1-ma’ruza

OBYEKTGA MO’LJALLANGAN DASTURLASH ASOSLARI

REJA

- Rivojlanish tarixi
- Obyektga mo’ljallangan yondashuv
- Protseduraviy va obyektga mo’ljallangan dasturlash
- Obyektga mo'ljallangan yondashuvning afzalliklari va maqsadlari

RIVOJLANISH TARIXI

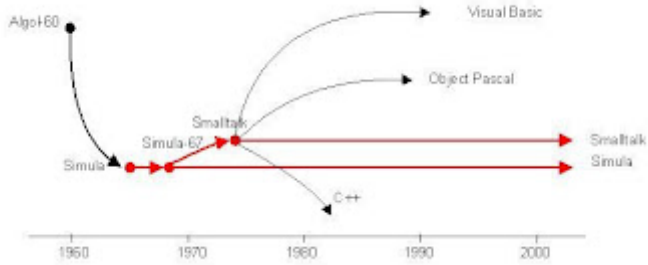
Obyektga mo’ljallangan dasturlash (OMY, объектное ориентирование программирование, object oriented programming – OOP) protsedurali dasturlash paradigmasining rivojlanishi natijasida vujudga keldi, bu yerda ma'lumotlar va ularni qayta ishlashning kichik dasturlari (protseduralar, funktsiyalar) rasmiy ravishda bog'liq emas. Obyektga yo'naltirilgan dasturlashni yanada rivojlantirish uchun ko'pincha voqea (hodisaga yo'naltirilgan dasturlash) va komponent (komponentga yo’naltirilgan dasturlash, KYD) tushunchalari katta ahamiyatga ega.

RIVOJLANISH TARIXI

Dastlabki tushunchalar taklif qilingan, keyinchalik paradigmaga aylangan birinchi dasturlash tili **Simula** edi, lekin "obyekt yo'nalishi" atamasi bu tilni ishlatish kontekstida ishlatilmadi. 1967-yilda paydo bo'lganida, unda inqilobiy g'oyalar taklif qilingan: obyektlar, sinflar, virtual usullar va boshqalar, lekin bularning hammasini zamondoshlari muhim tushunchalar deb bilishmagan.

RIVOJLANISH TARIXI

Aslida, Simula sinflar bilan Algol bo'lgan, bu protsedurali dasturlashda ko'plab murakkab tushunchalarni ifodalashni osonlashtirgan. Simuladagi sinf tushunchasini Algol konstruksiyalari yordamida to'liq aniqlash mumkin (ya'ni Simuladagi sinf - bu primitivlar yordamida tasvirlangan murakkab tushuncha)



Kristen Nigord(1926-2002)

Ole-Yoxan Dal (1931-2002)

RIVOJLANISH TARIXI

Smalltalkda Alan Kaye va Den Ingalls dasturlash (protseduradan tashqari) ga yangi nuqtai nazarni kiritdilar. Bu yerda sinf konsepsiyasi tilning boshqa barcha konstruktsiyalari uchun asosiy g'oyaga aylandi (ya'ni Smalltalkdagi sinf - bu primitiv, u orqali murakkab tuzilmalar tasvirlangan). Aynan u birinchi bo'lib obyektga yo'naltirilgan dasturlash tiliga aylandi.

Hozirgi vaqtda ob'ektga yo'naltirilgan paradigmani amalga oshiradigan amaliy dasturlash tillari soni (tillar ro'yxati) boshqa paradigmalarga nisbatan eng katta hisoblanadi. Sanoatdagi eng keng tarqalgan tillar (C++, Delphi, C#, Java va boshqalar) Simula obyekt modelini o'zida mujassam etgan. Smalltak modeliga asoslangan tillarga misollar: Objective-C, Python, Ruby.

OBYEKTGA MO’LJALLANGAN YONDASHUV

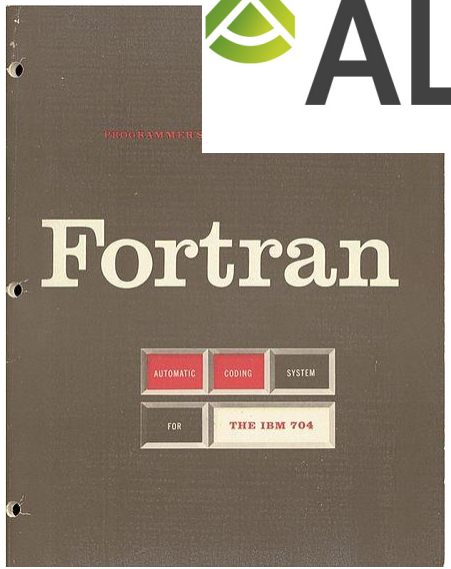
Obyektga mo'ljallangan yondashuv (OMY) dasturiy ta’minotning tabiiy rivojidagi navbatdagi pog‘onadir. Vaqt o'tishi bilan qaysi uslublar ishlash uchun qulay-u, qaysinisi noqulay ekanini aniqlash oson bo'lib bordi. OMY eng muvaffaqiyatli, vaqt sinovidan o‘tgan uslublarni o'zida samarali mujassam etadi.

OBYEKTGA MO’LJALLANGAN YONDASHUV

Dastlab dasturlar kommutatsiya bloki orqali kompyuterning asosiy xotirasiga to‘g‘ridan to‘g‘ri kiritilar edi. Dasturlar mashina tillarida ikkilik tasavvurda yozilar edi. Dasturlarni mashina tilida yozishda tez-tez xatolarga yo'l qo'yilar edi, buning ustiga ularni tizimalashtirishning imkoni bo'lmagani tufayli kodni kuzatib borish amalda deyarli mumkin bo’lmagan hol edi. Bundan tashqari, mashina kodlaridagi dasturni tushunish g’oyat murakkab edi.

OBYEKTGA MO’LJALLANGAN YONDASHUV

Vaqt o‘tishi bilan kompyuterlar tobora kengroq qo'llana boshlandi hamda yuqoriroq darajadagi protsedura tillari paydo bo’ldi. Bularning dastlabkisi FORTRAN tili edi. Biroq OMYning rivojiga asosiy ta’sirni keyinroq paydo bo‘lgan. Masalan, ALGOL kabi protsedura tillari ko‘rsatdi.



ALGOL

PROTSEDURAVIY VA OBYEKTGA MO’LJALLANGAN DASTURLASH

Shu vaqtgacha dasturlar berilgan ma'lumotlar ustida biror-bir amal bajaruvchi protseduralar ketma-ketligidan iborat edi. Protsedura yoki funksiya ham o'zida aniqlangan ketma-ket bajariluvchi komandalar to'plamidan iborat. Bunda berilgan ma’lumotlarga murojaatlar protseduralarga ajratilgan holda amalga oshiriladi.

Protsedura tillari dasturchiga axborotga ishlov berish dasturini pastroq darajadagi bir nechta protseduraga bo’lib tashlash imkonini beradi. Pastroq darajadagi bunday protseduralar dasturning umumiy tuzilmasini belgilab beradi. Ushbu protseduralarga izchil murojaatlar protseduralardan tashkil topgan dasturlarning bajarilishini boshqaradi.

PROTSEDURAVIY VA OBYEKTGA MO’LJALLANGAN DASTURLASH

Dasturlashning bu yangi paradigmasi mashina tilida dasturlash paradigmasiga nisbatan ancha ilg‘or bo'lib, unga tuzilmalashtirishning asosiy vositasi bo’lgan protseduralar qo'shilgan edi. Kichik funksiyalarni nafaqat tushunish, balki sozlash ham osonroq kechadi.

Strukturaviy dasturlashning asosiy g'oyasi «bo‘lakla va hukmronlik qil» prinsipiga butunlay mos keladi. Kompyuter dasturni masalalar to'plamidan iborat deb qaraymiz. Oddiy tavsiflash uchun murakkab bo'lgan ixtiyoriy masalani bir nechta, msbatan kichikroq bo‘lgan, tarkibiy masalalarga ajratamiz va bo’linishni toki masalalar tushunishi uchun yetarli darajada oddiy bo'lguncha davom ettiramiz.

PROTSEDURAVIY VA OBYEKTGA MO’LJALLANGAN DASTURLASH

Misol sifatida kompaniya xizmatchilarining o'rtacha ish haqini hisoblashni olamiz. Bu masala sodda emas. Uni qator qism masalalarga bo'lamiz:

1. Har bir xizmatchining oylik maoshi qanchaligini aniqlaymiz.
2. Kompaniya xodimlari sonini aniqlaymiz.
3. Barcha ish haqlarini yig’amiz.
4. Hosil bo'lgan yig'indini kompaniya xodimlari soniga bo'lamiz.

PROTSEDURAVIY VA OBYEKTGA MO’LJALLANGAN DASTURLASH

Xodimlarning oylik maoshlari yig‘indisim hisoblash jarayonini ham bir necha bosqichlarga ajratish mumkin.

1. Har bir xodim haqidagi yozuvni o‘qiymiz.
2. Ish haqi to'g'risidagi ma’lumotni olamiz.
3. Ish haqi qiymatini yig'indiga qo'shamiz.
4. Keyingi xodim haqidagi yozuvni o'qiymiz.

PROTSEDURAVIY VA OBYEKTGA MO’LJALLANGAN DASTURLASH

O'z navbatida, har bir xodim haqidagi yozuvni o‘qish jarayonini ham nisbatan kichikroq qism operatsiyalarga ajratish mumkin:

1. Xizmatchi faylini ochamiz.
2. Kerakli yozuvga o'tamiz.
3. Ma'lumotlarni diskdan o'qiymiz.

Strukturaviy dasturlash murakkab masalalarni yechishda yetarlicha muvafaqqiyatli uslub bo'lib qoldi. Lekin 1980-yillar oxirlarida strukturaviy dasturlashning ham ayrim kamchiliklari ko'zga tashlandi.

PROTSEDURAVIY VA OBYEKTGA MO’LJALLANGAN DASTURLASH

Birinchidan, berilgan ma'lumotlar (masalan, xodimlar haqidagi yozuv) va ular ustidagi amallar (izlash, tahrirlash) bajarilishining bir butun tarzda tashkil etilishidek tabiiy jarayon realizatsiva qilinmagan edi. Aksincha, protseduraviy dasturlash berilganlar strukturasini bu ma’lumotlar ustida amallar bajaradigan funksiyalarga ajratgan edi.

Ikkinchidan, dasturchilar doimiy tarzda eski muammolarning yangi yechimlarini ixtiro qilar edilar. Bu vaziyat ko'pincha velosipedni qayta ixtiro qilish ham deb aytiladi. Ko‘plab dasturlarda takrorlanuvchi bloklarni ko‘p martalab qo'llash imkoniyatiga bo'lgan xohish tabiiydir. Buni radio ishlab chiqaruvchi tomonidan priyomnikni yig'ishga o'xshatish mumkin. Konstruktor har safar diod va tranzistorni ixtiro qilmaydi. U oddiygina – oldin tayyorlangan radio detallaridan foydalanadi xolos. Dasturiy ta’minotni ishlab chiquvchilar uchun esa bunday imkoniyat ko'p yillar mobaynida yo‘q edi.

PROTSEDURAVIY VA OBYEKTGA MO’LJALLANGAN DASTURLASH

Boshqa tomondan, protsedurali dasturlash koddan takroran foydalanish imkonini cheklab qo'yadi. Va, nihoyat, shu narsaaniq bo‘ldiki, protsedurali dasturlash usullari bilan dasturlarni ishlab chiqishda diqqatni ma’lumotlarga qaratishning o'zi muammolarni keltirib chiqarar ekan. Chunki ma'lumotlar va protsedura ajralgan, ma'lumotlar inkapsulatsiyalanmagan. Bu nimaga olib keladi? Shunga olib keladiki, har bir protsedura ma’lumotlarni nima qilish kerakligini va ular qayerda joylashganini bilmog'i lozim bo'ladi. Agar protsedura ma'lumotlar ustidan noto'g'ri amallarni bajarsa, u ma'lumotlarni buzib qo'yishi mumkin. Har bir protsedura ma’lumotlarga kirish usullarini dasturlashi lozim bo'lganligi tufayli, ma’lumotlar taqdimotning o'zgarishi dasturnnig ushbu kirish amalga oshirilayotgan barcha o'rinlarining o'zgarishiga olib kelar edi. Shunday qilib, hatto eng kichik to'g'rilash ham butun dasturda qator o'zgarishlarni sodir bo'lishiga olib kelar edi.

PROTSEDURAVIY VA OBYEKTGA MO’LJALLANGAN DASTURLASH

Modulli dasturlashda, masalan, Modula2 kabi tilda protsedurali dasturlashda topilgan ayrim kamchiliklarni bartaraf etishga urinib ko'rildi. Modulli dasturlash dasturni bir necha tarkibiy bo’laklarga, yoki, boshqacha qilib aytganda, modullarga bo'lib tashlaydi. Agar protsedurali dasturlash ma'lumotlar va jarayonlarm bo'lib tashlasa, modulli dasturlash, undan farqli o‘laroq, ularni birlashtiradi. Modul ma’lumotlarning o'zidan hamda ma’lumotlarga ishlov beradigan protseduralardan iborat. Dasturning boshqa qismlariga moduldan foydalanish kerak bo’lib qolsa, ular modul interfeysiga murojaat etadi. Modullar barcha ichki axborotni dasturning boshqa qismlarida yashiradi.

PROTSEDURAVIY VA OBYEKTGA MO’LJALLANGAN DASTURLASH

Biroq modulli dasturlash ham kamchiliklardan holi emas. Modullar kengaymas bo'ladi, bu degani kodga bevosita kirishsiz hamda uni to’g’ridan to’g'ri o’zgartirmay turib modulni qadam-baqadam o'zgartirish mumkin emas. Bundan tashqari, bitta modulni ishlab chiqishda, uning funksiyalarini boshqasiga o'tkazmay (delegat qilmay) turib boshqasidan foydalanib bo’lmaydi. Yana garchi modulda turni belgilab bo’lsa-da, bir modul boshqasida belgilangan turdan foydalana olmaydi.

Modulli va prosedurali dasturlash tillarida turni kengaytirish usuli, agar «agregatlash» deb ataluvchi usul yordamida boshqa turlarni yaratishni hisobga olmaganda, mavjud emas.

PROTSEDURAVIY VA OBYEKTGA MO’LJALLANGAN DASTURLASH

Va nihoyat, modulli dasturlash - bu yana protseduraga mo'ljallangan gibridli sxema bo'lib, unga amal qilishda dastur bir necha protseduralarga bo'linadi. Biroq endilikda protseduralar ishlov berilmagan ma'lumotlar ustida amallarni bajarmaydi, balki modullarni boshqaradi.

Amaliyotga do’stona foydalanuvchi interfeyslari, ramkali oyna, menu va ekranlarning tadbiq etilishi dasturlashda yangi uslubni keltirib chiqardi. Dasturlarni ketma-ket boshidan oxirigacha emas, balki uning alohida bloklari baiarilishi talab qilinadigan bo'ldi. Biror-bira niqlangan hodisa yuz berganda dastur unga mos shaklda ta'sir ko'rsatishi lozim. Masalan, bir tugma bosilganda faqatgina unga biriktirilgan amallar bajariladi. Bunday uslubda dasturlar ancha interaktiv bo'lishi lozim. Buni ularni ishlab chiqishda hisobga olish lozim.

PROTSEDURAVIY VA OBYEKTGA MO’LJALLANGAN DASTURLASH

Obyektga mo’ljallangan dasturlash (OMD) bu talablarga to’la javob beradi. Bunda dasturiy komponentlarni ko‘p martalab qo’llash va berilganlarni manipulatsiya qiluvchi usullar bilan birlashtirish imkoniyati mavjud.

Obyektga mo'ljallangan dasturlashning asosiy maqsadi berilganlar va ular ustida amal bajaruvchi protseduralarni yagona obyekt deb qarashdan iboratdir.

OBYEKTGA MO'LJALLANGAN YONDASHUVNING AFZALLIKLARI VA MAQSADLARI

OMY dasturiy ta’minotni ishlab chiqishda oltita asosiy maqsadni ko'zlaydi. OMY paradigmasiga muvofiq ishlab chiqilgan dasturiy ta’minot quyidagi xususiyatlarga ega bo‘lmog’i lozim:

1) tabiiylik;

2) ishonchlilik;

3) qayta qo'llanish imkoniyati;

4) kuzatib borishda qulaylik:

5) takomillashishga qodirlik;

6) yangi versiyalarni davriy chiqarishning qulayligi

OBYEKTGA MO'LJALLANGAN YONDASHUVNING AFZALLIKLARI VA MAQSADLARI

Ishonchlilik. Yaxshi dasturiy ta'minot boshqa har qanday mahsulotlar, masalan, muzlatgich yoki televizorlar kabi ishonchli bo’lmog'i lozim.

Puxta ishlab chiqilgan va tartib bilan yozilgan obyektga mo'ljallangan dastur ishonchli bo'ladi. Obyektlarning modulli tabiati dastur qismlaridan birida, uning boshqa qismlariga tegmagan holda, o'zgartirishlarni amalga oshirish imkonini beradi. Obyekt tushunchasi tufayli, axborotga ushbu axborot kerak bo’lgan shaxslar egalik qiladi, ma’suliyat esa berilgan funksiyalarni bajaruvchilar zimmasiga yuklatiladi.

OBYEKTGA MO'LJALLANGAN YONDASHUVNING AFZALLIKLARI VA MAQSADLARI

Kuzatib borishda qulaylik. Dasturiy mahsulotning ish berish davri uning ishlab chiqilishi bilan tugamaydi. Dasturni ishlatish jarayonida *kuzatib borish* deb nomlanuvchi jarayon muhimdir. Dasturga sarflangan 60 foizdan 80 foizgacha vaqt kuzatib borishga ketadi. Ishlab chiqish esa ish berish sikllning 20 foizinigina tashkil etadi.

Puxta ishlangan obyektga mo'ljallangan dastur ishlatishda qulay bo'ladi. Xatoni bartaraf etish uchun faqat bitta o'ringa to'g'rilash kiritish kifoya qiladi. Chunki ishlatishdagi o'zgarishlar, boshqa barcha obyektlar takomillashtirish afzalliklaridan avtomatik ravishda foydalana boshlaydi. O‘zining tabliyligi tufayli dastur matni boshqa ishlab chiquvchilar uchun tushunarli bo’lmog'i lozim

OBYEKTGA MO'LJALLANGAN YONDASHUVNING AFZALLIKLARI VA MAQSADLARI

Kengayishga qodirlik. Foydalanuvchilar dasturni kuzatib borish paytida tez-tez tizimga yangi funksiyalarni qo'shishni iltimos qiladilar. Obyektlar kutubxonasini tuzishning o'zida ham ushbu obyektlarning funksiyalarini kengaytirishga to'g'ri keladi.

Dasturiy ta'minot statik (qotib qolgan) emas. Dasturiy ta’minot foydali bo'lib qolishi uchun, uning imkoniyatlarini muttasil kengaytirib boorish lozim. OMY da dasturni kengaytirish usullari ko'p. Vorislik, polimorfizm, qayta aniqlash, vakillik hamda ishlab chiqish jarayonida foydalanish mumkin bo'lgan ko'plab boshqa shablonlar shular jumlasidandir

OBYEKTGA MO'LJALLANGAN YONDASHUVNING AFZALLIKLARI VA MAQSADLARI

Yangi versiyalarning davriy chiqarilishi. Zamonaviy dasturiy mahsulotning ish berish davri ko'p hollarda haftalar bilan o'lchanadi. OMY tufayli dasturlarni ishlab chiqish davrini qisqartirishga erishildi, chunki dasturlar ancha ishonchli bo'lib bormoqda, kengayishi osonroq hamda takroran qo'llanishi mumkin.

Dasturiy ta’minotning tabiiyligi murakkab tizimlarning ishlab chiqilishini osonlashtiradi. Har qanday ishlanma hafsala bilan yondashuvni talab qiladi, shuning uchun tabiiylik dasturiy ta’minotning ishlab chiqish davrlarini qisqartirish imkonini beradi, chunki butun diqqat-e’tiborni yechilayotgan masalaga jalb qildiradi.

OBYEKTGA MO'LJALLANGAN YONDASHUVNING AFZALLIKLARI VA MAQSADLARI

Dastur qator obyektlarga bo'lingach, har bir alohida dastur qismini boshqalari bilan parallel ravishda ishlab chiqish mumkin bo'ladi. Bir nechta ishlab chiquvchi sinflarni bir-birlaridan mustaqil ravishda ishlab chiqishi mumkin bo'ladi. Ishlab chiqishdagi bunday parallellik ishlab chiqish vaqtini qisqartiradi.

E’TIBORINGIZ UCHUN RAHMAT