

OB'YEKTLI ALGORITM

N. A. Otaxanov NamDU

Ushbu maqolada algoritm tushunchasi ob'yektli dasturlash tillariga mos ko'rinishda talqin qilingan. Ob'yektli algoritmgga oid asosiy tushunchalar va ularning xossalari va ahamiyati namunalar yordamida ommabop ko'rinishda izohlab berilgan.

Biz yaxshi bilamizki, algoritm tushunchasi buyuk o'zbek matematigi Al-Xorazmiy nomidan kelib chiqib, uning mashhur "Al-jabr va al-Muqobala" asarining tarixiy ahamiyati bilan bo'liq.

XX asming oxiri va XXI asrning boshlarida dasturchilar ixtiyoriga o'z imkoniyatlariga ko'ra ana'naviy dasturlash tillariga qaraganda bir necha marta boy va qulay bo'lgan ob'yektga yo'naltirilgan dasturlash texnologiyalari taklif qilindi. Bu texnologiyalarning keyingi versiyalari avvalgilariga qaraganda mukammalroq hamda borgan sari o'z ichiga ko'plab INTERNET tizimi, tarmoqlar, animatsiyalar va ma'lumotlar bazalari bilan ishlash kabi yangi imkoniyatlar uchun zarur bo'lgan komponentlarni olish hisobiga kengayib bormoqda.

Bugungi kunga kelib, dasturlash tillarini o'rganuvchilar uchun bir qaraganda, algoritm tushunchasi kerak emas, uning dasturlash bilan aloqasi yo'q qabilidagi tasavvurlar paydo bo'lib qoldi. Ular masalalarni yechish bosqichlari, algoritm qurishning yo'l-yo'riqlari bilan tanishmasdan turib, dasturlash sohasiga kirishga harakat qilishadi. Na'ajada, ular dasturlash tillarida ko'zda tutilgan qonun-qoidalar, buyruqlar tizimini yetarici darajada o'rganishlari (shunchaki yodlashlari) mumkin, ammo aniq qo'yilgan masala uchun dastur yozishda juda katta qiyinchiliklarga duch kelishadi. Ana shunday vaziyatlarga tushib qolmaslik uchun biz bu dasturchilarga dasturlashni algoritm qurishdan boshlashni tavsiya qilamiz.

Bir qaraganda, ana'naviy dasturlash tillariga xos bo'lgan algoritm tushunchasi o'z ahamiyatini yo'qotgandek bo'lmoqda. Aslida ham shundaymi? Bizning nazarimizda bu savolga "yo'q" deb javob berish mumkin. Chunki, hozirgi kunda bu tushuncha boshqacha ko'rinishda, ya'ni zamonaviy dasturlash tillarining eng muhim tushunchasi bo'lgan ob'yeatlarga nisbatan qo'llanmoqda. Shuning uchun bu

tushunchani yangicha talqin qilishga to'g'ri keladi.

Ma'lumki, ixtiyoriy masalani hal qilish uchun maxsus algoritm ishlab chiqish talab qilinadi. Shunday algoritmning mayjud emasligi masala yechimining mayjud emasligini anglatadi.

Zamonaviy ob'yektga yo'naltirilgan dasturlash tillari uchun N. Vitrning "algoritm+berilganlar=dastur" ta'rifi boshqacharoq talqin qilinadi. Asosiy e'tiborni algoritm tuzishga emas, balki ma'lumotlarning berilganlar strukturasiga qaratiladi.

Algoritmni ishlab chiqishda ob'yekt, klass va metodlardan keng foydalaniladi. 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.

Ta'rif. Ob'yektli algoritm deb qo'yilgan masalani to'la hal qilish uchun zarur bo'lgan ob'ektlarning xususiyatlarini o'zgartirish uchun ijrochining bajarishi talab qilingan amallar va metodlar ketma-ketligining qat'iy tartibiga aytiladi.

Ob'yektli algoritm qurish uchun uning qanday xususiyatlarga ega ekanligi hamda ana shu xususiyatlar ustida bajarish mumkin bo'lgan amallarni bilish lozim. Xususiyatlar va ularning qiymatlari bir-biridan "." (nuqta) belgisi yordamida ajratib ko'rsatiladi

1-misol. Uyni isitgich yordamida isitish talab qilingan bo'lsin. Bu masalani yechishda ijrochi, ya'ni uy egasi isitgich ob'yektidan foydalanadi. Isitgich ob'yekti rozetka, o'chirgich hamda o'zining maxsus ish rejimlari kabi xususiyatlariga ega. Ijrochining tarmoqda elektr kuchlanishining mavjudligi qiziqitirmaydi. Chunki kuchlanishni muhit ta'minlab beradi. Ma'lumki, rozetka xususiyati umumiy elektr tarmog'iga "ulangan" yoki "ulanmagan", o'chirgich xususiyati "yoqilgan" yoki "o'chirilgan", ish rejimi xususiyati esa "kuchli", "o'rta" va "kuchsiz" qiymatlarni olishi mumkin. Demak, ijrochi bu xususiyatlarni o'zgartirib, oldiga qo'yilgan masalani hal qilishi mumkin.

Bu misoldagi xususiyatlarni quyidagi tartibda belgilab qo'yish mumkin:

- 1) o'chirgich.ulangan (yoki o'chirgich.rost);
- 2) Ish_rejimi.kuchli;
- 3) Rozetka. O'chirilgan (yoki Rozetka.yoqilgan).

2-misol. Shashka donasi rang, gorizontaal va vertikalidagi o'rni kabi xususiyatlarga hamda yurish, urish va taxtadan chiqish olish kabi amallarga (metodlarga) ega. O'yinchi berilgan xususiyatlarni o'zgartirib, mumkin bo'lgan amallarni bajarib, o'yin taqdirini hal qilishi mumkin.

Klassning strukturasi tushunchasi ham muhim hisoblanadi. U sistema ichidagi vositalar boyligini namoyon qiladi. Ma'lumki, biror yaroqdagi fotosintez jarayonini o'rganish uchun shu yaroqdagi bitta hujayrani ko'rish yetarli, chunki qolgan hujayralar ham o'zini ana shu o'rganilgan hujayra kabi tutadi. Biz ham algoritmlarni qurish jarayonida biror klass, tip yoki bo'limga mansub bo'lgan ob'yektni ko'rar ekanmiz, faraz qilishimiz mumkinki, uning xulqi ham shu tipdagi boshqa obyektlarniki kabi bo'ladi.

Ob'yektlilik algoritmlar quyidagi talablarga javob berishi lozim, aks holda u ob'yektlilik algoritmi bo'la olmaydi.

1. Algoritmida foydalanilayotgan har bir ob'yekt qandaydir bitta klassga taalluqli bo'lishi lozim.

2. Algoritm foydalanuvchilar uchun mo'ljallangan, ya'ni masala shartida berilgan ma'lumotlarni kiritish hamda talab qilingan ma'lumotlarni chiqarish uchun interfeys, ya'ni muloqot oynasiga ega bo'lishi shart;

3. Algoritmning har bir qadami buyruq tarzida rasmiylashtiriladi;

4. Algoritm chekli xondagi qadam va buyruqlardan iborat bo'lishi kerak;

5. Amal yoki metodlarda foydalanilayotgan o'zgaruvchilarning qiymatlari oldindan aniqlanishi lozim;

6. Har bir amal yoki metodning natijasi bir qiymatli bo'ladi;

7. Interfeysga ishni boshlash uchun bir yoki bir nechta tugmali oynalar kiritilishi hamda bu tugmalarning birini tanlanishiga qarab algoritmlar reaksiyasini aniqlash kerak.

8. Algoritm ommaviy bo'lishi lozim.

Bu talablarning ayrimlarini izohlaymiz.

1-talabga ko'ra, har bir ob'yekt biror klassga taalluqli boladi, ya'ni shu klassga xos bo'lgan barcha xususiyatlarni olgan bo'ladi. Ehtiyoj paydo bo'lganda bu xususiyatlarni algoritm qadamlari yordamida o'zgartirilishi mumkin. Masalan, 1-misolidagi shashka donasi rangi, taxtaning gorizontal va vertikal qatoridagi turgan o'rni kabi xususiyatlarga ega. O'yin jarayonida shashka donasining u yoki bu xususiyatlari ijrochi, ya'ni o'yinchi tomonidan o'zgartiriladi.

2-talabda foydalanuvchi bilan algoritm ijrochisi o'rtasidagi muloqot oynasi bo'lgan interfeys yaratilishi kerak. Bu interfeysdagi barcha ob'yektlarning xususiyatlarini o'zgartirish ehtiyojga qarab yoki oldindan, yoki algoritm ko'rsatmalarining bajarilishi davomida o'zgartiriladi.

4-talabda algoritmning qadamlari soni chekli bo'lishi ta'kidlangan. Demak, ijrochi chekli sondagi elementar amallarni bajarishidan so'ng, masalaning kutilgan yechimiga kelishi lozim. Ayrim hollarda bu talabning buzilishi natijasida "cheksiz sikl" effekti sodir bo'ladi va bu sikl hech qachon tugamaydi. Masalan, quyidagi buyruqlarga e'tibor bering:

1. Hisoblansin $x := 1$;

2. Hisoblansin $x := x + 1$;

3. 1-qadamga o'tilsin.

Demak, albatta, algoritm qurayotganda bunday vaziyatlarni nazarda tutish lozim.

5-talab bo'yicha shuni aytish mumkinki, amal yoki metodlarda qatnashayotgan o'zgaruvchilarning qiymatlarini oldindan aniqlash masalaning to'r'r'i hal qilinishiga olib boradi. Bunday o'zgaruvchilarning boshlang'ich qiymatlarini ko'rsatmaslik ularni har xil talqin qilinishiga olib keladi. Masalan, ijrochi qiymati oldindan aniqlanmagan x o'zgaruvchi uchun $y := 2 * x - 1$; buyrug'ini qay tarzda bajarishi lozim?

6-xossaga ko'ra, algoritmning qadamlarini tashkil qilishda "biror sonni o'ylang" yoki $x_{1,2} = \pm \sqrt{\frac{d - 4ac}{2a}}$ ko'rinishidagi buyruqlardan foydalanish mumkin emas.

8-talabda algoritmning ommaviy bo'lishi ta'kidlangan. Bu shuni anglatadiki, birinchidan, ob'yektlı algoritm mumkin bo'lgan barcha boshlang'ich qiymatlarnı qayta ishlashga mo'ljallangan bo'ladi, ikkinchidan, algoritm qo'yilgan masalaning umumiy holi uchun quriladi hamda uni shu masalaning ixtiyoriy xususiy holi uchun qo'llash mumkin bo'ladi, uchinchidan, shu masala bilan bitta sinfga kirgan masalalar uchun ham qo'llash mumkin bo'ladi.

Ob'yektlarning xususiyatlarini ikki xil usulda ko'rsatilishi mumkin.

I. Algoritmning xususiyatlarini va ko'rsatmalarini bajarishni oldindan belgilab qo'yish. (1-misolga qarang.) Ko'rinib turibdiki, uyni isitish uchun isitgichning xususiyatlarini belgilab qo'yish talab qilinadi.

2) Algoritmning xususiyatlarini bajarish vaqtida o'zgartirish. Zarur bo'lganda ob'yektning u yoki bu xususiyatini "eski" qiymatini "yangi" si bilan almashtirib qo'yishga ehtiyoj paydo bo'lishi mumkin. Masalan, isitgich ishlab turgan vaqtda uy haddan tashqari isib ketgan bo'lsa, uning ish rejimi qiymatini

Ish_rejimi:=o'rta (yoki Ish_rejimi:=kuchsiz) ; (*)

tarzida o'zgartirib qo'yish mumkin.

Ko'pincha, berilgan masalalarni hal qilish uchun ularning xususiyatlarini to'g'ridan-to'g'ri o'zgartirishning iloji bo'lmaydi. Ularning qiymati qandaydir amallarning natijasiga bo'liq bo'lib qoladi. Demak, ob'yektlı algoritmlarda qo'shimcha ravishda qandaydir o'zgaruvchilarning qiymatlari kiritilishi, ijrochi tomonidan tanlanishi yoki ko'rsatiladigan amallar yordamida hisoblab topilishi mumkin. Bundan tashqari, zarur bo'lgan ayrim ma'lumotlarning ko'rsatilishini talab qilish mumkin. Bu maqsadlar uchun ob'yektlı algoritmlarning maxsus kiritish va chiqarish oynalaridan ehtiyojga qarab foydalanish mumkin. Bu oynalardagi ma'lumotlar matn shaklida bo'oladi. (*) buyruqni kiritish oynasidan foydalanib,

1. Kiritilsin (a);

2. Ish_rejimi:=a;

k'rinishida ham yozish mumkin.

Zarur bo'lganda, kiritilgan matn shaklidagi ma'lumotlarni maxsus funksiyalar

yordamida sonli yoki boshqa tipdagi ma'lumotlarga aylantirish mumkin. Bunda, mos ravishda **son(x)** yoki **matn(x)** metodlaridan foydalaniladi.

3-misol. $ax^2+bx+c=0$ kvadrat tenglamani yeching.

Yechisk. Bu masalani hal qilish uchun a , b va c o'zgaruvchilarning qiymatlarini kiritilishi lozim. Ularni kiritish uchun matnli ma'lumotlarga moslashtirilgan **Kiritilsin** oynasidan foydalaniladi.

1. Kiritilsin a.matn;

2. Kiritilsin b.matn;

3. Kiritilsin c.matn;

4. $a1:=son(a.matn);$

5. $b1:=son(b.matn);$

6. $c1:=son(c.matn);$

7. $D:=b^2-4ac;$

8. agar $D \leq 0$ bo'lsa chiqarilsin("yechim yo'q"); 13 ga o't;

9.
$$r = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

10.
$$r = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

11. Chiqarilsin $\text{matn}(x, y)$;
12. Chiqarilsin $\text{matn}(j, C_j)$;
13. Tamom.

Ayrim hollarda berilgan masalani yechish uchun bajarilishi talab qilingan amallarning soni juda katta bo'lishi mumkin. Turgan gapki, bu holda algoritm qurish jarayoni murakkablashib ketadi. Yuzaga kelgan noqulaylikning oldini olish uchun algoritmni blokli usulda qurish tavsiya qilinadi. Blokli usulda berilgan masalani bir nechta kichik-kichik masalalarga (bloklarga) ajratiladi va bu bloklarni bir-biriga masalaning xarakteridan kelib chiqqan holda bitta mantiqiy ketma-ketlikka birtashtirish mexanizmi ishlab chiqiladi. Har bir blok uchun algoritm yoziladi va shunday keyin bu bloklarni ishlab chiqilgan ketma-ketlik asosida birlashtiriladi.

4-misol Uchburchak uchlarining koordinatalari berilgan bo'lsin. Koordinatalar boshi shu uchburchak ichida yotadimi?

Berilgan masala uchun mantiqiy ketma-ketlik quyidagicha quriladi:

1. Berilgan uchburchakning yuzi S_b bo'lsin;
2. Koordinatalar boshi bo'lgan $(0, 0)$ nuqta uchburchakning uchlari bilan birlashtirilsin;
3. Hosil bo'lgan uchburchaklarning yuzalari hisoblanib, mos ravishda S_p , S_2 va S_t bilan belgilansin;
4. Agar $S^p + S^2 + S^t = S_b$ bo'lsa chiqarilsin ("ha"); aks holda chiqarilsin ("yo'q");
5. Tamom.

Tashkil qilingan bloklarning algoritmilarini qurish juda ham sodda bo'lgani uchun ularni keltirmaymiz.

Ob'ektlilik algoritmlardan eng zamonaviy dasturlash tillari bo'lgan ob'ektlilikga yaratilgan dasturlash tillarini o'qitishda keng foydalanish mumkin.

Adabiyotlar

Laripov M.M., Otaxanov N. A. DELPHI dasturlash tili. Namangan, 2007. 492 b.

2. Byn T. O'zbekiston Respublikasi Prezidentining 2017-yil 12-avgustdagi qaroriga asosan qabul qilingan "O'zbekiston Respublikasida dasturlash tillari to'g'risida"

нрНМеНеННН. М., КоНКОра, 1992.

3. Н. ВНТЛТ. 1989. АнропнТМhi н сТрыiaуphi aaHHwx. МокКBa, Мnp, 360 СТр.

4. КНырJT МсКyccTBonporpaMMHpOBaHHfl. М., Мnp, 1991.