USLUB KO'RSATKICHLARI VA DASTURLARNING TUSHUNARLILIGI

Shonazarov Sarvarbek
TATU DIF talabasi, sshonazarov938@gmail.com, +998942342082
Sultonov Hayotjon
TATU DIF talabasi, sultonovhayot871@gmail.com, +998936083334
Tohirov Quvonchbek
TATU DIF talabasi, anonymousdeveloper2005@gmail.com +998919651910

Dasturlash tili va dasturlarning o'zi nafaqat dasturchi va kompyuter o'rtasidagi aloqa vositasi, balki dasturchilar o'rtasidagi aloqa vositasidir. Dasturlarning o'qilishi va idrok etilishi uchun yangi talablar mavjud bo'lib, ularga rioya qilish aloqani soddalashtiradi. Ishlab chiqish guruhida, shuningdek, to'g'ridan-to'g'ri ishlab chiquvchilar ishtirokisiz dasturlarni saqlash va tuzatishga imkon beradi.Uslub va tushunarlilikning eng oddiy ko'rsatkichi F dasturining sharhlar darajasini baholashdir: $\mathbf{F= N_{kom}/N_{qat}}$

$\mathbf{N_{kom}}$- dasturdagi sharhlar soni; $\mathbf{N_{qat}}$ - manba matndagi satrlar yoki bayonotlar soni.

F ko'rsatkichi dasturning sharhlar bilan to'yinganligini aks ettiradi. Amaliy tajribaga asoslanib, odatda $F \geq 0.1$, ya'ni. Dasturning har o'n qatori uchun kamida bitta sharh bo'lishi kerak. Tadqiqotlar shuni ko'rsatadiki, sharhlar dastur matni bo'ylab notekis taqsimlangan: dastur boshida ular ortiqcha, o'rtada yoki oxirida esa etishmasligi. Bu dasturning boshida, qoida tariqasida, ko'proq "zich" sharhlashni talab qiladigan identifikatorlarni tavsiflovchi bayonotlar mavjudligi bilan izohlanadi. Bundan tashqari, dasturning boshida ijrochi, tabiati, dasturning funktsional maqsadi va boshqalar haqida umumiy ma'lumotlarni o'z ichiga olgan "sarlavhalar" mavjud. Bunday to'yinganlik dasturning asosiy qismida sharhlarning etishmasligini qoplaydi va shuning uchun yuqoridagi formula dastur matnining funktsional qismini sharhlashni to'g'ri aks ettirmaydi. Butun dastur n ta teng segmentga bo'linganda va ularning har biri uchun F_i aniqlanganda yanada muvaffaqiyatli variant:

$\boldsymbol{F_i=sign(N_{kom}/N_{qat}-0.1)}$**, bu uchun** $F = \sum_{i=1}^{n} Fi$

Quyidagi shart bajarilsa, dastur izohlari darajasi normal hisoblanadi: F=n. Aks holda, dasturning har qanday bo'lagi oddiy darajaga sharhlar bilan to'ldiriladi. Shuni ta'kidlash kerakki, dasturlarning uslubi va tushunarliligi dasturlarning hajmi va murakkabligi bilan chambarchas bog'liq. Shuning uchun, dastur ko'rsatkichlarini guruhlashning ba'zi an'anaviyligini unutmaslik kerak.

Xolsted metrikasi: $\hat{N}$ dasturining nazariy uzunligini o'lchash uchun M. Halsted taxminiy formulani kiritadi:

$$\hat{N} = h_1 \log_2 + h_2 \log_2 h_2$$

Bu yerda h1 - operatorlar lug'ati; h2 - dastur bayonotlarining lug'ati. Bunday holatlar N ning h ni o'zgartirmasdan o'zgarishiga olib keladi. M.Xolsteadning ta'kidlashicha, stilistik jihatdan to'g'ri dasturlar uchun nazariy uzunlik N ni haqiqiy N uzunlikdan baholashda og'ish 10% dan oshmaydi.h1, h2, N1 va N2 ni o'lchashni avtomatlashtirishda N ni aniqlash

qiyin emas. M lug'at h bilan dastur uzunligi uchun mos yozuvlar qiymati sifatida ishlatiladi. To'g'ri tuzilgan dasturning uzunligi N, ya'ni. h lug'atga ega ortiqcha bo'lmagan dastur mn nazariy dastur uzunligidan 10% dan ortiq chetga chiqmasligi kerak.Shunday qilib, h1, h2, N1 va N2 ni o'lchash va N qiymatlarini solishtirish va ba'zi bir dastur uchun 10% dan ortiq og'ish bilan dasturda stilistik xatolar, ya'ni kamchiliklar mavjudligi haqida gapirish mumkin. Dasturning to'g'riligi ko'rsatkichlariga tegishli bo'lgan yana bir xususiyat, Xollstedning fikriga ko'ra, dasturlash sifati L darajasi (dastur darajasi). $L = V^* / V$ bu erda V va V* aniqlanadi

$$V = N \log_2 h; \qquad V^* = h^* \log_2 h^*$$

Ushbu xarakteristikani kiritishning boshlang'ich nuqtasi dasturlashning stilistik sifatining pasayishi bilan dasturning kontent yuki va har bir komponentining kamayishi va natijada asl algoritmni amalga oshirish hajmining kengayishi haqidagi taxmindir. Buni hisobga olsak, V* potentsial hajmga nisbatan matnning kengayish darajasiga qarab dasturlash sifatini baholash mumkin. Shubhasiz, ideal dastur uchun L=1, haqiqiy dastur uchun esa har doim L<1. Ko'pincha dastur darajasini uning nazariy ko'lamini baholashga murojaat qilmasdan aniqlash foydali bo'ladi, chunki dastur parametrlari ro'yxati ko'pincha amalga oshirishga bog'liq va sun'iy ravishda kengaytirilishi mumkin. Bu dasturlash sifati ko'rsatkichining oshishiga olib keladi. M.Holstead bu taxminni faqat haqiqiy parametrlarni o'z ichiga olgan ifoda bilan taxmin qilishni taklif qiladi, ya'ni. Haqiqiy dastur parametrlari:

$$\hat{L} = \frac{2h_2}{h_1 N_2}$$

Mutaxassislar dasturlarni baholashda L xarakteristikasidan foydalanishni to'g'ri deb hisoblashadi.Xususiyatga ega bo'lgan Xelsted I xarakteristikani kiritadi, uni u ma'lum bir algoritmning intellektual mazmuni, ishlatiladigan amalga oshirish tillariga nisbatan o'zgarmas deb hisoblaydi: $I=LV$. I xarakteristikaning kiritilishi dasturni yaratishning asosiy aqliy xarajatlarini aniqlash imkonini beradi. Dasturni yaratish jarayoni shartli ravishda ketma-ket amallar sifatida ifodalanishi mumkin: 1) ma'lum algoritm taklifini tushunish; 2) algoritm taklifini ishlatiladigan dasturlash tili nuqtai nazaridan yozish, ya'ni. tegishli o'quv tilining lug'atidan qidirish, uning semantik mazmuni va yozib olish. Bu rasmiylashtirishni Halsted usulida qo‘llagan holda aytishimiz mumkinki, dasturni avvaldan ma’lum bo‘lgan algoritm bo‘yicha yozish h1 dastur lug‘atidan operator va operandlarni - katta tanlash, taqqoslash soni esa (saralash algoritmlariga o‘xshash) ~ log2h. Agar har bir namuna-taqqoslash o'z navbatida bir qancha aqliy elementar yechimlarni o'z ichiga olishini hisobga olsak, bu elementar yechimlarning murakkabligi va sonini har bir dastur konstruktsiyasining mazmun yukiga mos kelishi mumkin. Buni L xarakteristikasi yordamida aniqlash mumkin, chunki 1/L ni ma'lum bir dastur uchun yuklanish tezligiga ta'sir qiluvchi o'rtacha murakkablik omili sifatida ko'rib chiqish mantiqiy.Keyin dasturni yozish uchun zarur bo'lgan intellektual harakatlarni baholash quyidagicha o'zgarishi mumkin:

$$E = \frac{\hat{N} \log_2 \eta}{L}$$

E - dastur yozishda talab qilinadigan elementar yechimlar sonini. **E** faqat dasturlarni yozish bo'yicha dastlabki sa'y-harakatlarni yetarli darajada tavsiflaydi, chunki **E** ning qurilishi boshqa xarakterdagi intellektual xarajatlarni talab qiladigan nosozliklarni tuzatish ishlarini hisobga olmaydi.

Xulosa qilib, keling, avvalgilaridan bir oz farq qiladigan yana bir ko'rsatkichni ko'rib chiqaylik. U dastur hujjatlari uzunligidagi tebranishlarni o'lchashdan foydalanadigan baholash printsipiga asoslanadi. Boshlang'ich nuqta - bu dastur hujjatlariga qanchalik kam o'zgartirishlar va tuzatishlar kiritilsa, ishning barcha bosqichlarida hal qilinishi kerak bo'lgan vazifalar shunchalik aniq shakllantiriladi, degan taxmin. Dasturiy ta'minotni yaratishdagi noaniqliklar va noaniqliklar hujjatlardagi tuzatishlar va o'zgarishlar sonining ko'payishiga olib keladi. Aksincha, hujjat uzunligidagi bir oz o'zgarishlarga ega bo'lgan o'tish davri puxta o'ylangan g'oya, yaxshi tahlil, dizayn va aniq dastur tuzilishining tabiiy natijasidir. Ushbu o'zaro bog'liqliklar ushbu baholash usuli uchun asosiy bo'lib, ularning mohiyati quyidagicha.

FOYDALANILGAN ADABIYOTLAR:

1. Jarkova G. A. C++ da dasturlash:Universitetlar uchun darslik.-Ulyanovsk: UlGU, 2009 yil.

2.Glass R., Noiseau R. Dasturiy ta'minotga texnik xizmat ko'rsatish. - M.: Mir, 1983 yil.

3. Dasturiy ta'minotni himoya qilish: D. Grover tomonidan tahrirlangan. - M.: Mir, 1992 yil.