# OBYEKTGA YO'NALTIRILGAN DASTURLASH PRINSIPLARI.

**Yusupov Mirsaidjon**
Amaliy matematika va informatika
kafedrasi katta o'qituvchisi
mirsaidbeky@gmail.com
**Ikromaliyeva Oynisa Islomjon qizi**
Farg'ona davlat uiversiteti 3-kurs talabasi
oynisaikromaliyeva050@gmail.com

**Annotatsiya:** Ushbu maqolada obyektga yo'naltirilgan dasturlash (OOP) prinsiplari nazariy va amaliy jihatdan tahlil qilinadi. Python dasturlash tilida inkapsulyatsiya, merosxo'rlik, polimorfizm va abstraksiya tamoyillari misol tariqasida yoritiladi. Mazkur tamoyillar dasturiy tizimlarni modullarga ajratish, kodni qayta ishlatish, tizimning kengaytiriluvchanligini oshirish va murakkab loyihalarni boshqarishni soddalashtirishga imkon beradi. Shuningdek, maqolada Python dasturlarida OOP tamoyillarining real loyihalarda qo'llanilishi va ularning afzallik hamda kamchiliklari tahlil qilinadi. Tadqiqot natijalari OOPni zamonaviy dasturlash jarayonida samarali qo'llash va Python dasturlarini sifatli ishlab chiqishda muhim ekanligini ko'rsatadi.

**Kalit so'zlar:** ob'yektga yo'naltirilgan dasturlash, python dasturlash tili, inkapsulyatsiya, vorislik, polimorfizm, abstraksiya, masturiy ta'minot arxitekturasi, modullik va qayta ishlatish

**Abstract:** This article provides a theoretical and practical analysis of object-oriented programming (OOP) principles. The principles of encapsulation, inheritance, polymorphism, and abstraction are illustrated using the Python programming language. These principles enable modular software design, code reusability, system scalability, and simplification of managing complex projects. Additionally, the article examines the application of OOP principles in real Python projects and analyzes their advantages and limitations. The results of this study demonstrate the importance of effective OOP implementation in modern programming and in developing high-quality Python software.

**Keywords:** object-oriented programming, python programming language, encapsulation, inheritance, polymorphism, abstraction, software architecture, modularity and reusability

**Аннотация:** В данной статье проводится теоретический и практический анализ принципов объектно-ориентированного программирования (ООП). Принципы инкапсуляции, наследования, полиморфизма и абстракции иллюстрируются на примере языка программирования Python. Эти принципы позволяют создавать модульные программные системы, повторно использовать код, повышать масштабируемость системы и упрощать управление сложными проектами. Также рассматривается применение принципов ООП в реальных проектах на Python и анализируются их преимущества и ограничения. Результаты исследования показывают важность эффективного применения ООП в современном программировании и при разработке качественного программного обеспечения на Python.

**Ключевые слова:** объектно-ориентированное программирование, язык программирования python, инкапсуляция, наследование, полиморфизм, абстракция, архитектура программного обеспечения, модульность и повторное использование

**Kirish.** Zamonaviy axborot texnologiyalarining jadal rivojlanishi dasturiy ta’minot yaratish jarayoniga tubdan yangicha yondashuvlarni shakllantirdi. Kun sayin murakkablashib borayotgan axborot tizimlari, katta hajmdagi ma’lumotlar, sun’iy intellekt va avtomatlashtirilgan boshqaruv tizimlarining keng qo‘llanilishi dasturlash texnologiyalari oldiga yuqori darajadagi ishonchlilik, moslashuvchanlik va kengaytiriluvchanlik talablarini qo‘ymoqda. Bunday sharoitda obyektga yo‘naltirilgan dasturlash (OOP) paradigmalarining o‘rni alohida ahamiyat kasb etadi. Chunki OOP murakkab tizimlarni modullarga ajratish, ularni real obyektlar sifatida tasvirlash hamda dastur kodini boshqarish va rivojlantirish jarayonini sezilarli darajada soddalashtiradi.Obyektga yo‘naltirilgan dasturlashning shakllanishi 1960–1970-yillarda Simula-67 va Smalltalk dasturlash tillaridan boshlangan bo‘lsada, bugungi kunda C#, Java, Python, Kotlin, C++ kabi aksariyat zamonaviy dasturlash tillari ushbu paradigma asosiga qurilgan. Shu bois OOP nafaqat dastulashning yo‘nalishi, balki dasturiy arxitekturani

loyihalash, modellashtirish, tizimlararo bog‘liqlikni tashkil etish va katta hajmli dasturiy loyihalarning sifatini oshirishda asosiy metodologiya sifatida qaraladi.

Mavzuning dolzarbligi shundaki, hozirgi IT bozorida ishlab chiqilayotgan loyihalarning mutlaq ko‘pchiligi OOP tamoyillarini qo‘llagan holda yaratiladi. Xususan, bank tizimlari, ta’lim platformalari, mobil ilovalar, onlayn xizmatlar, sun’iy intellekt modullari, avtomatlashtirilgan boshqaruv tizimlari va o‘yin dvijoklari OOP konseptlarisiz samarali ishlamaydi. Komponentlarni qayta ishlatish, kodni modullashtirish va murakkab tizimlarni boshqarish imkoniyati aynan OOPning afzalliklaridan kelib chiqadi.

Mazkur maqolaning maqsadi — obyektga yo‘naltirilgan dasturlashning asosiy tamoyillarini (inkapsulyatsiya, vorislik, polimorfizm, abstraksiya) chuqur o‘rganish, ularning nazariy mohiyatini ochib berish hamda amaliy dasturiy tizimlarda qo‘llanilishiga tahliliy yondashuvni amalga oshirishdan iborat.

Tadqiqotning ilmiy yangiligi shunda ko‘rinadiki, maqolada OOP tamoyillari nafaqat nazariy nuqtai nazardan, balki amaliy dasturiy tizimlar misolida tahlil qilinadi, ularning kuchli va zaif jihatlari solishtiriladi hamda zamonaviy dasturlash jarayonidagi o‘rni keng qamrovda yoritiladi.

Tadqiqotning amaliy ahamiyati esa shundan iboratki, OOP tamoyillarining to‘g‘ri qo‘llanilishi dasturiy ta’minotning sifatini, barqarorligini va ishlash samaradorligini sezilarli darajada oshiradi. Bu esa dasturchilarga yirik loyihalarda xatoliklarni kamaytirish, kodni qayta ishlatish va tizimni kengaytirish imkoniyatini beradi.Shu tariqa, obyektga yo‘naltirilgan dasturlash prinsiplari nafaqat dasturchilar uchun muhim bilim manbai, balki zamonaviy dasturiy ta’minotning poydevori hisoblanadi. Ushbu maqolada OOP tamoyillarining nazariy asoslari va amaliy qo‘llanilishi ilmiy tahlil asosida keng yoritiladi. Zamonaviy axborot texnologiyalarining jadal rivojlanishi dasturiy ta’minot yaratish jarayoniga tubdan yangicha yondashuvlarni shakllantirdi. Kun sayin murakkablashib borayotgan axborot tizimlari, katta hajmdagi ma’lumotlar, sun’iy intellekt va avtomatlashtirilgan boshqaruv tizimlarining keng qo‘llanilishi dasturlash texnologiyalari oldiga yuqori darajadagi ishonchlilik, moslashuvchanlik va kengaytiriluvchanlik talablarini qo‘ymoqda. Bunday sharoitda obyektga yo‘naltirilgan dasturlash (OOP) paradigmalarining o‘rni alohida ahamiyat kasb etadi. Chunki OOP murakkab tizimlarni

modullarga ajratish, ularni real obyektlar sifatida tasvirlash hamda dastur kodini boshqarish va rivojlantirish jarayonini sezilarli darajada soddalashtiradi.Obyektga yo‘naltirilgan dasturlashning shakllanishi 1960–1970-yillarda Simula-67 va Smalltalk dasturlash tillaridan boshlangan bo‘lsa-da, bugungi kunda C#, Java, Python, Kotlin, C++ kabi aksariyat zamonaviy dasturlash tillari ushbu paradigma asosiga qurilgan. Shu bois OOP nafaqat dastulashning yo‘nalishi, balki dasturiy arxitekturani loyihalash, modellashtirish, tizimlararo bog‘liqlikni tashkil etish va katta hajmli dasturiy loyihalarning sifatini oshirishda asosiy metodologiya sifatida qaraladi.

Mavzuning dolzarbligi shundaki, hozirgi IT bozorida ishlab chiqilayotgan loyihalarning mutlaq ko‘pchiligi OOP tamoyillarini qo‘llagan holda yaratiladi. Xususan, bank tizimlari, ta’lim platformalari, mobil ilovalar, onlayn xizmatlar, sun’iy intellekt modullari, avtomatlashtirilgan boshqaruv tizimlari va o‘yin dvijoklari OOP konseptlarisiz samarali ishlamaydi. Komponentlarni qayta ishlatish, kodni modullashtirish va murakkab tizimlarni boshqarish imkoniyati aynan OOPning afzalliklaridan kelib chiqadi.

Mazkur maqolaning maqsadi — obyektga yo‘naltirilgan dasturlashning asosiy tamoyillarini (inkapsulyatsiya, vorislik, polimorfizm, abstraksiya) chuqur o‘rganish, ularning nazariy mohiyatini ochib berish hamda amaliy dasturiy tizimlarda qo‘llanilishiga tahliliy yondashuvni amalga oshirishdan iborat.

**Obyektga yo‘naltirilgan dasturlashning nazariy asoslari.**Obyektga yo‘naltirilgan dasturlash — bu real hayotdagi obyektlarni kompyuter modeli sifatida tasvirlashga asoslangan konseptual paradigma bo‘lib, u tizimlarni modullarga bo‘lish, ularning o‘zaro aloqasini soddalashtirish va dasturiy ta’minotning umumiy sifatini oshirishni maqsad qilgan. OOPning shakllanishi dasturiy ta’minot muhandisligi tarixining eng muhim bosqichlaridan biridir. Chunki u prosedural dasturlashdagi asosiy muammolar — kodning takrorlanishi, tizimning murakkab tuzilishi va modullarning bir-biriga kuchli bog‘liqligini bartaraf etadi.OOPning nazariy asoslari “obyekt”, “sinf”, “interfeys”, “metod”, “xususiyat” kabi fundamental tushunchalarga tayangan. Obyekt — holat va xatti-harakatni o‘zida mujassam etuvchi mustaqil moduldur. sinf esa shu obyektlarni yaratish uchun andaza rolini bajaradi. Ushbu model inson tafakkuriga juda yaqin bo‘lgani uchun, dasturchilar murakkab tizimlarni ham sodda mantiqiy birliklar orqali boshqarish

imkoniyatiga ega bo‘ladi.Zamonaviy dasturlash tillari OOPning sifatli arxitekturani yaratishdagi rolini tasdiqlagan. Misol uchun, C# tilidagi .NET platformasi, Java tilidagi Spring Framework, Pythonning Django yoki Flask platformalari OOP tamoyillari asosida shakllangan. Bu esa OOPning nafaqat nazariy, balki amaliy jihatdan ham yuqori samaradorlikka ega ekanini ko‘rsatadi.

Obyektga yo‘naltirilgan dasturlash to‘rt asosiy tamoyilga tayanadi: inkapsulyatsiya, vorislik, polimorfizm va abstraksiya. Ushbu tamoyillar birgalikda tashqi muhitdan mustaqil, xavfsiz, kengaytiriladigan va boshqarilishi oson bo‘lgan tizimlarni yaratish imkonini beradi. Quyida har bir tamoyil boy nazariy tushuntirish, amaliy misollar va real loyihalardagi tahlillar bilan keng yoritiladi.

Inkapsulyatsiya— obyektning ichki tuzilishini tashqi muhitdan yashirish, unga faqat maxsus interfeyslar orqali murojaat qilish tamoyilidir. Bu tamoyil dastur xavfsizligini oshirish va xatolik ehtimolini kamaytirishda asosiy ahamiyatga ega.Inkapsulyatsiya yordamida obyektning holatini faqat kerakli joylarda o‘zgartirishga ruxsat beriladi. Bu esa tizimni himoyalangan, tartibli va boshqarilishi oson qiluvchi mexanizmdir. Masalan, C# dasturlash tilida private, protected, public kabi modifikatorlar inkapsulyatsiyaning amaliy vositasidir.Katta hajmli bank tizimlarida mijoz balansini to‘g‘ridan-to‘g‘ri o‘zgartirish taqiqlanadi. Buning o‘rniga maxsus metodlar orqali tranzaksiya amalga oshiriladi. Bu inkapsulyatsiya tamoyilining amaliy ko‘rinishidir.

Vorisliklik — mavjud sinf asosida yangi sinf yaratish tamoyilidir. Bu dasturchiga kodni takrorlamasdan, funksionallikni kengaytirish imkonini beradi.Vorislik dastur strukturasi ichida iyerarxik bog‘lanishlarni yaratadi. Masalan, “Hayvon” nomli umumiy sinfdan “Mushuk”, “It”, “Ot” kabi ixtisoslashgan sinflar hosil qilish mumkin. Bunda umumiy xususiyatlar ota sinfda, maxsuslari esa farzand sinflarda saqlanadi.Lekin vorislik noto‘g‘ri qo‘llansa, tizim haddan tashqari murakkab bo‘lib ketadi. Shu sababli zamonaviy arxitekturada “Vorisdan ko‘ra kompozitsiya ustun” tamoyili keng qo‘llaniladi.

Polimorfizm obyektlarning bir xil nomdagi metodlarga turlicha javob qaytarishi tamoyilidir. Bu tizimni moslashuvchan, imkoniyatlari keng va o‘zgartirishga qulay holga keltiradi. Statik (overloading) — kompilyatsiya vaqtida aniqlanadi.Dinamik (overriding) — bajarilish vaqtida ishlaydi.Bu tamoyil dasturga “bir interfeys — ko‘p amalga oshirish”

imkonini beradi.Masalan, “Chizish()” metodi turli shakllarda turlicha natija beradi: doira uchun bir xil, kvadrat uchun boshqa, uchburchak uchun boshqacha. Shunga qaramay, hammasi bir xil metod nomi orqali chaqiriladi.Polimorfizm OOP arxitekturasining eng kuchli tamoyillaridan biri bo‘lib, plugin tizimlar, modul dasturlash va xizmatlarga asoslangan arxitekturalarda keng qo‘llaniladi.

Abstraksiya — obyektning ortiqcha xususiyat va funksiyalarini yashirib, faqat muhim jihatlarini ko‘rsatish tamoyilidir.Abstraksiya yordamida dasturchi real hayotdagi murakkab jarayonlarni soddalashtirilgan model ko‘rinishiga keltiradi. Bu dasturiy tizimni tushunishni va boshqarishni osonlashtiradi.Masalan, mobil telefon foydalanuvchiga faqat interfeysni ko‘rsatadi, ichidagi millionlab jarayonlar esa yashirilgan. OOPda ham sinf foydalanuvchiga kerakli metodlarni ko‘rsatib, ichki logika yashiriladi.

**Python dasturida Obyektga yo‘naltirilgan dasturlash prinsiplari.**Obyektga yo‘naltirilgan dasturlash (OOP) zamonaviy dasturlash paradigmalari orasida eng qulay va keng qo‘llaniladigan yondashuv hisoblanadi. Python tili to‘laqonli obyektga yo‘naltirilgan til bo‘lib, unda har bir narsa — raqamlar, satrlar, ro‘yxatlar va hatto funksiyalar ham obyekt sifatida qaraladi. Shu sababli Python dasturchilarga kodni strukturaviy, modulli va qayta ishlatishga qulay shaklda tashkil etish imkonini beradi. OOPning asosiy tamoyillari inkapsulyatsiya, vorislik, polimorfizm va abstraksiyadir.

Python dasturida inkapsulyatsiya ma’lumotlarning tashqi aralashuvdan himoya qilinishini bildiradi. Masalan, bank hisobini yaratishda balans qiymati private atribut sifatida berilishi mumkin, shunda foydalanuvchi uni faqat maxsus metodlar orqali o‘zgartira oladi. Shu tarzda obyektning ichki holati himoyalanadi va xatolik ehtimoli kamayadi.

Vorislik tamoyili yordamida bitta sinfdan boshqa sinflar hosil qilinadi. Masalan, “Animal” umumiy sinfidan “Dog” va “Cat” sinflari meros oladi va o‘ziga xos metodlarni qayta aniqlaydi. Bu kodni qayta ishlatishga, tizimni modullarga ajratishga va loyihani kengaytirishga imkon beradi. Shu bilan birga, noto‘g‘ri vorislik tizimni murakkablashtirishi mumkin.

Polimorfizm tamoyili esa obyektlarning bir xil nomdagi metodlarga turlicha javob berishini bildiradi. Masalan, “sound()” metodi turli hayvon sinflari uchun turlicha natija

beradi: mushuk uchun “Meow”, it uchun “Woof”. Bu yondashuv tizimning moslashuvchanligini oshiradi va kengaytiriluvchanlikni ta’minlaydi.Abstraksiya esa ortiqcha tafsilotlarni yashirib, faqat muhim funksiyalarni taqdim etish tamoyilidir. Python’da bu abstrakt sinflar yordamida amalga oshiriladi. Misol uchun, “Shape” nomli abstrakt sinfdan voris bo‘lgan “Circle” sinfi faqat o‘zining maydonini hisoblaydigan metodni amalga oshiradi. Shu tarzda murakkab tizimlarning ichki logikasi foydalanuvchidan yashiriladi, faqat kerakli interfeys taqdim etiladi.Python tilida OOPning amaliy ahamiyati juda katta. Django va Flask kabi web platformalar, Pygame o‘yin dvijoklari, TensorFlow va Scikit-learn kabi sun’iy intellekt kutubxonalari OOP tamoyillari asosida ishlaydi. OOP dasturiy loyihani modullarga ajratishga, kodni qayta ishlatishga, testlashni osonlashtirishga va tizimni kengaytirish imkonini beradi. Shu sababli Python dasturchilari OOP konseptlarini chuqur o‘rganishlari zarur.Tahliliy nuqtai nazardan qaraganda, Python dasturida OOPdan foydalanishning afzalliklari quyidagilardan iborat: loyihani boshqarish soddalashadi, kodni qayta ishlatish imkoniyati oshadi, modullik kuchayadi va dastur kengaytirilishiga moslashadi. Biroq, kichik dasturlar uchun ortiqcha murakkablik yuzaga kelishi, vorislikning noto‘g‘ri ishlatilishi va abstraktsiya noto‘g‘ri qo‘llanilganda tushunarlilikni kamaytirishi mumkin.Shunday qilib, Python dasturida obyektga yo‘naltirilgan dasturlash prinsiplari dasturiy ta’minot yaratishda samarali yondashuvni ta’minlaydi va zamonaviy dasturlashning ajralmas qismidir.

**Xulosa.**

Obyektga yo‘naltirilgan dasturlash (OOP) prinsiplari zamonaviy dasturlash jarayonida asosiy rol o‘ynaydi. Python dasturlash tilida OOPning to‘rtta asosiy tamoyili — inkapsulyatsiya, vorislik, polimorfizm va abstraksiya — dasturiy tizimlarni samarali, modulga bo‘lingan va boshqarilishi oson holatda yaratishga imkon beradi. Inkapsulyatsiya ma’lumotlarni tashqi aralashuvdan himoya qilib, xatolik ehtimolini kamaytiradi; vorislik kodni qayta ishlatish va loyihani kengaytirishga yordam beradi; polimorfizm tizimning moslashuvchanligi va dinamik xatti-harakatlarini ta’minlaydi; abstraksiya esa murakkab tizimlarni soddalashtirib, foydalanuvchi uchun tushunarli interfeyslar yaratadi.

Python misolida OOP tamoyillari real loyihalarda keng qo‘llaniladi: Django va Flask web-ilovalari, Pygame o‘yinlari, TensorFlow va Scikit-learn kabi sun’iy intellekt

kutubxonalari shular jumlasidandir. Shu bilan birga, OOPning noto‘g‘ri qo‘llanilishi tizimni murakkablashtirishi va tushunarliligini kamaytirishi mumkin. Shuning uchun har bir tamoyilni to‘g‘ri va maqsadga muvofiq qo‘llash muhimdir.

Umuman olganda, OOP tamoyillari dasturchilarga murakkab dasturiy tizimlarni modullarga ajratish, kodni qayta ishlatish va kengaytirish imkonini beradi. Shu bois, obyektga yo‘naltirilgan dasturlashni chuqur o‘rganish va Python kabi zamonaviy dasturlash tillarida uni to‘g‘ri qo‘llash dasturiy ta’minot sifatini oshirishda muhim ahamiyatga ega. Mazkur maqola tahliliy nuqtai nazardan OOPning nazariy asoslari va amaliy qo‘llanilishining samaradorligini ochib berdi hamda Python dasturida ushbu tamoyillarni qo‘llashning amaliy jihatlarini yoritdi.

**Foydalanilgan adabiyotlar:**

1. Lutz, M. Learning Python, 5th Edition. O’Reilly Media, 2013.
2. Sweigart, A. Automate the Boring Stuff with Python, No Starch Press, 2019.
3. Gamma, E., Helm, R., Johnson, R., Vlissides, J. Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994.
4. Albahari, J., Albahari, B. C# 8.0 in a Nutshell, O’Reilly Media, 2020.
5. Beazley, D., Jones, B. Python Cookbook, 3rd Edition, O’Reilly Media, 2013.
6. Larman, C. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd Edition, Prentice Hall, 2004.
7. McKinney, W. Python for Data Analysis, 2nd Edition, O’Reilly Media, 2017.
8. Python Software Foundation. Python Documentation, https://docs.python.org/3/, 2025.
9. Zakas, N. Understanding Object-Oriented JavaScript, No Starch Press, 2020.
10. Martin, R.C. Clean Architecture: A Craftsman’s Guide to Software Structure and Design, Prentice Hall, 2017.