shartli ifodalar (if, elif, else) yordamida dastur oqimini boshqarish usullari ko'rib chiqiladi. Mantiqiy operatorlar orqali shartlarni tekshirish va ular asosida turli qarorlar qabul qilish mexanizmlari tushuntiriladi. Amaliy misollar orqali foydalanuvchi kiritmalariga javob qaytarish, turli holatlarni aniqlash va dastur kodining mos ravishda ishlashini ta'minlash usullari yoritiladi.

Kalit so'zlar: Python, mantiqiy operatorlar, and, or, not, if, elif, else, shartli ifodalar, qaror qabul qilish, kod oqimi.

Annotation: This topic explores the use of logical operations (and, or, not) and conditional expressions (if, elif, else) in the Python programming language. It explains how these structures are used to control program flow and make decisions based on conditions. Practical examples demonstrate how to handle user input, evaluate different scenarios, and execute code accordingly.

Keywords: Python, logical operators, and, or, not, if, elif, else, conditional expressions, decision-making, control flow.

Аннотация: В данной теме рассматривается использование логических операций (and, or, not) и условных выражений (if, elif, else) в языке программирования Python. Объясняется, как с помощью этих конструкций управлять потоком выполнения программы и принимать решения на основе условий. Приводятся практические примеры обработки пользовательского ввода и реализации различных сценариев поведения программы.

Ключевые слова: Python, логические операторы, and, or, not, if, elif, else, условные выражения, принятие решений, поток выполнения.

Kirish. Zamonaviy dasturlash tillari, algoritmlar va mantiqiy muammolarni yechishda mantiqiy operatsiyalar va shartli ifodalar muhim o'rin egallaydi. Ular kompyuterga qaror qabul qilish, turli holatlarni tekshirish, shartli natijalarni chiqarish imkonini beradi. Ushbu maqolada mantiqiy operatsiyalar va shartli ifodalarning mohiyati, turlari va ular yordamida masalalarni qanday yechish mumkinligi haqida

ma'lumot beriladi. Har qanday dasturiy ta'minot foydalanuvchining harakatlariga yoki tizim holatiga qarab turli yo'nalishda ishlashi mumkin. Masalan, foydalanuvchi yoshini kiritganda, tizim "voyaga yetgan" yoki "voyaga yetmagan" deb chiqarishi kerak bo'ladi. Bunday shartli qarorlarni dasturda amalga oshirish uchun mantiqiy operatsiyalar va shartli ifodalar ishlatiladi.

Mantiqiy operatsiyalar — berilgan hadlari va natijasi mulohaza (fikr) dan iborat amallar. Berilgan hadlar soniga qarab mantiqiy amallar bir o'rinli, ikki o'rinli va h.k. deb yuritiladi. Bir o'rinli mantiqiy amallar soni to'rtta: berilgan fikrdan qat'i nazar natijasi doim chin (aynan haqiqat) amal, natijasi doim yolg'on (aynan yolg'on) amal, natijasi berilgan fikr bilan mos tushadigan amal va nihoyat, berilgan fikr chin bo'lsa, natijasi yolg'on, berilgan fikr yolg'on bo'lsa, natijasi chin bo'ladigan amal. So'nggi mantiqiy amal bir o'rinli mantiqiy amallardan eng muhimi bo'lib, u inkor amal deyiladi. A fikrning inkori $\sim A$ kabi belgilanib, "A emas" deb o'qiladi. Mas, 1 Oy sayyora — "Oy sayyora emas", ($2*2=4$) — ikki karra ikki to'rt emas. Ikkilik kodda yozilgan mashina so'zlari ustida Mantiqiy amallar mos razryadlar bo'yicha bajarilib, i o'rniga 1, 1 o'rniga 0 olinadi, matn shakliga aylantiriladi va ma'lumot ko'rinishida chiqish qurilmasiga beriladi. Mantiq-informatsion mashina tez ishlashi, "xotira" hajmining kattaligi bilan oddiy hisoblash mashinalaridan farq qiladi. Mantiq-informatsion mashina ilmiy tadqiqot natijalarini ishlash, adabiyot topishni avtomatlashtirish, sanoat, qishloq xo'jaligi va transportga oid statistik ma'lumotlarni, davolash muassasalarida bemorlarni kuzatishdan olingan natijalarni, meteorologik, seysmologik stansiyalardan, Yer sun'iy yo'ldoshlaridan olingan ma'lumotni ishlash va tarjima ishlarida qo'llaniladi.

Protsessor tarkibidagi arifmetik-mantiqiy qurilmaning ishlash prinsipini tushunish uchun avval insonning mantiqiy fikrlash va xulosa chiqarish usullarini ko'rib chiqamiz. Insonlar kundalik hayotda o'zaro muloqot qilish uchun turli mulohazalardan foydalanishadi. Ma'lumki, mulohaza – narsa yoki hodisalarning xususiyatini anglatuvchi darak gapdir. Boshqacha aytganda, mulohaza – rost yoki yolg'onligi haqida so'z yuritish mumkin bo'lgan darak gap. Mulohazalar sodda va murakkab bo'lishi mumkin. Biror shart yoki usul bilan bog'lanmagan hamda faqat bir holatni ifodalovchi mulohazalar sodda mulohazalar deyiladi. Sodda mulohazalar ustida amallar bajarib, murakkab mulohazalarni hosil qilish mumkin. Odatda murakkab mulohazalar sodda mulohazalardan "VA", "YOKI" kabi bog'lovchilar, "EMAS" shaklidagi ko'makchilar yordamida tuziladi.

Mulohazalarni lotin alifbosi harflari bilan belgilash (masalan, A= "Bugun havo issiq") qabul qilingan. Har bir mulohaza faqat ikkita: "rost" yoki "yolg'on" mantiqiy qiymatga ega bo'lishi mumkin. Qulaylik uchun "rost" qiymatni 1 raqami bilan, "yolg'on" qiymatni esa 0 raqami bilan belgilab olamiz. A va B sodda mulohazalar bir paytda rost bo'lgandagina rost bo'ladigan yangi (murakkab) mulohazani hosil qilish

amali mantiqiy ko'paytirish amali deb ataladi. Bu amalni konyunksiya (lotincha: conjunctio– bog'layman) deb ham atashadi. Mantiqiy ko'paytirish amali ikki yoki undan ortiq sodda mulohazalarni “VA” bog'lovchisi bilan bog'laydi hamda “A va B” , “A and B” , “A $\wedge$ B” , “A·B” kabi ko'rinishda yoziladi. Mantiqiy ko'paytirishni ifodalaydigan quyidagi jadval rostlik jadvali deb ataladi: A va B mulohazalarning kamida bittasi rost bo'lganda rost bo'ladigan yangi murakkab mulohazani hosil qilish amali mantiqiy qo'shish amali deb ataladi. Bu amalni diziunksiya (lotincha: disjunctio – ajrataman) deb ham atashadi Mantiqiy qo'shish amali ikki yoki undan ortiq sodda mulohazalarni “YOKI” bog'lovchisi bilan bog'laydi hamda va “A yoki B”, “A or B” , “A $\vee$ B” , “A + B” kabi ko'rinishlarda yoziladi.

Mantiqiy qo'shish amalining rostlik jadvali quyidagicha: A mulohaza rost bo'lganda yolg'on, yolg'on bo'lganda esa rost qiymat oladigan mulohaza hosil qilish amali mantiqiy inkor amali deb ataladi. Bu amalni inversiya (lotincha: Inversio – to'ntaraman) deb ham atashadi Mantiqiy inkor amali “A EMAS” , “not A” , “ $\neg$ A” , “~” ko'rinishlarda yoziladi. Mantiqiy inkor amalining rostlik jadvali quyidagicha: Ko'rinish turibdiki, mantiqiy o'zgaruvchilar, munosabatlar, mantiqiy amallar va qavslar yordamida mantiqiy ifodalar hosil qilish mumkin ekan. Mantiqiy ifodalarda mantiqiy amallar quyidagi tartibda bajariladi: inkor ($\neg$), mantiqiy ko'paytirish ($\wedge$), mantiqiy qo'shish ($\vee$). Teng kuchli yoki bir xil amallar ketma-ketligi bajarilayotganda amallar chapdan o'ngga qarab tartib bilan bajariladi, ifodada qavslar ishtirok etganda dastlab qavslar ichidagi amallar bajariladi. Ichma-ich joylashgan qavslarda eng ichkaridagi qavs ichidagi amallar eng avval bajariladi.

Shartli ifodalar— bu dasturlashda yoki mantiqiy fikrlashda ma'lum bir shart bajarilishiga qarab harakat yoki natija tanlanadigan ifodalardir. Ular odatda: rost (true) yoki yolg'on (false) qiymatga ega bo'ladigan mantiqiy ifodalar asosida ishlaydi. “Agar biror narsa rost bo'lsa – bu ishni qil, aks holda boshqa ishni qil” degan mantiqda ishlaydi.

1.if Operator: if operatori shartni tekshiradi va agar shart true bo'lsa, tegishli blokni bajaradi.

```
if(shart) {
    //shart true bo'lsa bajariladigan kod
}
```

2.else Operator: else operatori if sharti false bo'lsa bajariladigan kodni belgilash uchun ishlatiladi.

```
if (shart) {
    // shart true bo'lsa bajariladigan kod
} else {
    // shart false bo'lsa bajariladigan kod
}
```

3. else if Operator: else if operatori bir nechta shartlarni ketma-ket tekshirish uchun ishlatiladi. Agar if sharti false bo'lsa, else if sharti tekshiriladi va u true bo'lsa tegishli kod bloki bajariladi.

4. switch Operator: switch operatorining ishlash logikasi yuqorida muhokama qilingan ichma-ich joylashgan if operatorlari konstruksiyasini eslatadi. Biroq, prinsipial farq shundaki, switch operatorida tekshirilayotgan ifoda faqat butun son yoki belgili qiymatni qaytarishi mumkin, lekin true yoki false kabi mantiqiy qiymatlarni emas. switch operatori bir o'zgaruvchining qiymatini bir nechta ehtimoliy qiymatlar bilan solishtirib, tegishli kod blokini bajaradi. Bu operator bir nechta else if dan foydalanishdan ko'ra qulayroq bo'lishi mumkin.

```
switch (o'zgaruvchi) {
    case qiymat1:
        // o'zgaruvchi qiymat1 ga teng bo'lsa bajariladigan kod
        break;
    case qiymat2:
        // o'zgaruvchi qiymat2 ga teng bo'lsa bajariladigan kod
        break;
    // qo'shimcha case bloklari
    default:
        // o'zgaruvchi hech bir qiymatga teng bo'lmasa bajariladigan kod
}
```

switch operatori bajarilganida, dumaloq qavs ichidagi ifoda qiymati hisoblanadi. Keyin bu qiymat har bir case blokining boshida keltirilgan qiymatlar bilan ketma-ketlikda, yuqoridan pastga qarab, taqqoslanadi. Moslik topilishi bilan, mos keluvchi blokning buyruqlari bajariladi. switch operatorining nozik jihati shundaki, moslik topilgandan so'ng, switch operatorining oxirigacha bo'lgan barcha buyruqlar, shu jumladan pastda joylashgan case bloklaridagi buyruqlar ham bajariladi. Agar faqat bitta blokning buyruqlari bajarilishini xohlasangiz, bu blokni break operatori bilan yakunlash kerak. Operatorni default bloki bilan yakunlash mumkin. Ushbu blokning buyruqlari faqat nazorat qiymatlari bilan hech qanday moslik topilmasa bajariladi. default blok butun tuzilmani yakunlaydigan blok bo'lgani sababli, unda break operatori ishlatilmaydi.

Xulosa:

Mantiqiy operatorlar va shartli ifodalar dasturlashning eng muhim va ko'p qo'llaniladigan elementlaridandir. Ular kompyuter dasturiga tanlov qilish, qaror qabul qilish va turli holatlarga mos harakatni bajarish imkonini beradi. Dasturchi biror muammoni hal qilishda foydalanuvchining kiritgan ma'lumotlarini, dastur holatini yoki tashqi muhitdagi shartlarni tekshirib, ular asosida dasturning keyingi yo'nalishini belgilaydi. Bu esa dasturga moslashuvchanlik va intellektual yondashuvni beradi. Mantiqiy operatorlar (masalan, AND, OR, NOT, XOR) yordamida bir nechta shartlar

birgalikda tahlil qilinadi, bu esa yanada murakkab, real hayotga yaqin shartlarni ifodalash imkonini yaratadi. Masalan, foydalanuvchining yoshi va ro'yxatdan o'tganligi tekshirilib, unga muayyan ruxsat berilishi mumkin. Aynan shunday amaliy hollarda mantiqiy operatorlar va shartli ifodalar o'z kuchini namoyon etadi. Shartli ifodalar (if, elif, else) esa ushbu mantiqiy tekshiruvlar asosida bajarilishi kerak bo'lgan amallarni aniqlaydi. Ular dastur oqimini nazorat qilishda beqiyos vositadir. Har qanday interaktiv, foydalanuvchi bilan ishlaydigan yoki ma'lumotlarga asoslangan dasturda shartli ifodalarsiz ishlash deyarli imkonsizdir. Xulosa qilib aytganda, dastur kodining ishlash mantiqini belgilovchi asosiy vosita sifatida mantiqiy operatorlar va shartli ifodalarni to'g'ri tushunish va ulardan samarali foydalanish — har bir dasturchi yoki texnologiya bilan shug'ullanuvchi shaxs uchun zarur ko'nikmadir. Ushbu bilimlar nafaqat dastur yozishda, balki kundalik hayotdagi muammolarni tahlil qilish va yechimini topishda ham mantiqiy yondashuvni kuchaytiradi.

Foydalanilgan adabiyotlar:

1. Informatik ava axborot texnologiyalari: umumiy o'rta ta'lim maktablarining 9-sinflar uchun darslik M. R. Faysiyeva, D. M. Sayfurov, N. S. Xaytullayeva – Toshkent: Tasvir, 2020 – 112b
2. Boltayev B., Azamatov A., Asqarov A., Sodiqov M., Azamatova G. Informatika va hisoblash texnikasi asoslari. Umumiy o'rta ta'lim maktablarining 9-sinfi uchun darslik. Toshkent: "Cho'lpon" nomidagi NMIU, 2015. – 160 b.
3. Chris Roffey. Computer science. Programming book for Python. – USA: Cambridge university press. 2017, – p. 204
4. Chris Roffey. Python basics. Coding club. Level 1,2. – USA: Cambridge university press. 2012, – p. 85
5. Eric Matthes. Python crash course: a hands-on, project-based introduction to programming. – San-Francisco: No Starch Press, 2015. – p. 562
6. Matt Harrison. Illustrated guide to Python 3. 2017, – p. 267
7. Dan Bader. Python tricks the book. Anja Pircher Design, 2017, – p. 299
8. Jamie Chan. Learn python in one day and learn it well. – p. 132