

## OBJECT ORIENTED PROGRAMMING(OOP) TUSHUNCHASI BILIMLARINI DASTURLASHNI O'RGANISHDAGI O'RNI MUHOKAMASI

**Muxtarov Farrux Muxammadovich**

PhD, Toshkent Axborot Texnologiya Universiteti Farg'ona Filiali Farg'ona(O'zbekiston)

**Musojonov Xumoyun Muhammadayubxon o'g'li**

Toshkent Axborot Texnologiya Universiteti Farg'ona Filiali , Magistranti,  
Farg'ona(O'zbekiston)

**Tojiddinov Azizbek Ilhomjonov o'g'li**

Toshkent Axborot Texnologiya Universiteti Farg'ona Filiali , Talaba, Farg'ona(O'zbekiston)

<https://doi.org/10.5281/zenodo.7298775>

*Annotatsiya.* Ushbu maqolada object oriented programming(OOP) tushunchasi bilimlarini dasturlashni o'rganishdagi o'rnini ko'rib chiqiladi. Ko'pchilik uchun bu bosqich juda murakkab va tushunarsizdek ko'rinadi. Ushbu maqolada shunday mulohaza va muammolarni shu mavzuga doir namuna va ma'lumotlar bilan hal qilish masalalarini ko'rib chiqiladi.

*Kalit so'zlar:* object oriented programming(OOP), class , obyekt , inkapsulatsiya , polimorfizm.

## ОБСУЖДЕНИЕ РОЛИ КОНЦЕПЦИИ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ (ООП) В ОБУЧЕНИИ ЗНАНИЯМ ПО ПРОГРАММИРОВАНИЮ

*Аннотация.* В данной статье рассматривается роль концепции объектно-ориентированного программирования (ООП) в изучении знаний по программированию. Многим этот этап кажется очень сложным и непонятным. В данной статье рассматриваются вопросы решения таких соображений и задач с примерами и данными по данной теме.

*Ключевые слова:* объектно-ориентированное программирование (ООП), класс, объект, инкапсуляция, полиморфизм.

## DISCUSSION OF THE ROLE OF THE CONCEPT OF OBJECT-ORIENTED PROGRAMMING (OOP) IN TEACHING KNOWLEDGE IN PROGRAMMING

*Abstract.* This article considers the place of the concept of object oriented programming (OOP) in learning programming knowledge. For many people, this stage seems very complicated and incomprehensible. In this article, the issues of solving such considerations and problems with examples and data on this topic are considered.

*Keywords:* object oriented programming (OOP), class, object, encapsulation, polymorphism.

### KIRISH

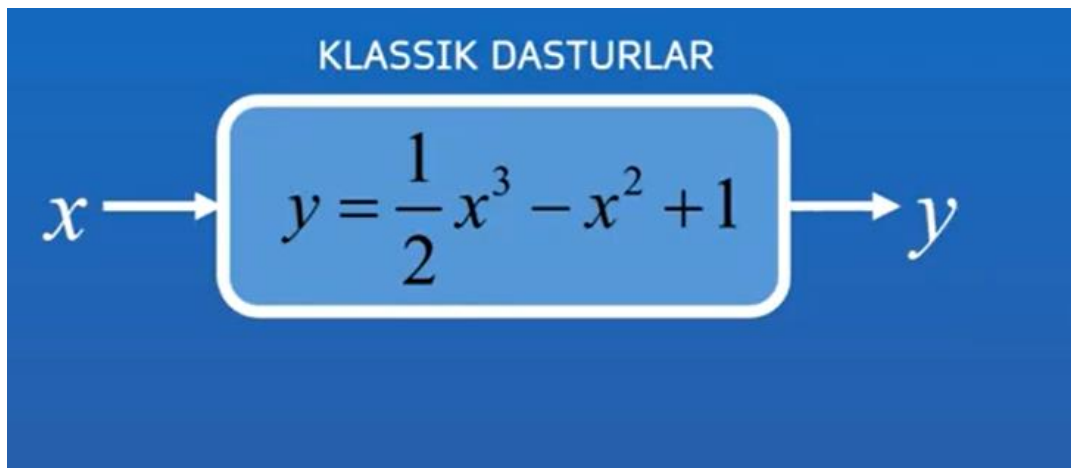
Kompyuter texnikasi va axborot tizimlarining iqtisodda, boshqarishda, aloqada, ilmiy tadqiqotlarda, ta'limda, xizmat ko'rsatish sohasida, tijorat, moliya va inson faoliyatining boshqa sohalarida qo'llanilishining rivoji axborotlashtirish va, umuman, jamiyat rivojini belgilovchi yo'nalish hisoblanadi. Kompyuter texnikasining qo'llanishi evaziga erishiluvchi samara axborot ishlanishi ko'lamining oshishi bilan ortib boradi. Ushbu texnikaning qo'llanish sohalarini va ko'lami uning ishlashining ishonchligi va barqarorligi muammolari bilan bir qatorda unda aylanuvchi axborot xavfsizligini ta'minlash muammosini tug'diradi.

Yani har bir kompyuter texnikasi va uning axborot tizimlari qandaydir dasturiy taminot yoki qandaydir dasturiy muhitda ishlaydi. Va bu dasturiy taminot yoki muhitlarni biz asosiy qo'big'ini yoki asosi xaqida so'zlashamiz.

## TADQIQOT MATERIALLARI VA METODOLOGIYASI

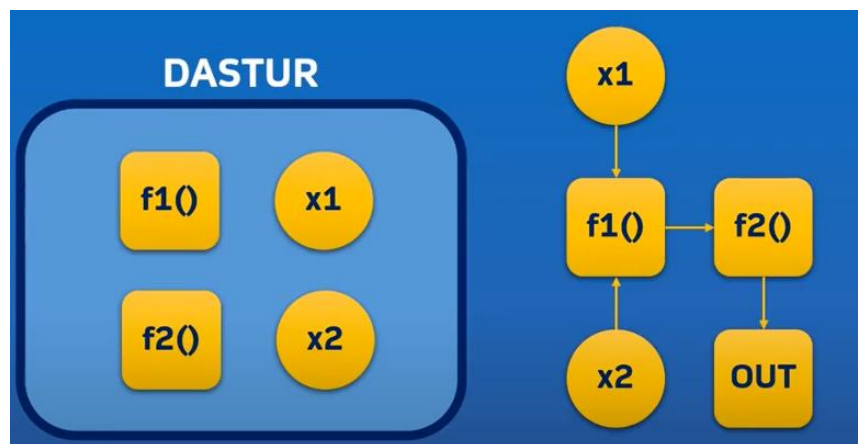
Object oriented programming(OOP) daturlashni tushinish uchun avval klassik dasturlashni ko'rib chiqaylik. Chunki ilk kompyuterlar va dasturlar matematik muammolarni xal qilishga qaratilgan bunday dasturlar foydalanuvchidan malumotlarni qabul qilib olgan va qat'iy ketma-ketlik yani tartibga amal qilgan xolda turli arifmetik amallarni bajargan va dastur so'ngida foydalanuchi kutgan natijani qaytargan(1-rasm). Chuning uchun ham bunday dasturlar chiziqli(tartibli) dasturlar deb ataladi.

1-rasm.



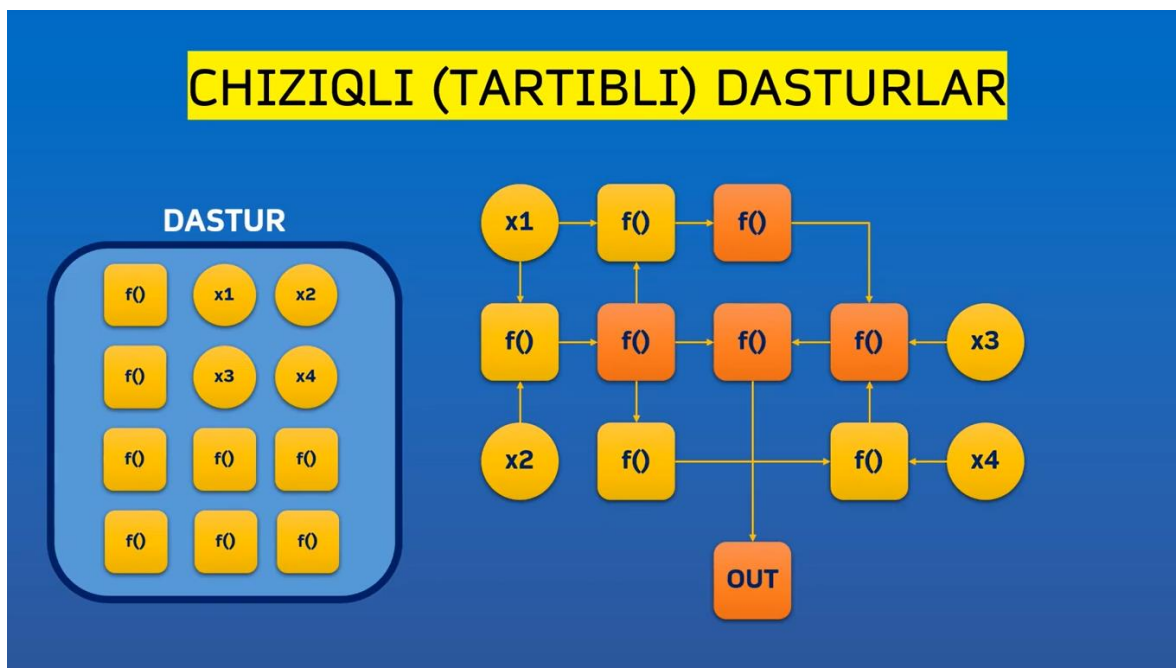
Albatta agar biz dasturlashga endi qadam qo'yayotgan bo'lsak mana shunday chiziqli dasturlarni yozishdan boshlaymiz. Bizning dasturimiz bir nechta o'zgaruvchilar va funksiyalardan iborat bo'ladi. Bu funksiyalar va o'zgaruvchilar bir-biri bilan munosabatga kiradi va dastur yakunida biz kutgan natijani beradi.

2-rasm.



Biroq dastur kattalashgani sari o'zgaruvchilar va funksiyalar soni oshib boradi. Ular o'rtasidagi aloqa ham chigallashib , kodlarimiz murakkab va tushinishga qiyin bo'lib ketadi(3-rasm). Misol uchun:

3-rasm.



Lekin dasturlash davomida 1ta funksiyaga o'zgartirish kiritsak unga bog'liq boshqa funksiyalarni ishdan chiqshiga va dasturimizni xato natija berishiga olib kelishi mumkin.

## TADQIQOT NATIJALARI

Yuqoridagi keltirilgan sabablar tufayli biz chiziqli dasturlashni afzalliklari va kamchiliklarini muhokama qilamiz. Afzalliklari: dasturlashni o'rganish uchun qulay, Sodda va tushnarli kod, dastur algoritmini kuzatish oson va dastur xotirada kamroq joy egallaydi. Kamchiliklari: Murakkab dasturlarni chiziqli usulda yozish qiyin yoki ilojisiz, bir dastur uchun yozilgan koddan boshqa dasturda qayta foydalanib bo'lmaydi, dastur ichidagi malumotlar(o'zgaruvchilar) barcha funksiyalar uchun ochiq va zamonaviy dasturlar chiziqli emas. Albatta vaqt o'tib dasturlarga qo'yilgan talablar murakkablashib borganligi tufayli chiziqli dasturlash tamoyili talabga javob bermay qo'ydi va 1970-yillarda OBJECT ORIENTED PROGRAMMING(OOP) tamoyili olg'a surila boshladi.

Object Oriented Programming(OOP) dasturlashda o'zaro bog'liq bo'lgan funksiyalar va o'zgaruvchilar 1ta konteynerga jamlanadi va bunday konteynerlar obyekt deb ataladi. 1 obyektga tegishli bo'lgan o'zgaruvchilar uning xususiyatlari, unga tegishli funksiyalar esa metodlari deb ataladi. Misol tariqasida AVTOMOBIL deb nomlangan obyektни ko'ramiz. Avtomobilga tegishli bo'lgan model, rang va narx uning xususiyatlari. Avtomobilga tegishli bo'lgan start(), stop() va tezlash() kabi amallar esan uning metodlari xisoblanadi.

## MUHOKAMA

Agar real dasturdan misol keltiradigan bo'lsak istalgan dastur ichidagi tugma bu obyekt, uning shakli, rangi va matni esa uning xususiyatlari bo'ladi. Unda bajariladigan amallar tugmaning metodlari deyiladi. Misol uchun: tugmani bosish, uzoq bosish yoki uning ustiga sichqonchani olib borish va boshqalar.

Object Oriented Programming(OOP) dasturlar o'nlab, balki yuzlab obyektlardan iborat bo'ladi. Bunday dasturlar uchun, dastur boshi yoki oxiri degan tushunchalar nisbiy bo'ladi. Object Oriented Programming(OOP) dasturlar bajarishda ketma-ketlikka amal qilinmaydi. Foydalanuvchi istalgan obyektga murojat qilishi, uning ustida turli amallar bajarishi mumkin. O'z navbatida 1ta obyektga murojat ortidan boshqa obyektlar ham faollashishi mumkin. Misol

uchun mobil ilovalarda obyektlar bu dastur ichdagi tugmalar , matnlar , rasmlar va boshqa elementlardir. Foydalanuchi istalgan tugmani bosishi , istalgan matnni ajratib olishi va boshqa amallarni istalgan tartibda bajarishi mumkin. Bunda 1ta tugma yani obyektни bosish bilan boshqa obyekt misol uchun rasm o'zgarishi mumkin. Zamonaviy kompyuter o'yinlari ham minglab obyektlardan iborat , foydalanuchi esa virtual o'yin olamida erkin harakat qilishi , istalgan tarafga yurishi , istalgan vaqtda turli obyektlar ustida turli amallar bajarishi mumkin.

Object Oriented Programming(OOP) xaqida gaplashar ekanmiz uning fundamental tushunchalari CLASS va OBJECT xaqida gapirmaslikni iloji yo'q. CLASS – bu obyekt yaratish uchun shablon(qolip)dir. Bitta class dan biz istalgancha nusxa olishimiz va yangi obyektlar yaratishimiz mumkin. Demak obyekt bu biror CLASS ning xususiy ko'rinishi , odatda CLASSlarning nomi o'zgarmas undan yaratilgan obyektlar esa istalgancha nomlanishi mumkin. Avval aytkanimizdek dasturimiz yuzlab obyektlardan iborat bo'lishi mumkin , Classlar esa bizga obyektlarni yaratishni yengillashtirib beradi. Bu hoh dastur interfeysidagi o'nglab turli xil tugmalar bo'lsin yoki kompyuter o'yinidagi qahramonlar bo'lsin , har bir tugma yoki o'yin qahramonlar bo'lsin , har bir tugma yoki o'yin qahramoni va uning xarakterlarini qayta qayta yozmasdan , bir marta yaratilgan Classdan nusxa olib o'nlab balki yuzlab obyektlarni yaratishimiz mumkin.

OOP ni tamolillari. Biz OOP xaqida gapira turib malum bir obyektga tegishli bo'lgan xususiyatlar va metodlarni bitta konteynerga joylaymiz dedik bu jarayon inkapsulatsiya yani kapsulaga solish deb ataladi. Inkapsulatsiya bizga classlar yaratishga va keyinchalik bu classlardan boshqa obyektlarni yaratishga yordam beradi. Abstraktsiya yordamida biz kodimizning ichki tuzulishini yashiramiz , yani tashqaridan qaraganda obyektimiz ikkita parametr va ikkita metoddan iborat bo'lishi mumkin lekin obyekt to'g'ri ishlashi uchun uning ichida o'nlab boshqa o'zgaruvchilar va funksiyalar yashirin bo'ladi. Classdan foydalanishda esa uning ichki tuzulishini va qanday ishlashini talab qilinmaydi. Bu o'z navbatida bizga ham, boshqa dasturchilarga ham bu Classdan foydalanishda qulaylik yaratadi.

Keyingi tamoyil Vorislik deb ataladi. Dasturlash davomida biz bir Classdan boshqa Classlar yaratishimiz mumkin. Misol uchun: Bizda Transport classi bor , biz bu classdan qo'shimcha Avtomobil , Avtobus , Kema va Poyezd kabi classlarni yaratishimiz mumkin , bunda bizning asl classimiz Ota yoki Supper Class deb ataladi , undan yaratilgan classlar esa voris classlar deyiladi , voris classlar ota classning bazi yoki barcha xususiyat va metodlariga ega bo'ladi.

Voris class supper classdan o'zlashtirgan nomini saqlagan xolda uning ishlashini o'zgartirishiga esa Polimorfizm deyiladi. Misol uchun: Biz kompyuter o'yini uchun qahramon yarataypmiz bir misol ko'ramiz , Qahramon degan class yaratdik , Qahramon bir nechta xususiyatlarga va metodlarga ega. Jumladan attack yani hujum qilish metodi bor. Endi biz bu super classdan boshqa voris classlarni yaratamiz , birinchi classimiz qilichboz va bu qahramon hujum qilganda qilichi bilan hujum qiladi , ikkinchi qahramonimiz esa jangchi va u quolsiz bo'lgani sababli qo'l va oyoqlari bilan hujum qiladi , uchunchi qahramonimiz pistolet bilan va oxirgisi esa kamon va yoqlar bilan qurollangan. To'rtala qahramonimiz ham supper classdan attack yani hujum metodini meros oldi , lekin biz bu metodni har bir qahramon uchun turli xil ko'rinishda yozishimiz va talqin qilishimiz mumkin. Bu esa o'z navbatida bizni turli qahramonlar va turli hujum turlari uchun alohida alohida metodlar yozishdan qutqaradi mana shular Object Oriented Programming(OOP) ning asosiy tamoyillari ekan.

## XULOSA

Albatta bizga OOP tushunchasi xaqidagi bilim va tajribalarimiz qolgan dasturlar va dasturiy muhitlarni o'rganishda , hamda tur xil loyihalar ustida ishlaganimizda as qotadi. OOP ning afzalliklar shundaki – Parallel dasturlar – bir loyihaning turli qislari bir vaqtda turli dasturchilar tomonidan yaratilishi mumkin , Vorislik tamoyili classlardan qayta foydalanish imkonini beradi , Polimorfizm tamoyili classlarni moslashuvchan qiladi va Classlardan boshqa dastur va loyihalarda qayta-qayta foydalanish mumkin.

OOP ning kamchiliklari ham qisman mavjud – dasturlashga yangi qadam qo'yganlar uchun biroz tushunarsiz , Har doim ham samarali emas , misol uchun: chiziqli va kichik dasturlarda va bazida dasturimiz haddan tashqari.

## REFERENCES

1. G'aniyev\_S\_K,\_Karimov\_M\_M,\_Tashiyev\_K\_V\_Axborot\_xavfsizligi\_2017 Muharrir: Tex. muharrir: Musawir: Musahhih: Kompyuterda sahifalovchi: Sh.Aliyeva M.Holmuhamedov D.Azizov N.Hasanova N.Raxmatullayeva.
2. Igam berdiyev X.Z. - Toshkent davlat texnika universiteti “Boshqarishda axborot texnologiyalari” kafedrası professorı, texnika fanlari doktori; Ahmedova O.P. - “Unicon.UZ” DUK, Kriptografiya ilmiy-tadqiqot bolim i boshlig'i, t.f.n.