**Ob’ektga yo'naltirilgan dasturlash tillari.**

Ob’ektga yo‘naltirilgan dasturlash atamasi  dastlab 60-yillarning  o‘rtalarida,  Simula-67 dasturlash tilida paydo bo‘lgan. Ammo, bu til FORTRAN, ALGOL, PL/1 kabi  tillarga  nisbatan o‘zining imkoniyatlari  past bo‘lganligi hamda shu davrda yechish talab qilingan masalalar uchun yetarli vositalarni taklif qila olmagani uchun  dasturchilar orasida keng ommalasha olmadi. Keyinchalik ob’ektga yo‘naltirilgan dasturlash  Smalltalk, Object Pascal, C++, Ada, hozirda esa C# kabi yuqori darajali dasturlash tillarining topologiyasi asosida rivojlandi va ommalashdi.

Ob’ektga yo‘naltirilgan dasturlash – bu dasturlashning shunday yangi yo‘nalishiki, dasturiy tizimda o‘zaro aloqada bo‘lgan ob’ektlar majmuasi sifatida  qaraladi va xar bir ob’ektni ma’lum bir klassga mansub hamda xar bir klass qandaydir shajarani  hosil qiladi deb hisoblanadi. Alohida olingan klass ma’lumotlar to‘plami va ular ustida bajariladigan amallarning to‘plami sifatida qaraladi. Bu klassning elementlariga faqat shu klassda aniqlangan amallar orqali murojaat qilish mumkin. Dasturdagi ma’lumotlar va ular ustida bajariladigan amallar o‘rtasidagi o‘zaro bog‘liqlik an’anaviy dasturlash tillariga nisbatan dasturiy tizimlarning ishonchliligini ta’minlaydi. Ob’ektga yo‘naltirilgan dasturlashning eng asosiy tushunchasi ob’ekt va klass hisoblanadi.

Ob’ekt. Oldimizda turgan olma mevasini dasturlash terminologiyasi orqali ko‘raylik. An’anaviy dasturlashda biz uni qismlarga bo‘lib o‘rganganmiz: S-olma po‘stining yuzi, J – olma mevasidagi sharbat hajmi, F – po‘st orasidagi  meva og‘irligi, D-urug‘lar soni va x.k. Endi ana shu olmaga rassom ko‘zi bilan qaraylik. Olmaning rasmi bu meva emas, balki olma mevasining yassi tekisliklikdagi tasviridir. Uni har biri alohida va bir-biriga bog‘liq bo‘lmagan biror ma’lumotlar segmentida turgan bir nechta ma’lumotlardan iborat bo‘lgan ko‘rinishda abstraksiyalash mumkin emas. Olmaning komponentalari doimo birgalikda va bu komponentalar orasidagi o‘zaro aloqalarni saqlagan holda qaraladi.

Ob’ekt**-**biz yashayotgan olamdagi biror elementga hos bo‘lgan barcha ma’lumot va hulqlarni, ya’ni  shu element ustida bajarish mumkin bo‘lgan xarakatlarni ifodalaydi hamda  ma’lumotlarning  tugal abstraksiyasidan iborat bo‘ladi. Bu ma’lumot va hulqlar ob’ektga yo‘naltirilgan dasturlash atamasida mos ravishda hususiyat va metod deb ataladi. Hususiyatni ob’ektning maydoni deb ham yuritiladi. Masalan, shashka ob’ekti rang, vertikal maydondagi o‘rni, gorizontal maydondagi o‘rni kabi maydonlarga, surish, urish, «damka» ga chiqish, shashka taxtasidan chetga olish kabi metodlarga ega bo‘ladi. Maydon va metodlar birgalikda ob’ektning a’zolari deyiladi. Ob’ektlarning strukturasi ularning  o‘zaro aloqasini ifodalaydi. Ob’ektlar o‘zlarining barcha xarakteristika va hulqlarining o‘ziga hos tomonlarini birgalikda saqlaydi.

Klass. Har bir ob’ekt qandaydir bitta klassga taaluqli bo‘ladi. Klass – bu murakkab struktura bo‘lib, o‘z ichiga ma’lumotlarni, protsedura va funksiyalarni ifodalashdan tashqari, klasslarning vakili bo‘lmish ob’ektlar ustida bajarilishi mumkin bo‘lgan amallarni ham oladi. Klassdagi ma’lumotlar maydonlar, protsedura va funksiyalar esa metodlar deb ataladi. Klassning strukturasi tushunchasi ham muhim hisoblanadi. U tizim ichidagi vositalar boyligini namoyon qiladi. Ma’lumki, biror yaproqdagi fotosintez jarayonini o‘rganish uchun shu yaproqdagi bitta xujayrani ko‘rish yetarli, chunki, qolgan xujayralar ham o‘zini ana shu o‘rganilgan xujayra kabi tutadi. Biz xam biror klass, tip yoki bo‘limga mansub bo‘lgan ob’ektni ko‘rar ekanmiz, faraz qilishimiz mumkinki, uning hulqi ham shu tipdagi boshqa ob’ektlarniki kabi bo‘ladi.
Ob’ektga yo‘naltirilgan dasturlash bizning hulqimizga o‘xshab ketadigan dasturlash usulini ifodalaydi. U dasturlash tillarini ishlab chiqishdagi yangiliklarning tabiiy evolyusiyasi hisoblanadi. U barcha avvalgi dasturlash tillariga qaraganda strukturaliroq, ma’lumotlarni abstraktlashga urinishda abstraktliroq va modulliroqdir.
Ob’ektga yo‘naltirilgan dasturlash o‘zining uchta asosiy hususiyatlari  bilan xarakterlanadi:

1. Inkapsulyasiya – yozuvlarni shu yozuvlarning maydonlari ustida amallar bajarish uchun mo‘ljallangan funksiyalar va protseduralar bilan birgalikda olib boradi. Inkapsulyasiya yashirish prinsipiga ham ega bo‘ladi. Buning ma’nosi shuki, ob’ekt ustida bajarish mumkin bo‘lgan xarakatlarni amalga oshirish vositalari shu ob’ektdan foydalanayotgan dasturchilar ko‘zidan yashirib qo‘yiladi. Dasturchi faqat shu ob’ektning ayrim metod va maydonlari bilangina ishlay oladi. Boshqacha aytganda, barcha maydon va metodlar ichki va tashqi guruhlarga bo‘linadi. Ob’ektning ichki a’zolari dasturchiga «ko‘rinmaydi» va ob’ektning hulqi va imkoniyatlarini aniqlaydi, tashqi a’zolari esa dasturchiga «ko‘rinib turadi» hamda ob’ektni boshqarish imkonini beradi. Ob’ektning dasturchiga ko‘rinib, ob’ektni boshqarishga yordam beradigan metod va maydonlari  (hususiyatlari) ob’ektning interfeysi deb ataladi. Dasturchi ob’ekt bilan ishlash uchun uning interfeysini bilishi kifoya.

Masalan, yengil avtomobilni boshqarishni o‘rganish uchun uning matorining ishlash prinsipi, g‘ildiraklarning burilishini, tormoz mexanizmini o‘rganishning hojati yo‘q, rulni burash, pedal yoki uzatmalar qutisi richagini bosishni bilish yetarli.

2. Vorislik – yangi ob’ektni aniqlashga ehtiyoj paydo bo‘lganda  oldindan ma’lum bo‘lgan ob’ektlardan foydalanish.  Ob’ektlar o‘zlarini yaratishda ishtirok etgan ota ob’ektlarning xarakteristika va hulqlarini meros qilib olishlari mumkin. OYD konsepsiyasi yangi klasslarni mavjud klasslarga yangi maydonlar, hususiyatlar va metodlarni qo‘shish orqali yaratish imkonini ham beradi. Yangi klasslarni tashkil qilishning bunday usuli yuzaga keltirish deb ataladi.  Bu holda yuzaga kelgan yangi klass o‘zining bazaviy ota klassiga hos bo‘lgan hususiyat va metodlarini meros oladi.

Masalan hayotdan misol qilib, hasharotlar klassini olish mumkin. U ikki guruhga bo‘linadi: qanotlilar va qanotsizlar. Qanotli hasharotlarga kapalaklar, pashshalari, parvonalar va x.k. lar kiradi. Shuning uchun, pashshalarni qaytadan qanotli deb ta’riflashning hojati yo‘q, u bu hususiyatni ota klassi bo‘lgan qanotli hasharotlar  klassidan meros qilib oladi.

3.  Polimorfizm– xarakatga biror nom berish hamda undan birgalikda ob’ektlar shajarasining quyi va yuqori qismlarida foydalanish. Bunda shajaraning har bir ob’ekti bu xarakatni o‘zi uchun hos bo‘lgan usul bilan bajaradi. Polimorfizm — bu turli klasslarga kirgan metodlar uchun bir hil nomlardan foydalanish imkoniyatidir. Polimorfizm konsepsiyasida ob’ektga nisbatan metod qo‘llanganida aynan ob’ektning klassiga mos keluvchi metoddan foydalanishni ta’minlaydi.
Object Pascal  dasturlash tili  ob’ektga yo‘naltirilgan dasturlashga hos bo‘lgan barcha vositalarni bera oladi: strukturalilik, modullik, katta abstraktlilik. Bu xarakteristikalarning hammasi xizmat ko‘rsatish uchun juda ham sodda, osongina boshqa vaziyatlarga moslashtirila oladigan, kuchliroq strukturaga ega bo‘lgan  kodlarda o‘z aksini topadi.
Ob’ektga yo‘naltirilgan dasturlash uzoq yillar davomida an’anaviy, ya’ni standart hisoblangan dasturlashga nisbatan hos bo‘lgan tasavvurlarni bir chetga qo‘yishni talab qiladi. Natijada ob’ektga yo‘naltirilgan dasturlash juda ham sodda, ko‘rgazmaliligi yuqori bo‘lib, dasturiy ta’minot yaratishdagi ko‘plab muammolarni hal qilishning juda ajoyib vositasiga aylanadi.

Yuqorida aytilgan fikrlarni hisobga olsak, ob’ektga yo‘naltirilgan dasturlash texnologiyalari quyidagicha masalalarni hal qilishi mumkin:

1. An’anaviy dasturlash tillarida mavjud bo‘lgan kamchiliklarni bartaraf qilish;

2. An’anaviy dasturlash tillari yordamida yechib bo‘lmaydigan yoki juda katta qiyinchiliklar bilan yechilishi mumkin bo‘lgan masalalarni hal qilish;

3. Qayta ishlash mumkin bo‘lgan ma’lumotlar va ularning tiplari doirasi an’anaviy dasturlash tillariga nisbatan ancha keng;

4. Foydalanuvchilar uchun qulay bo‘lgan muloqot interfeysini yaratish;

5. Kiritilayotgan va chiqarilayotgan turli tipdagi ma’lumotlarni nazorat qilish;

6. Yangi tipdagi ma’lumotlar, klasslar va modullarni osongina tashkil etish va ma’lumotlarni  nazorat qilish;

7. Multimedia va animatsion vositalaridan foydalanib, turli darajadagi tovushli va  xarakatli effektlarni hosil qilish va qayta ishlash;

8. Ma’lumotlar bazasi va undagi ma’lumotlar ustida amallarni bajarish, SQL so‘rovnomalari yordamida ma’lumotlarni qidirib topish kabi masalalar juda osonlik bilan hal qilish;

9. OLE konteyneri yordamida WINDOWS muhiti uchun mo‘ljallangan ilovalardagi ob’eklar bilan ishlash;

10. Foydalanuvchilar uchun yaratilgan dasturiy ta’minotdan foydalanish uchun yordamchi ma’lumotnomalar tizimini yaratish;

11. Dasturiy ta’minotni boshqa kompьyuterlarga ko‘chirish uchun o‘rnatuvchi disklarni yaratish;

12. Dastur matnini tashkil qilishda yuzaga kelishi mumkin bo‘lgan xatoliklar bilan ishlash masalasini hal qilish va x.k.

Ko‘rinib turibdiki, echilayotgan masalalarni an’anaviy dasturlash tillari yordamida echishda yuzaga kelishi mumkin bo‘lgan kattakon bo‘shliqni ob’ektga yo‘naltirilgan dasturlash texnologiyalari to‘ldiradi hamda zamonaviy dasturlashning ko‘plab talablariga javob beradi.