

**O'zbekiston Respublikasi oliy va o'rta maxsus ta'lim
vazirligi**

**Buxoro oziq-ovqat va engil sanoat
texnologiyasi instituti**

“Informatika va axborot texnologiyalari” kafedrası

Tuzuvchi: k. o'q. Narziev U. Z.

“Dasturlash texnologiyasi” fanidan

MA`RUZALAR MATNI

Buxoro – 2006 y.

Ma`ruzalar matni O'zbekiston Respublikasi Oliy va o'rta - maxsus ta`lim vazirligini Oliy o'quv yurtlari boshqarmasi tasdiqlagan namunaviy o'quv dasturi va kafedrada ishlab chiqilgan ishchi o'quv dasturi asosida tuzilgan.

Ma`ruza matni institut "Informatika va informatcion texnologiyalar" bakalavr yo'nalishi talabalari uchun mo'ljallangan.

Ma`ruzalar matni "I va AT" kafedra majlisida ko'rib chiqilgan (_ -son yig'ilishi bayoni " _ ". " _ ". 2005 yil).

Ma`ruzalar matni "Kasbiy ta`lim" fakulteti uslubiy kengashida ko'rib chiqilgan (№_, « _ ». « _ ». 2005 y.).

Tuzuvchi:

kat. o'q. Narziev U. Z.

Taqrizchi :

BuxDU "Amaliy matematika va
axborot texnologiyalari" kafedrası
dotc. Yodgorov O.O.

Mundarija

1- ma`ruza. Dasturiy ta`minotni ishlab chiqish texnologiyasi	4
2-ma`ruza. Dastur vositalari yaratishning umumiy printiplari. DV hayot tcikli. Tizimli tahlil.	11
3-ma`ruza. Dastur mahsulotini loyihalash.	14
4-ma`ruza. Uskunaviy ta`minot. Dastur vositasini tashkil qilish. Tizimli tahlil.	18
6-ma`ruza. Dastur va tizimlari arxitekturasi. Dastur qismlari tuzilishi.	25
7-ma`ruza. Dastur vositasini testdan o'tkazish. Yuqoridan pastga va pastdan yuqoriga testlash. Dastur vositalari tuzilishini umumiy qoidalari.	27
8-ma`ruza. Dastur vositasining tashqi tavsifi. Dasturlash usullari. Strukturali, modulli ob`ektga yo'naltirilan dasturlash.	30
9 - ma`ruza. Modulli dasturlash. Turli modullardan foydalanish. Modullarni yaratish.	33
10- ma`ruza. Dasturning umumiy tuzilishi va ulaning yozilishi. Izohlar.	37
11-ma`ruza. Foydalanuvchi bilan muloqotni ta`minlash. Foydalanuvchi interfeysini yaratish. Asosiy qoidalar.	42
12-ma`ruza. Dastur aniqligi. Ichki spetcifikatciya hujjatlari.	46
13-ma`ruza. Xatoliklar. Tizimli xatolik, masalani tavsiflashdagi xatolik. Algoritm tanlashdagi xatolik.	51
14-ma`ruza. Dasturning asosiy xarakteristikalar. Ishonchlik, samaradorlik, inson faktorini hisobi.	55
15-ma`ruza. Dasturning asosiy xarakteristikalar. Dasturni loyihalashda texnologik jarayoni.	59
16-ma`ruza. Dasturiy ta`minot mahsulot sifatida. Dasturiy mahsulotni loyihalash usullari va komponentlari.	63
17-ma`ruza. Dasturiy mahsulotni foydalanisga chiqarish.	68
18-ma`ruza. Dasturlarni hujjatlashtirish.	73

1- ma`ruza. Dasturiy ta`minotni ishlab chiqish texnologiyasi

R E J A :

1. Izohlar.

2. Bo'sh satrlar qoldirish. Bo'sh xonalar.

3. Fayllar nomi. Qisqartirishlar.

Adabiyotlar

1. Van Tassel D. Stil, razrabotka, effektivnost, otladka i ispytanie programm. M. , Mir, 1991.
2. Linger R. , Teoriya i praktika strukturnogo programmirovaniya M. , Mir, 1985.
3. V. V. Lipaev. "Proektirovanie programmných sredstv", M. : "VSh", 1991.
4. Foks Dj. "Programmnoe obespechenie i ego razrabotka". Per. s angl. M. : Mir, 1985.
5. S. S. Gulomov. "Iqtisodiy informatika". Toshkent 1999 y.

Tayanch so'zlar: *Izohlar, bo'sh satrlar qoldirish, bo'sh xonalar nomlash, tartibli raqamlash. O'zgaruvchilarning nomini tanlash, fayllar nomi, qisqartirishlar, operatorlarni joylash.*

Dasturlash usullari dasturni o'qilishini qulayligi bilan o'zaro bog'liq. Usullar deganda tajribali dastur tuzuvchilar to'g'ri natijali, foydalanishda qulay, qulay bo'lgan dasturlarni tuzishda foydalanadigan usul va uslublar nazarda tutiladi.

Dastur usulining eng yaxshi qoidasi - bu tajribali dasturchilar o'rtasidagi kelishishlaridir. Chunki bitta tajribali dasturchiga nisbatan 2 ta yoki 3 ta tajribali dasturchilar tuzgan dastur yaxshiroqdir.

Dastur shunday tuzilishi kerakki, uni birinchi qatorda mashina yoki EHM emas balki inson o'qib tushunsin. Chunki dastur insonlarga dasturlarni tuzishda, shakllantirishda va ularni qo'llashda kerak bo'ladi.

Dastur - bu keyinchalik ishlatishga va takomillashtirishga mo'ljallangan hujjat, algoritmlarni kodlashtirish uchun o'quv material va hakoazolardir.

Demak dasturlash tillari bizga o'qishda qulay bo'lgan dasturni yaratishni ta'minlab berishi kerak. Dastur iloji boricha algoritimni tuzilishini va mantiqni bizga tushunarliroq qilib etkazib berishi kerak.

Usullarni standartlash.

Albatta biz juda ko'p joylarda va jarayonlarda standartlarga asosan ish olib boramiz. Lekin dasturlash usullarida bunga qarshi bir necha fikrlar aytilgan. Ulardan birini keltirib o'tamiz: dasturlash usuli - bu shaxsiy fikr va didga bog'liqdir. Shuning uchun unga hech qanday chegaralanishlar qo'yilmasligi kerak.

Usullarni standartlash quyidagilardir:

Agar biror - bir narsani yaratishda 1 tadan ortiq mavjud bo'lib, tanlash sizga bog'liq bo'lsa, unda siz 1 ta usulni tanlab, oxirigacha shu usuldan foydalaning.

Ammo standartlash jarayonining o'ziga xos kamchiliklari bor:

standartlashni qo'llaganda kelajakda dasturlash yo'llarini sekinlashtiribgina qolmay, balki foydalanishda qulay va ortiqcha chegaralarga ega bo'lishi mumkin. Chunki ular yangidan - yangi vazifalarni bajarishga sizning aqlingizni bog'lab qo'yadi.

Izoh yoki sharh.

Tushuntirish izohlariga ega bo'lgan dasturlarni to'g'rilash yoki tuzatish osonroq, chunki ular dastur bilan ishlash uchun qo'shimcha ma'lumotlarni saqlaydi. Dastur tuzuvchi ko'pincha boshqa dasturchining dasturini ko'rib chiqayotgan vaqtda, xatolarni tuzatishda, yoki dastur mantiqini kuzatayotgan vaqtda juda ko'p vaqt sarflaydi. Bu holatda u o'zining me'yorlangan vaqtini juda katta qismni sarflab qo'yadi.

Izohlanmagan dastur - bu taxminimizcha dastur tuzuvchining eng katta xatosidir yoki ishqibozlik yon qarashni bildiradi. Dastur yozish jarayonida izoh berish yaxshi qoidadir. Kelajakda buni hamma dasturchilar hisobga oladilar degan umiddamiz. Izohning

yo'zishning asosiy qonuni - bu ko'proq izoh berishdir. Albatta, har bir kishi dasturdagi izohni o'qiyotib, shu dastur xaqida ko'proq tushunchaga ega bo'ladi.

Izoh berishdan maqsad - dasturni tushinishni osonlashtirishdir.

Izohlar dasturni kodlashtirganidek yaxshi o'ylagan va ishlab chiqilgan bo'lishi kerak. Izohlar loyihalash va sozlash jarayoni vaqtda va albatta keyinchalik ham kerak bo'ladi. Lekin dastur tugagandan keyin izoh berish foydasiz va maonosizdir. Izoh berishni 3 ta shakli bor:

- Kiritish izohlari
- Mundarija izohlari
- Tushuntirish izohlari

Kiritish izohlari.

Xar bir dastur, kichik dastur va jarayonlar nima vazifani bajarishligi xaqidagi izohlardan boshlanadi. Kiritish izohlaridagi ma'lumotlar quyidagilardan iborat bo'ladi:

1. Dasturdan maqsad.
2. Dasturni chaqirish va undan foydalanish xaqidagi ko'rsatkichlar.
3. Asosiy massiv va o'zgaruvchilarning ro'yxati va vazifasi.
4. Kiritish - chiqarish xikidagi ko'rsatkichlar, hamma fayllar ro'yxati.
5. Foydalanilyotgan barcha qism dasturlar ro'yxati.
6. Foydalanilgan matematik uslublar va ular haqida ma'lumotga ega bo'lgan adabiyotlar ro'yxati.
7. Dasturni bajarilgan vaqti haqida ma'lumot.
8. Kerak bo'lgan xotira hajmi.
9. Foydalanuvchiga maxsus ko'rsatmalar.
10. Muallif xaqida ma'lumot.
11. Dastur yozilgan sana.

Mundarija izohlari.

Agar dastur katta hajimga ega bo'lsa, uning boshiga izoh shaklida mundarija qo'yish maqsadga muvofiqdir. Mundarija o'z ichiga dastur modulining nomini, joyini va funktsiyasini jamlaydi.

Modullar ismiga, izoh yoki funktsiyalarga ko'rsatmalar bilan ta'minlangan bo'lishi kerak.

Tushuntirish izohlari.

Dasturning izohlarsiz tushunib bo'lmaydigan qismlarga tushuntirish izohlari beriladi. Dasturni mantiqini tushunishdan oldin sonlar yoki shartli operatorlardan ilgari ularning xaqidagi ko'rsatmalardan iborat izohlar chiqib turishi kerak. Yuqori pog'onada yozilgan 10 ta qatorga 1 ta izohlash qatori o'rtacha meyor hisoblanadi. Izohlar operatorlar yaratayotgan harakatlarni ta'riflamay, shu operator guruhlarini aytib o'tish kerak. Izoh dasturi boshqacha ifodalanmasligi va ma'lum bir kerakli ma'lumotlarga ega bo'lishi kerak. Izohlar to'g'ri bo'lib, Dastur o'zgarganda o'zgaruvchan bo'lishi kerak. Dasturdagi noto'g'ri izohlar ularni yo'qligiga nasbatan ancha yomonroqdir.

Qatorlarni tashlab ketish - ko'pincha yaxshi baholanmaydigan, lekin dasturni ko'rinishini yaxshilaydigan uslubdir. 1 ta qatorni tashlaganda mantiqan bog'langan operatorlarni har birini ajratamiz. 2 ta qatorni tashlaganda dasturni asosiy mantiqiy qismlarini ajratib olishimiz mumkin bo'ladi. To'ldirmagan qatrolardan foydalanish, dasturdagi alohida bo'laklarni aniq topishni engillashtirish.

Oraliq (Probel). Dasturlash tilida oraliqlar ixtiyoriy ravishda qo'yiladi. Oraliqlardan keng foydalanish dasturini o'qishni osonlashtiradi.

Tenglashtirish va ketma - ketlikni belgilash.

Dasturni ketma - ketligini belgilashda biz undagi operatorlarni ketma- ketligi tartiblarda sodir bo'ladigan xatoliklardan forig' bo'lamiz. Dasturni boshlangich qismidan belgilab berishda, bizga uni tekshirishimizda va sozlashimizda katta yordam beradi. Chunki tartib belgi yoki sonlar bizga katta dasturda kerakli qatorlarni aniqlashimizda yordam beradi. tenglashtirilgan va ketma - ketligi belgilangan dasturlar chiroyli va o'qishga qulay bo'ladi.

O'zgaruvchilarga nom tanlash.

O'zgaruvchilarni nomini shunday tanlash kerakki, bu qismda o'zgaruvchilarni kattaliklari yaqqol bilinib turishi kerak. Dasturni o'qishi oson kechishi uchun isimni to'g'ri tanlashga ham bog'liq bo'ladi. Butun o'zgaruvchilarning nomi I, G, K, L, M, N harflaridan boshlanadi.

O'zgaruvchilar quyidagi turlarda bo'linadi: butun (tceloe), xaqiqiy (deystvitelnoe), kompleksli, belgili (simvolpnoe).

Fayllar nomi.

Fayllar bilan ishlayotgan vaqtda ularni solishtirish uchun qandaydir perefiks yoki suffiks tanlanadi. Agar har bir fayl o'zining perfiksiga ega bo'lsa, dasturni o'qish oson bo'ladi. Bu perfiks dasturni yozishda mos keluvchi maydonni aniqlab beradi va qaysi ishchi kimlar bilan maydonlar mantiqan bog'langanligini aniqlab beradi.

Dasturlarni har xil dasturdagi bir xil fayllarga bir xil nom berilishlari, shu dasturlarni o'zaro solishtirish jarayonlarini osonlashtiradi.

Standart qisqartirishlar. (Abrevyatura)

Dasturni ishlab chiqishda standart qisqartirishlarni kiritish shartdir. Chunki bular begona shaxslarni dasturingizni o'qishni osonlashtiradi. Standart qisqartirishlar dasturchiga kerak bo'lganda eski dasturlarni yangisini yozishga yordamlashadi. Standart qisqartirishlarni ishlab chiqish quyidagilardan iborat: har bir so'zning belgisi xotiraga olinishi kerak. Saqlangan ismdagi so'zlarning umumiy soni 3 dan ortmasligi kerak. Abbrevatura har doim so'zlarning bosh harfi kiritilishi shart. Unsiz harflar unli harflarga nisbatan axamyatliroq. So'zning boshi uning oxiridan ko'ra kerakliroq. Abbreviatura o'z ichiga 6 dan 15 gacha harflar olishi kerak.

Keyingi satrga ko'chirish. (Perenos).

Agar siz yozayotgan qatorga biron bir so'z sig'may qolsa, uni keyingi qatordan boshlang. Albatta ko'chirish ruxsat etiladi. Lekin u dasturni o'qishini va foydalanishni qiyinlashtiradi.

Operatorlarni joylashtirish. Ba'zi dasturlarda bitta qatorga bir necha operatorlarni joylashtirish mumkin. Lekin bu 2 ta sababga ko'ra noqulay: birinchidan, dasturni o'qish qiyinlashadi. Ikkinchidan, u paragraflarga bo'lish kabi uslubdan foydalanishni almashtiradi. Yana bir sababdan biri shundaki, sintaksis xato haqida xabar borayotgan vaqtda, qatorning raqami ham beriladi.

Alfavit buyicha ro'yxatni tartiblash.

Dasturlash tilida juda ko'p o'zgaruvchilar qatnashadi va ularni ismi buyicha aniqlash tuzuvchining o'ziga bog'liqdir. Shuning uchun tuzuvchining ishini osonlashtirish uchun ro'yxatlar tuziladi va ro'yxatlar 2 ta turga bo'linadi:

1. O'zgaruvchilar turi e'lon qilingandagi o'zgaruvchilar nomini ro'yxati.
2. Jarayon parametrlarining ro'yxati.

Alfavit buyicha tartiblashni operator argumetlari ro'yxatida jarayonlarni chaqirib tartiblash ham mumkin. Kichik ma'lumotlarni tartiblashning boshqa usuli - bu belgilarni raqamlash. Ro'yxatlar ustun shaklida bo'lishi kerak. Dasturchi ro'yxatni ustun shaklida tuzish uchun har bir o'zgaruvchilar ro'yxatidagi ismlarning eng ko'p belgilariga mos ravishda ismlar ostida joy qoldirish shart. Bunday joylashtirishlar kirish - chiqishdan tortib har bir berilganlar ro'yxatida bo'lishi kerak.

Qavslar. (Skopki).

Qavslardan to'g'ri foydalanish dasturni o'qishni osonlashtiradi, chunki operatsiyalar bajarilayotgan vaqtda bo'limlar ajralishi ma'lum bo'ladi va bunda dasturchi bir necha qavslardan foydalanishi mumkin, lekin bunda dasturni o'qish va tuzatish qiyinlashadi. Asosiy qoidalaridan biri - gumonli joylarda qavs qo'ying. Bu esa dasturni tushunarli qilib, xato qilishni oldini oladi.

Qator boshidan o'tish. Operatorlarni yozishda ularni o'rtasidagi bog'liqni ko'rsatish uchun bir satrlar boshidan o'tishlar bajariladi. To'g'ri bajarilgan o'tishlar dasturni tarkibini ko'rsatadi. Bu matnni tushunarli tilda paragraflarga guruhliy so'zlarni mantiqiy bog'lab bo'ladi. Tcikllar - o'tishlarida foydalanishdagi oddiy holat. O'tishlarda faqat tcikllarda emas, balki buyruqlarni qulay guruhlashda ham kiritiladi.

Nazorat savollari:

1. Quyidagi iboralarga izoh bering:
a) kirish izohlar; v) tushintiruvchi izohlar; va s) sarlavxali (mundarija)
2. Nima uchun dasturlar qulay o'qiladigan bo'lishlari kerak?
3. Izohlarni tez-tez berilishi shartmi?
4. Dasturda bo'sh qatorli nima maksadda ishlatiladi?
5. Fayllarni nomini ko'rsatishda nimaga e'tibor berish kerak?
6. Kirish izohlari nechta elementdan iborat bo'lishi shart?
7. Dastur tuzishda qator boshidan tashlab yozishdan qachon foydalaniladi?

8. Dasturda standart qisqartarishlarni qo'llashni afzalli

2-ma`ruza. Dastur vositalari yaratishning umumiy printsiplari. DV hayot tcikli. Tizimli tahlil.

R E J A :

- 1. Bo'limlar nomini tanlash.**
- 2. "O'kilmaydigan" dasturlar.**
- 3. Dasturchiga masalahatlar.**

Adabiyotlar

1. Van Tassel D. Stil, razrabotka, effektivnost, otladka i ispytanie programm. M. , Mir, 1991.
2. Linger R. , Teoriya i praktika strukturnogo programmirovaniya M. , Mir, 1985.
3. V. V. Lipaev. "Proektirovanie programmnykh sredstv", M. : "VSh", 1991.
4. Foks Dj. "Programmnoe obespechenie i ego razrabotka". Per. s angl. M. : Mir, 1985.
5. S. S. Gulomov. "Iqtisodiy informatika". Toshkent 1999 y.

Tayanch so'zlar: *Bo'limlar nomini tanlash. "O'kilmaydigan" dasturlar. Dasturchiga maslaxatlar, bo'sh xonalar nomlash, tartibli raqamlash. O'zgaruvchilarning nomini tanlash, fayllar nomi, qisqartirishlar, operatorlarni joylash.*

Bo'limlarning nomini tanlash.

Bo'limlarning nomi shu bo'limning maqsadini bildirib turishi kerak. Bundan tashqari, bo'limlardan foydalanishda izohlarga qulay joy ajratib beradi. Har bir bo'limning boshida shu bo'lim xaqida izoh joylashtirish kerak.

Dasturlash usullari dasturni o'qilishini qulayligi bilan o'zaro bog'liq. Usullar deganda tajribali tastur tuzuvchilar to'g'ri natijali, foydalanishda qulay, o'qishda qulay bo'lgan dasturlarni tuzishda foydalanadigan usul va uslublar nazarda tutiladi.

Demak dasturlash tillari bizga o'qishda qulay bo'lgan dasturni yaratishni ta'minlab berishi kerak. Dastur iloji boricha algoritimni tuzilishini va mantiqni bizga tushunarliroq qilib etkazib berishi kerak.

"O'qilmaydigan " dasturlar.

Dastur ishlab chiqilgandan so'ng uni javobgar shaxs, bevosita loyihalovchi tasdiqlaydi. Aniqlikka talab standart usuli so'rab o'tirilmaydi. Bunday dasturlar kirayotgan ma'lumotni qabul qiladi va chiqaruvchi ma'lumotni chiqaradi. Xuddi "oldim, berdim" kabi. Shuning uchun bu dasturlar katta o'zgarishlar sodir bo'lgunga qadar kerakdir. Bu o'zgarishlardan so'ng bunday dasturlar tashlab yuborilib yangilari paydo bo'ladi. Chunki eskisini takomillashtirishdan ko'ra yangisini loyihalash osonroq. Shuning uchun bunday dasturlarga 2 chi shaxslar qo'l qo'yishi va dasturning usuli va o'lchami, hamda ishlashiga javob berishlari maqsadga muvofikdir.

Operator IF yordamida aniqlanadigan operatorlarga qator boshidan bir o'tishlar bilan yoziladi. Kiritish - chiqarish operatoriga yozishda ham o'tishlardan foydalaniladi. teng hukukli fayllarni taoriflashda ham o'tishlardan foydalaning.

Agar dasturchi tuzayotgan dasturni biron - bir boshqa dasturchi yoki kishini o'qishini bilsa, u shu dasturni kodlashdagi fikri keskin o'zgaradi. U dasturni dasturlarga bo'lib, izohlar bilan to'ldiradi. Natijada dastur aniq va tushunarli bo'ladi.

Bir xilgi loyihalarda dasturni oddiyligini ta'minlovchi uslub dasturni nazaorat qiluvchi tizimdan foydalaniladi.

Bu tizimda dastur loyihalanayotgan paytda kamida 2 ta dasturchidan foydalaniladi.

1 chisi dastur tuzsa, 2 chisi shu dasturni tuzish uchun ko'rib, o'qib chiqadi. Bu tizimni 2 ta yutug'i bor. Birinchidan u oddiy va tushunarli dastur ishlab chiqishni ta'minlaydi. Ikkinchidan, birinchi dasturchi loyihani tugata olmasa, ikkinchi dasturchi uni o'rniga tugatib qo'yadi.

Dasturni standartlarini bajarishlikka baholash uslubi ham shunga o'xshash uslubdir.

Xulosa.

- Keragidan ko'proq va yana ko'proq izoh bering.
- Kiritish izohlaridan foydalaning.
- Katta dasturlarda mundarija qiling.
- Izohlar boshini aylantirmasdan qo'shimcha ma'lumotlar bersin.
- Noto'g'ri izohdan ko'ra, uning yo'qligi yaxshi.
- Dasturni oson o'qilishi uchun oraliqlar bajaring.
- Fayllarni nomlashda prefiks va suffikslardan foydalaning.

- Alvafit bo'yicha ro'yxatlar tuzing.
- Dastur tarkibini aniqlash uchun o'tishlar hosil qiling.
- Ma'lumot tarkibini aniqlashda ham erinmang. Qator boshidan o'tishlardan foydalaning.

Nazorat savollari:

1. Kirish izohlari nechta elementlardan iborat bo'lishi kerak?
2. Dastur tuzishda qator boshidan tashlab yozish qaysi vaqtlarda amalga oshiriladi.
3. Dasturda standart qisqartirishlarni qo'llanish afzalligi qanday?
4. Nima uchun dasturlar qulay o'qiladigan bo'lishlari kerak?
5. Izohlarni tez-tez berilishi shartmi?
6. Dasturda bo'sh qatorli nima maksadda ishlatiladi?
7. Fayllarni nomini ko'rsatishda nimaga e'tibor berish kerak?
8. Kirish izohlari nechta elementlardan iborat bo'lishi kerak?
9. Dastur tuzishda qator boshidan tashlab yozish qaysi vaqtlarda amalga oshiriladi?

3-ma`ruza. Dastur maxsulotini loyihalash.

R E J A :

1. Masalani tavsiflash.
2. Algoritimni tanlash.
3. Ma`lumotlarni tavsiflash.
4. Dasturlash tilini tanlash.

Adabiyotlar

1. Van Tassel D. Stil, razrabotka, effektivnost, otladka i ispytanie programm. M. , Mir, 1991.
2. Linger R. , Teoriya i praktika strukturnogo programmirovaniya M. , Mir, 1985.
3. V. V. Lipaev. "Proektirovanie programmných sredstv", M. : "VSh", 1991.
4. Foks Dj. "Programmnoe obespechenie i ego razrabotka". Per. s angl. M. : Mir, 1985.
5. S. S. Gulomov. "Iqtisodiy informatika". Toshkent 1999 y.

Tayanch so'zlar: *Masalani tavsiflash, algoritimni tanlash, ma`lumotlarni tavsiflash, dasturlash tilini tanlash, bo'sh xonalar nomlash, tartibli raqamlash. O'zgaruvchilarning nomini tanlash, fayllar nomi, qisqartirishlar, operatorlarni joylash.*

Dasturni loyihalash bosqichi dasturlash usuliga, ishonchligiga, sozlashga, foydalanish tarkibiga, va xakazolarga ta`sir ko'rsatadi.

Dasturning loyihalashning oson kechishi - bu dasturni oson o'qilishi uchun quyilgan birinchi qadam.

Kodlashtirish oddiroq bo'lsin.

KISS deb nomlanuvchi printcipga rioya kilinsin.

Ortiqcha shakllantirilgan dastur sozlashda yoki o'zgartirishda qimmatga tushadi. Dasturdan faqat uning muallifi emas balki boshqa shaxslar ham foydalanib, ham sozlashda qiynalmasligi kerak. Dasturning tarkibi uning mantiqni ko'rsatishi shart, kodlarning doimiyliigi dasturni ya`nada osonlashtiradi. Masalan: dasturni o'zgarturuvchilar doimo bir

xil ravishda foydalanish kerak. "1" belgisi o'qilgan, "0" belgisi o'chirilgan. Bundan tashqari dasturdagi har bir jadval bir xil ta'svirlansin.

Dasturni o'qish.

Bir xilgi loyihalarda dasturni oddiyligini ta'minlovchi uslub dasturni nazorat qiluvchi tizimdan foydalaniladi.

Bu tizimda dastur loyihalalanayotgan paytda kamida 2 ta dasturchidan foydalaniladi. 1 chisi dastur tuzsa. 2 chisi shu dasturni tuzish uchun ko'rib, o'qib chiqadi. Bk tizimni 2 ta yutug'i bor. Birinchidan u oddiy va tushunarli dastur ishlab chiqishni ta'minlaydi. Ikkinchidan, birinchi dasturchi loyihani tugata olmasa, ikkinchi dasturchi uni o'rniga tugatib qo'yadi. Dasturni standartlarini bajarishga baholash uslubi ham shunga o'xshash uslubdir.

Masalani qo'yilishi va tarkibi.

Loyihalash vatida har - xil muammolarni yuzaga kelishini hisobga olish kerak. Asosan bu muammolar shu dasturni buyuruvchilarning aniq maqsadga ega bo'lmasliklaridan kelib chiqadi. Foydalanuvchi noaniq bergan so'rovga aniq va to'g'ri javob beruvchi dastur - bu eng yuksak dasturdir. Vazifa yaxshi yoki yomon deb aniqlanishi mumkin. Yomon shakllantirilgan vazifaga aniq bir dasturni loyihalab bo'lmaydi.

Aniq belgilangan vazifa esa loyihalash davomida ortiqcha ishdan vaqt sarflashdan voqif bo'ladi. Vazifa dasturning hajmiga qarab ixcham va tushunarli bo'lishi kerak. Agar dastur kichkina bo'lsa vazifa bir necha qatordan, katta bo'lsa butun - bir kitobdan iborat bo'lishi mumkin. Bu tavsiflar katta songa ega bo'lgan hujjatlarga asos bo'ladi. Keyinchalik vazifalarga kiritilgan o'zgartirishlar yoki qo'shimcha vazifalar dasturni bugibgina qolmay uni tugatishni umuman sekinlatib qo'yadi. vazifani aniqlashda qo'yilgan loqaydliklar keyinchalik katta muammolar keltirib chiqaradi.

Algoritm tanlash.

Algoritmni to'g'ri tanlash bu dasturni to'g'ri va samarali bo'lishiga qo'yiladigan asosiy qoidadir. Bunda dasturni tilini va tarkibini to'g'ri tanlash mumkin. Shunday qilib

yaxshi algoritm kerakli, lekin dastur uchun etarli sharoit emas. Yaxshi algoritmni qanday tanlash kerak?

Birinchidan - miyaga to'g'ri kelgan algoritmgaga dastur tuzmang. Bir necha variantlarni ko'rib chiqib ularning ichidan eng yaxshisini tanlang.

Ma'lumotlarni tavsiflash.

Yaxshi tavsiflangan ma'lumotlar dasturni ancha qisqartiradi.

Demak ma'lumotlar massivni tashkillashda zarur bo'lgandagina qo'llash kerak. Boshqacha usulida tayanch va ko'rsatkichlardan foydalanish kerak.

Ma'lumotlar xaqida ko'rilayotgan vazifaga mos holda ta'ssurotga ega bo'lish kerak. Hisobda har bir ma'lumotning tarkibini hamma tilga ko'chirish mumkin. Lekin siz mo'ljallangan ma'lumot tarkibini mujassamlagan dasturlash tilida foydalanishingiz makulroq.

Etarli murakkablikdagi strukturasi modullarini tuzish.

Dasturni modullarga bo'lish va yuqoridan pastga loyhalash murakkab strukturalarni nazorat qilish imkoniyatini berib, uni darajasini belgilaydi.

Modullarga bo'lish - bu murakkablik darajasini boshqarish vositasidir. Agar pastgi qismlaridagi ma'lumotlarga talab bo'lmasa ular qoldirilishi mumkin.

Birinchidan modullar o'zgaruvchilar to'plamini bo'lishga sharoit yaratadi. Buning oqibatida modullar zarur bo'lmagan o'zgaruvchilardan xalos bo'lib, kuzatilishi mumkin bo'lgan xatoliklardan ma'lum darajada himoyalanaadi. Bundan tashqari modullar qanchalik katta bo'lsa, uni tahlil qilish shunchalik qiyin. Shunday qilib modullarni kiritilishi dasturda bo'lishi mumkin bo'lgan yo'llar sonini kamaytiradi va uni boshqarishni engillashtiradi.

Dasturlash tilini tanlash.

Ko'pincha dasturlash tilini hisoblash tizimi yoki dasturchi foydalanuvchiga asosan tanlangan bo'ladi.

Agar dasturchi tanlashga ega bo'lib qolsa u albatda vazifaga mos eng yuqori dasturlash tilini tanlashi kerak. Agar tanlangan dasturlash tili berilgan vazifaga mos kelmasa dasturlashda va sozlashda muammolar kelib chiqishi mumkin. Tanlangan til loyihalashga ham katta ta'sir ko'rsatadi.

Demak dasturlash tillari bizga o'qishda qulay bo'lgan dasturni yaratishni ta'minlab berishi kerak. Dastur iloji boricha algoritmnituzilishini va mantiqni bizga tushunarliroq qilib etkazib berishi kerak.

Nazorat savollari:

1. Dasturni loyihalashtirishga tushincha bering?
2. ShEXM da axborot qanday tartibda kiritiladi?
3. Loyihalashda masalani qo'yilishi qanday ahamiyatga ega?
4. Algoritmni tanlash, dasturlash tilini tanlash dasturni loyihalashtirishda qanday ahamiyatga ega?
5. Nima uchun dasturlar qulay o'qiladigan bo'lishlari kerak?
6. Izohlarni tez-tez berilishi shartmi?
7. Dasturda bo'sh qatorli nima maksadda ishlatiladi?
8. Fayllarni nomini ko'rsatishda nimaga e'tibor berish kerak?
9. Kirish izohlari nechta elementlardan iborat bo'lishi shart?
10. Dastur tuzishda qator boshidan tashlab yozish qaysi vaqtlarda amalga oshiriladi.

4-ma`ruza. Uskunaviy ta`minot. Dastur vositasini tashkil qilish.

Tizimli tahlil.

R E J A :

- 1. Universallik.**
- 2. Kutubxonalar.**
- 3. Kiritish va chiqarish formatlari.**
- 4. Maqsadni aniqlash. Murakkablik.**

Adabiyotlar

1. Van Tassel D. Stil, razrabotka, effektivnost, otladka i ispytanie programm. M. , Mir, 1991.
2. Linger R. , Teoriya i praktika strukturnogo programmirovaniya M. , Mir, 1985.
3. V. V. Lipaev. "Proektirovanie programmných sredstv", M. : "VSh", 1991.
4. Foks Dj. "Programmnoe obespechenie i ego razrabotka". Per. s angl. M. : Mir, 1985.
5. S. S. Gulomov. "Iqtisodiy informatika". Toshkent 1999 y.

Tayanch so'zlar: *Universallik, kutubxonalar, kiritish va chiqarish formatlari, maqsadni aniqlash, murakkablik, bo'sh xonalar nomlash, tartibli raqamlash. O'zgaruvchilarning nomini tanlash, fayllar nomi, qisqartirishlar, operatorlarni joylash.*

Universallik.

Universallik deb aniq ma'lumotlar turkumiga bog'liq bo'lmagan dasturlarni ataymiz.

Parametrlar sifatida kontaktlar o'rniga o'zgaruvchilardan foydalaniladigan dasturni universal dastur deymiz. O'zgaruvchilarni kontak o'rnida qo'llanishi mumkin bo'lgan sharoitlarni ko'rsatib o'tamiz:

1. Ro'yxatlar, massivlar va jadvallar o'lchami.
2. Fizik konstanta va foizlar.
3. "Kiritish - chiqarish" moslamasini belgilanishi.

Dasturlashni universalligi loyihalashda vaqtdan yutishi va undan foydalanishiida keng sharoitlar ochib berishdir.

Universal protceduralar dastur kutubxonasiga keyingi dasturlarda ham foydalanish uchun kiritiladi.

Universallikni kidirishda asosan keyinchalik qo'llaniladigan modifikatciya va variantlarni aniqlashda ham yordam beradi.

Kutubxonalar.

Dasturni samaradorligini oshirish, sozlash va tekshirishni engillashtirish hamda dastur yaratishni qisqartirish uchun dastur kutubxonalari qo'llaniladi. Kutubxona uchun mo'lljallangan kichik dasturlar kategoriyasiga dasturlash tiliga ega bo'lgan funktciya yoki kichik dasturlarni o'z ichiga oladi. Bu kategoriyalarga barcha standart funktciyalar (sin, cos va hakoza), kiradi.

Ikkinchi funktciya va kichik dasturlar manbai sizning hisoblash tizimingizdan tashqarida joylashgan. Foydalunuvchiga u yoki bu tizimda bajarilgan dasturlar to'g'ri keladi. Kutubxona dasturlarda foydalanish sabablaridan biri - ishonchlikni oshirish va dasturlashdagi qiyinchiliklarni kamaytirishdir. kutubxona dasturlarining uchinchi manbaisi bu dasturlar va hisoblash mashinalari to'g'risidagi kitoblardir.

"Kiritish - chiqarish" formatlari.

Ma'lumotlarning "kiritish - chiqarish" formatlari loyihalash bosqichining bir qismi hisoblanadi. Kiritish formatlari foydalanuvchi uchun eng yuqori qulaylik va eng kam xatoliklar bo'lishini hisobga olgan holda bajariladi. chiqish ma'lumoti tashqi manbalar ta'sirisiz solishtirilishi kerak. U o'z ichiga:

1. chiqish yozuvini solishtirish.
2. Yozuv funktciyasi yoki tavsifi.
3. Sana.
4. Varaqlarning tartib raqamini olishi kerak.

Yaxshi o'ylab topilgan chiqish formatlari tuzuvchiga sozlashda va sinashda katta yordam beradi.

Maqsadni aniqlash.

Loyihalash vaqtida dasturchi o'z oldiga ma'lum bir aniq maqsadlar qo'yishi kerak. Quyida ularning bir nechasi keltirilgan.

1. Eng yuqori ishonchlilik.
2. Ma'lum bir ishning ayrim hajmini aniq bir sanaga bajarish.
3. Loyihalash uchun eng oz vaqt sarflash yoki tan - narxini arzonligi.
4. Foydalanishdagi qulayligi va oddiyligi.
5. Samaradorligi (xotira hajmi yoki tez ishlashi).
6. Takommilashtirishni hisobga olish.
7. Universallik.

Ayrim xollarda bitta dastur ustidan ishlayotgan ikkita dastur chi har xil maqsadlarni kuzlab ishlashadi.

Biri xotira hajmini kichkaytirishga xarakat kilsa, ikkinchisi loyihalash uchun sarf - xarajatni kamaytirishni hisoblaydi. Maqsadlarni oldindan va aniq qo'ying.

Dasturiy ta'minlash tizimsidan foydalanish vaqtida yangidan yangi maqsadlar yuzaga keladi. Yangi maqsadlarga asosan tayer dasturni o'zgartirish - pogonaga (tiqishtirish) deb ataladi.

Dasturga yangi funktsiyalar va eskisini yangilashda pogonaga qo'llaniladi.

Murakkablik.

Hammaga ma'lumki, dasturda 2 ta asosiy parametrlar: xotira va sanalari bor.

Ya'ni murakkablik parametrini ham hisobga olish kerak. Dastur qanchalik murakkab bo'lsa, uni tekshirish, nazorat qilish, sinash, tartiblash shunchalik qiyin bo'ladi. Murakkablik parametrlarini chegaralaganimizda biz boshqarishni ancha engillashtiramiz. Xozirgi vaqtda murakkablikni chegaralovchi usullar va uslublar bor. Murakkablikni boshqarish uslublari nima? Ular qanday bajaradi? Dasturlashda ularni qanday qo'llaniladi.

Sifatni oshirishning 2 ta yo'li bor: biri kattik va chukur tekshirishlar o'tkazish, ikkinchisi har bir loyihalash jarayonida tizimlarga eng yuqori ishonchlilik bilan ta'minlash.

Murakkablikni boshqarish uslubida asosan jarayon yoki tarkib bir necha parametrlarga bo'linadi va aniq bir funktsiyani bajarish uchun foydalaniladi. Bunday uslublar barcha ishlab chiqarishda, administrativ ishlarda va tashkilotlarda foydalaniadi. Murakkablikni boshqarish dasturlashdagi asosiy funktsiyalardan biridir.

Nazorat savollari:

1. Katta hajmdagi dasturlarni nima uchun dasturlashtirish murakkabroq?
2. Unversallik deganda nimani tushunasiz?
3. Kutubxonalardan dasturni loyihalashda foydalanish.
4. “Kiritish- chiqarish” formatlariga tushuncha bering?
5. Dasturni loyihalashtirishga tushincha bering?
6. ShEHM da axborot qanday tartibda kiritiladi?
7. Loyihalashda masalani qo'yilishi qanday ahamiyatga ega?
8. Algoritmni tanlash, dasturlash tilini tanlash dasturni loyihalashtirishda qanday ahamiyatga ega?
9. Nima uchun dasturlar qulay o'qiladigan bo'lishlari kerak?

5-ma`ruza. Dastur vositasini loyihalash. Tashki tavsifini loyihalash.

R E J A :

1. Kichik xajimdagi dasturlarni ishlab chiqarish usullari.
2. Katta xajimdagi dastur loyihalarini tadbik qilish yo'llari.
3. Dasturlar strategiyasi: Yuqoridan pastga jarayon.

Adabiyotlar

1. Van Tassel D. Stil, razrabotka, effektivnost, otladka i ispytanie programm. M. , Mir, 1991.
2. Linger R. , Teoriya i praktika strukturnogo programmirovaniya M. , Mir, 1985.
3. V. V. Lipaev. "Proektirovanie programmných sredstv", M. : "VSh", 1991.
4. Foks Dj. "Programmnoe obespechenie i ego razrabotka". Per. s angl. M. : Mir, 1985.
5. S. S. Gulomov. "Iqtisodiy informatika". Toshkent 1999 y.

Tayanch so'zlar: *Kichik hajimdagi dasturlarni ishlab chiqarish usullari. Katta xajimdagi dastur loyihalarni tadbik qilish yo'llari. Dasturlar strategiyasi: Yuqoridan pastga jarayoni.*

Dastur tuzishninig eng asosiy usullaridan bir bu strukturaliy dastur tuzishdir. Bu usulda dastur tuzish uchun uchta qism mavjud:

1. Yuqoridan pastga dastur tushish.
2. Modulli dasturlash.
3. Strukturali dasturlash.

Yuqoridan pastga dastur tuzishda dasturning yuqori qismdan boshlanadi. Dasturning asosiy qismi tuzilib, quyi qismidagi modullar esa vaqtinchalik fakt nomlari bilan atalgan proseduralar bilan almashtiriladi. Dasturni asosiy moduli tuzilib, testidan o'tkazilgandan so'ng ketma-ket vaqtincha yozilga modullarni yozish bilan dastur tuzish davom ettiriladi.

Modul programmalashtirishda dasturni mantiqiy qisimlariga bo'linadi. Bu modullar dasturda protceduralar va funktsiyalar orqali amalga oshiriladi.

Strukturali kodlash deganda har bir modulni gorizantal va vertikal qatorlarda to'g'ri nomlanishiga aytiladi. Bu usul yordamida modullardan tuzilgan dasturlar ishlaydigan testidan o'tkazishi qulay modifikattciya qilish uchun qulay dasturlar yaratish mumkin.

Dastur tuzatishning texnologik jarayoni.

Dastur tuzish quyidagi bosqichlardan amalga oshiriladi:

-Vazifani qo'yilishi. Bu bosqichda dasturchi buyurtmachi yordamida echilishi kerak bo'lgan vazifani qo'yadi. Texnik topshiriq tuziladi.

Bunda dasturning asosiy xarakteristikalar, muddatlar va ma'sul shaxslar aniqlanadi.

-Algoritimni tuzish. Dasturchi vazifani tahlil qilib kerakli bo'lgan algoritimni tanlaydi. Tanlangan algoritim to'liq tahlil qilinadi va uning blok-sxemasi chiziladi.

-Dasturlashtirish bosqichi. Dastur yaratish tili tanlanadi.

Dastur qabul qilingan algoritmda tuziladi.

-Dasturni tuzatish bosqichi.

-Dasturni testidan o'tkazish bosqichi.

Odatda dasturni testdan o'tkazishda bosqichlarga bo'lib urganiladi. Bunda har bir modulni tekshirishdan tortib, to butun tizimni yakuniy tekshirishlar kabi bosqichlarni oladi. Agar bunda biron bir ishonchli ketma - ketlikka yondashilmasa, ishonchli ta'minlovchi dastur olish juda qiyindir. Testlash strategiyasi ikkita usuldan birortasiga asosan bajariladi: odatiy quyidan - yuqoriga qarab testlash, yoki zamonaviy yuqoridan - pastga qarab testlash.

Quyidan - yuqoriga qarab testlash.

Bu usul keng tarkalgan usul bo'lib, unda eng kuyu pog'onadagi boshlangich yozilgan modullar tekshiriladi. So'ngra yuqori qatlamdagi elementlar dasturlanadi va testlanadi. Bu jarayon to yozilgan dastur butunlay yakunlanmaguncha davom etadi. Quyidan - yuqoriga qarab testlash usuli xozirgi vaqtda yuqoridan - pastga qarab testlovchi va dasturchilvr tomonidan qo'llanmayapti. Ularni fikricha bu usulda interfeys va algoritmdagi ko'pgina xatolar aniqlanmay qolib ketmokda. Bu esa dasturni qayta-qayta o'zgartirishdan so'ng buzishga olib keladi.

Ikkinchi kamchiligi esa: har xil pog'onadagi elementlarni testdan o'tkazishda yangidan yangi testlovchi moslamalarni, drayverlarni va testlovchi ma'lumotlarni talab qilmoqda. Bu esa o'z-o'zidan dasturlashda katta hajmda mehnat talab qiladi.

Yuqoridan pastga jarayoni.

Bu testlash usuli yuqoridan pastga qarab dasturlashni, yuqoridan pastga qarab kodlashni qo'shimcha bosqichi hisoblanadi. Bu usulda oldin asosiy dastur yoziladi va so'ngra past pog'onadagi loyihalanmagan elementlar urin bosuvchi dasturlar bilan almashtiriladi. Bunday skeletli dastur chaqiriluvchi dastur va har qanday malumotlar yo'qligida ham o'z ishini davom ettiradi. Bu tekshirish natijasida bazi hollarda bemani bo'lgan xatolar ham aniqlanadi. Keyingi qadam modul qo'shilishidan iborat bo'lib, unda bu modullar kiruvchi modullarni ko'paytiruvchi bo'lishi ham mumkin, - bu esa kiritish moduli, bazi bir yordamchi modul (oxirgini dasturlash tugash daqiqasiga qadar) bo'lishi mumkin. Bu tekshirishdan so'ng sinash oddiy bir sodda kiruvchi ma'lumotlar bilan o'tkazish mumkin.

Nazorat savollari:

1. Strukturali dasturlashtirishdagi qismlarni ko'rsating?
2. Loyihani qo'llash kutubxonasining maksadini ko'rsating?
3. Modulli dasturlashtirishni afzalligi nimada?
4. Dasturni loyihalashtirishga tushincha bering?
5. ShEXM da axborot qanday tartibda kiritiladi?
6. Loyihalashda masalani qo'yilishi qanday ahamiyatga ega?
7. Algoritmni tanlash, dasturlash tilini tanlash dasturni loyihalashtirishda qanday ahamiyatga ega?

6-ma`ruza. Dastur va tizimlari arxitekturasi. Dastur qismlari tuzilishi.

R E J A :

1. Amaliy dasturlashtirish.

2. Bosh dasturchining brigadasi.

Adabiyotlar

1. Van Tassel D. Stil, razrabotka, effektivnost, otladka i ispytanie programm. M. , Mir, 1991.
2. Linger R. , Teoriya i praktika strukturnogo programmirovaniya M. , Mir, 1985.
3. V. V. Lipaev. "Proektirovanie programmných sredstv", M. : "VSh", 1991.
4. Foks Dj. "Programmnoe obespechenie i ego razrabotka". Per. s angl. M. : Mir, 1985.
5. S. S. Gulomov. "Iqtisodiy informatika". Toshkent 1999 y.

Tayanch so'zlar: Amaliy dasturlashtirish. Bosh dasturchining brigadasi.

Amaliy dasturlash ham yuqoridan quyiga qarab bajariladi, bunda dastur boshida topshiriuklarni blshkaruvchi tillar operatorlari eziladi, undan keyin asosiy boshqaruvchi dastur eziladi.

Odatda loyiha "skeleti" bitta odam tomonidan bajariladi, bu ishlab chiqilgan loyihani ta`minlaydi. Modullik printcipi ishlatilganligi uchun bosh dastur kiska bo'lishi, shuningdek modul va quyi dasturlarni chaqirib bajariladigan bo'lishi kerak, buni "podigrqvayuo'iy" quyi dasturlarni yaratish yordamida amalga oshirish mumkin.

Ikkita loyihani kayta ishlash vaqtida strukturali dasturlashtirishning hamma uslublari birlashtirilib, bosh dasturchini brigadasi deb nomlanuvchi brigada yaratiladi.

Bosh dasturchini asosiy majburiyatlari quyidagilardan iborat:

- 1) texnik boshqarishda-loyiha texnik aspektini kuzatish;
- 2) xodimlarni boshqarishda-odam sonini nazorat qilish;
- 3)kontrakt shartlarni bajarishda-mijozlar bilan mulokotni bosh karish.

Qayta ishlashni tayanch kutubxonasi (KITK)-xamma strukturali dasturlarni saqlash va ishlatish uchun xizmat qiladi, bu dasturlar testdan o'tgan va tizimga qo'shilgan bo'ladi. KITK quyidagilar uchun mo'ljallangan:

1. Loyiha joriy holatini aniqlashni tashkil qilish va saqlash hamda xohlagan vaqtda undan foydalanishni tashkil qilish.

2. Loyiha xodimlarni kolgan kontselyariya ishdash ozod qilishni tashkil qilish.

Bunday tashkilot dasturlshni unumdorligini oshiradi. KITK quyidagi xususiyatlrga ega bo'lishi kerak:

1. Mashinaning ichki va tashki hujjatlarini saqlash kerak;

2. Butun loyiha davomida ishlashi kerak. Loyihalash bosqichidan to oxirgi yakunlovchi past-sinov bosqichigacha va loyiha buyicha hisobot tayerlashgacha

3. Tashki kutubxona faktgina eski tarixni saklabgina qolmay, balki loyihaning joriy holatini ham saklaydigan bo'lishi kerak.

4. Prrogrammalar uzlari berilgan formatlari, dastur ishlashi, algoritmlar va xokazo xaqida savol tugilgagnda ma'lumot berishlari kerak.

5. Iloji boricha kutubxonachi uzi hamma "progon"larni bajarishi va registratciya qilishi kerak. Bu loyiha butunligini va nazorat ostida bo'lishini ta'minlaydi.

Arxiv bo'lishi bir nechta sabablarga ko'ra foydalidir. Bunda dastur evolyutciyasini ko'rish imkoniyati yoki bu dasturning oldingi versiyalarini ko'rish imkoniyati yaratiladi.

KITK ning bitta qulayliklaridan biri shuki, bunda xohlagan faylni tiklash imkoniyati borligidir.

Nazorat savollari:

1. Amaliyot dasturlashtirishga tushuncha bering?
2. Bosh dasturchini brigadasining vazifasi
3. Birigada a`zolarining vazifalari?
4. Modulli dasturlashtirishni afzalligi nimada?
5. Dasturni loyihalashtirishga tushincha bering?

7-ma`ruza. Dastur vositasini testdan utkazish. Yukoridan pastga va pastdan yukoriga testlash. Dastur vositalari tuzilishini umumiy qoidalar.

R E J A :

- 1. Samaradorlik.**
- 2. Dasturlarni optimallashtirish.**
- 3. Kompilatciya jarayonida optimizatciyalash.**

Adabiyotlar

1. Van Tassel D. Stil, razrabotka, effektivnost, otladka i ispytanie programm. M. , Mir, 1991.
2. Linger R. , Teoriya i praktika strukturnogo programmirovaniya M. , Mir, 1985.
3. V. V. Lipaev. "Proektirovanie programmných sredstv", M. : "VSh", 1991.
4. Foks Dj. "Programmnoe obespechenie i ego razrabotka". Per. s angl. M. : Mir, 1985.
5. S. S. Gulomov. "Iqtisodiy informatika". Toshkent 1999 y.

Tayanch so'zlar: *Samaradorlik. Dasturlarni optimallashtirish. Kompilatciya jarayonida optimizatciyalash. Kichik hajmdagi dasturlarni ishlab chiqarish usullari. Katta hajmdagi dastur loyihalarni tadbiq qilish yo'llari.*

Dasturlash samaradorligi.

Dasturlashning asosiy vazifasi samarador dastur emas balki to'g'ri va ishonchli dastur yaratish. To'g'ri lekin samarasiz dasturni optimallab to'g'ri samarador dastur qilish mumkin. Lekin samarali, ammo noto'g'ri dasturni kamdan-kam hollarda to'g'rilash mumkin bo'ladi. Shuning uchun optimallashtirish dasturlashning ikkinchi bosqichi hisoblanadi.

To'g'ri dastur olish esa birinchi bosqich bo'ladi. Noto'g'ri dasturiy ta'minlash foydasiz bo'ladi. Chunki uni tushunish ko'p tomonlama mehnat sarfini talab etadi.

Ko'pincha vaqtning ko'p qismi (uning hajmiga nisbatan %) dasturni uncha katta bo'lmagan kiritik deb nomlanuvchi qismni bajarilishga sarflanadi. Bilimlarni samadorligini orkasidan quvish suiste'mol qilishga olib keladi.

Samaradorlikka munosabat.

Dasturning 3 xili mavjud bo'lib, ularning har biriga samaradorlik ham turlicha bo'ladi:

Birinchi turiga ko'p foydadanuvchi dasturlar kiradi. Bularga jarayon samaralari, ko'ramalar (komplyatorlar), tadbikiy dastur va adabiyotlarni hisobga oluvchi samaralar kiradi. Demak bular uchun samaradorlikni ularni ko'p qo'llanilishi va o'ziga xos bajarilishi sabab asosiy vazifasidir.

Ikkinchi turiga uzoq vaqt ullaniladigan ishlab chiqarish dasturlari kiradi. Bu turdagi dasturlarini usta tuzatuvchilar tuzatadilar. Dasturlarni samaradorlikka jiddiy qaralsada, ammo o'tgan dasturni ekpulatatciyasiga katta axamiyat beriladi.

Uchinchi turga dasturchilar tomonidan emas, balki ilmiy xodimlar va adminasrratorlar tomonidan tuzilgan dasturlar kiradi. Bu erda samaradorlik dastur uchun tegishli bo'lib, berilgan xotira hajmida joylashgan va qo'llanilaetgan vaqtida bajarilishi kerak.

Dasturni samaradorligini oshirish ko'pincha usular uni oson o'qilishi uchun ta'sir ko'rsatadi. Dasturni oson o'qilishiga uning samaradorligiga nisbatan ancha jiddiy qaraladi. Chunki bunday dasturlarni sozlash, modifitcirlash va undan foydalanish ancha osondir.

Optimallashtiruvchi kompyuterlar.

Samaradorlik dasturni loyihalashning ikkita bosqichida muhimdir: Ob`ekt dasturini samaradorligini yaratuvchi ko'rama juda katta hajmga ega bo'ladi. Lekin juda sekin ishlaydi. Chunki u ob`ekt dasturini optimallashtiradi.

Shu sababli xozirgi 1 ta mashina 2 tadan komplayatorlar har bir kirish tili uchun ishlaydi.

1 - chi komplyator tez ishlaydi, lekin samarasiz ob`ekt dasturlarini yaratadi. Bu komplyator dasturiy sozlashda ishlatiladi.

2 - chi komplyator sekin ishlaydi ammo dasturni optimllab samarali ishlab chiqaradi. Bu komplyatorlarda ob`ekt modullarini yaratilishiga foydalidir.

Programani oz hajmi bilan ishlash *samaradorlik* deb ataladi. Lekin samaralorlik 2-bosqich bo'lib, birinchi bosqichda dasturni to'g'ri ishlashini tashkil qilish kerak.

Inson faktorini hisobga olish.

Dasturni foydalanuvchi ishlatganligi munosabati bilan inson faktorini hisobga olish zarur. Shuning uchun foydalanuvchining faktorlarini ham hisobga olish kerak. Progaramma "do'stona" bo'lishi kerak. Ekranlar o'ta tez almashishi, ma'lumotlar tez chiqib turishi ham insonga salbiy ta'sir ko'rsatadi. Foydalanuvchi interfeysi (ekraniga) katta e'tibor berish kerak. Ekrandagi ranglar soni ham uncha ko'p bo'lmasligi kerak. Ranglar ham o'z o'rnida ishlatilishi kerak. Shriftlar ham o'ta katta yoki o'ta kichik kichik bo'lmasligi kerak.

Dasturni baholash imkoniyati yaratilishi kerak.

Modifikatsiya kilinishi.

Asosiy xarakteristkalaridan biri bo'lib, dasturga o'zgartirishlarni ishlab chiqarish jarayonida amalga oshirishga aytiladi. Dasturni qayta yozmasdan qandaydir qismlarini o'zgartiriladi.

Yaxshi tafsiflangan ma'lumotlar dasturni ancha qisqartiradi, demak ma'lumotlar massivini tanlashda zarur bo'lganicha qo'llash kerak. Boshqacha usulda tayanch va ko'rsatkichlardan foydalanish kerak. Ma'lumotlar xaqida ko'rilaetgan vazifaga mos xolda taosurotga ega bo'lish kerak. Hisobda har bir ma'lumot tarkibini hamma tilga o'zgartirish mumkin. Lekin siz mo'ljallagan ma'lumot tarkibini mujassamlagan dasturlash tilidan foydalanishingiz ma'qulroq.

Dasturiy foydalanuvchilar ehtiejlarini qondirish, keng tarqatish va sotish uchun mo'ljallangan.

Hozirgi paytda dasturiy maxsulotlarni ochiq (legal) tarqatishning boshqa variantlari ham mavjud, ular yalpi (global) va mintaqaviy kommunikatsiyalardan foydalanish bilan yuzaga keladi.

Nazorat savollari:

1. Dastur tuzishda qaysi bosqichda dasturning samaradorligi muxim?
2. Qanday vaqtda dasturni optimallashtirish kerak?
3. Dasturni optimallashtirish uchun qaysi usullardan foydalaniladi?
4. Dasturni optimallashtirishda asosiy mezonlarni ko'rsating?
5. Modulli dasturlashtirishni afzalligi nimada?
6. Dasturni loyihalashtirishga tushincha bering?

8-ma`ruza. Dastur vositasining tashki tavsifi. Dasturlash usullari. Strukturali, modulli ob`ektga yunaltirilgan dasturlash.

R E J A :

- 1. Tcikllarni olib tashlash.**
- 2. Tcikllarni almashtirish.**
- 3. Ogohlantiruvchi xabarlar.**

Adabiyotlar

1. Van Tassel D. Stil, razrabotka, effektivnost, otladka i ispytanie programm. M. , Mir, 1991.
2. Linger R. , Teoriya i praktika strukturnogo programmirovaniya M. , Mir, 1985.
3. V. V. Lipaev. "Proektirovanie programmných sredstv", M. : "VSh", 1991.
4. Foks Dj. "Programmnoe obespechenie i ego razrabotka". Per. s angl. M. : Mir, 1985.
5. S. S. Gulomov. "Iqtisodiy informatika". Toshkent 1999 y.

Tayanch so'zlar: *Tcikllarni olib tashlash. Tcikllarni almashtirish. Ogohlantiruvchi xabarlar, Kichik hajmdagi dasturlarni ishlab chiqarish usullari. Katta hajmdagi dastur loyihalarni tadbiq qilish yo'llari.*

Tcikllarni olib tashlash (o'chirish).

Iloji boricha tcikllardan foydalanmaslik kerak. Chunki ularni ishlatilganda bajarish vaqtini uchdan bir qismiga oshiradi. Tcikllarni ikkita va undan ortigini bitta tciklga joylashtirish orkali ularni sonini kamaytirish kerak. Tcikllarni dasturdagi vazifalari yaxshilab tekshirilib chiqilgandan so'ng birlashtirish imkoni bo'ladi.

Tcikllarni kamaytirish.

Tcikllarni kamaytirish yo'llari bilan ham dasturni optimallashtirish mumkin. Chunki tcikllarni bajarishga (parametrni qo'shishga va tekshirishga) ko'p vaqt sarflanadi. Tcikllarni ishlatish dasturni bajarilishini vaqtini birdan uchga ko'paytiradi. Shuning uchun hisoblashda tcikllarni kamaytirib dasturni tuzish kerak.

Tcikllarni bajarilishida ko'proq vaqtni tciklni ishga tushirish va uning indeksini tekshirishga ketadi. Ichma-ich joylashgan tcikllarni to'g'ri tashkil qilish bilan vaqtini kamaytirish mumkin.

Tcikllarni optimalashtirish.

Dasturni tezroq bajarishda tcikllarni bajarish vaqti asosiy faktor hisoblanadi. Ma'lumki tcikl ichidagi operatorlar bir necha ming marotaba bajariladi. Shu bajarilishda ozgina samaradorlik ham bir necha mingga ko'payadi.

Ichma-ich joylashgan tcikllrada optimalashtirishni ichki tcikl operatorlaridan boshlash kerak. Tcikllar bajarilish vaqtini kamaytirish va ishlatish xotirasini kamaytirish maqsadida ketma-ket yozilgan bir necha tcikllarini bittaga keltirish ishlatiladi.

Indeksatciya bilan optimallashtirish.

Indeksatciyalar bilan ishlashda kompyuterni vaqti va uning xotiradagi joyi ko'proq ishlatiladi. Shuning uchun indeksatciyalarni ham optimallashtirish dasturni optimalashtirishga olib keladi.

Agar bir yoki bir necha operatorlarni ichida indeksli o'zgaruvchiga bir necha marta murojat qilinaetgan bo'lsa, u holda bu indeksli o'zgaruvchining boshqa o'zgaruvchi bilan tenglashtirib olish kerak.

Misol uchun quyidagi ifodani $x=(A(I)+1/A(I))+A(I)$, optimallashtirish uchun quyidagicha yozish mumkin:

$$AI=A(I)$$

$$x=(AI+1/AI)+AI$$

Ogohlantiruvchi xabarlar.

Ko'pincha kompolyatciyani samaradlorligini oshrishda ahamiya bergan holda va dasturni bajarishda ma'lum bir shartlarni ogohlantirish xabarlarini keltirib chiqaruvchilar olib tashlanadi. Unga sabab ogohlanturuvchi xabarlar dasturni boshqarishda xech qanday xalakit bermaydi va samaradorlik pasaytirishi sababli ularni hisobga olinmaydi. Har safar ko'ramalashni bajarishga bunday xabarlarini chaqirishda ularga ancha vaqt sarf bo'ladi. Bu esa dasturni bajarishda va xotira sarfini oshiradi.

Ko'ramalashga ogohlantiruvchi xabarlarni qo'llamagan xolda dastur yaratishni oson usuli bu dasturni samaradorligini oshirgan holda bajarishdir. Ogohlantiruvchi xabarlar ko'pincha to'g'irlashda va qayta ko'rishda beriladi.

Nazorat savollari:

1. Tcikllarni qo'llanishini samaradorligi ?
2. Tcikllarni qanday optimallashtirish mumkin?
3. Tcikllar dasturda qanday tartibda joylashishi mumkin?
4. Modulli dasturlashtirishni afzalligi nimada?

9 - ma`ruza. Modulli dasturlash. Turli modullardan foydalanish.

Modullarni yaratish.

R E J A :

1. Yuklash modullari. Modullar.

2. Kompilyatorva mashina xaqidagi axborotdan foydalanish.

3. Yuklovchi modullar.

Adabiyotlar

1. Van Tassel D. Stil, razrabotka, effektivnost, otladka i ispytanie programm. M. , Mir, 1991.
2. Linger R. , Teoriya i praktika strukturnogo programmirovaniya M. , Mir, 1985.
3. V. V. Lipaev. "Proektirovanie programmnykh sredstv", M. : "VSh", 1991.
4. Foks Dj. "Programmnoe obespechenie i ego razrabotka". Per. s angl. M. : Mir, 1985.
5. S. S. Gulomov. "Iqtisodiy informatika". Toshkent 1999 y.

Tayanch so'zlar: *Yuklash modullari. Modullar. Kompilyator va mashina xakidagi axborotdan foydalanish.*

Kompilyatorlash natijasida olinadigan dastur ularni bajarish dasturlash buyumlarida ishlatilishi kerak. Dastur sozlanib testdan o'tkazilgandan so'ng, uni keyinchalik ish bajarish uchun yuklovchi modul yaratish kerak.

Diskdagi yuklovchi modullardan foydalanib aloqa o'rnatish va ularni kompilyatorlash jarayonini hisobga olmasdan ancha vaqt o'zgartirishda boshlang'ich dasturni har bir satrini ko'rib o'tishda kompilyatorlar murakkab jarayon bo'lib, optimallashtirishda unga qo'llanilmaydi.

Modullar.

Modullarni yaratish dasturni bajarish tartibini, uni sozlash, testdan o'tkazish va takomillashtirishni ancha osonlashtiradi. Bu esa dastur yozishni boshidanoq dasturlash vaqtidan va mashina vaqtidan yutishga olib keladi. Bu esa dasturchi uchun juda muxim,

chunki dasturni boshlangich vaqtidanoq mashina vaqtidan dasturning bajarilishi jarayoniga nisbatan ancha tayerlanadi. Dasturni loyihalash, sozlash, testdan o'tkazishni bir - biriga bog'liq bo'lmagan holda mustaqil bajarish mumkin. Natijada dasturchi mashina vaqtidan yutadi. Dasturdan foydalanishdagi kamchiliklardan biri uning hajmi katta bo'lishiga qaratiladi. Agar dasturni chiqarish nuqtaga qo'yib, dasturni biron - bir joyidan buyruq berilsa, u buyruqni samarali bajaradi.

Dasturni ketma-ketlik bilan bajarishda aloqa buyrug'i uzib qo'yiladi va buning oqibatida xotira va vaqt meyorlanadi.

Modulni dasturlash - bu dasturni modul deb ataluvchi mantiqiy qismlarga ajratish jarayonidir. Bu qismlar aloxida ketma - ket loyihalanadi, chunki katta masala bir necha kichik masalalarga bo'linsa uni hal etish yoki echish ancha engillashadi.

Agar masalani yuqoridan quyigacha loyihalansa, unda u albatta taxminiy modullar uchun bir necha kichik vazifalarga bo'linadi.

1). Modul qo'llanilgan konteksta bog'liq bo'lmagan, xolda, to'g'ri va aniq dasturiy modulga erishish kerak.

2). Oldindan modulning ichki ishi to'g'risida xech qanday ma'lumotga ega bo'lmagan xolda modul yordamida katta dasturni formalashga intilish lozim.

Tadbiqiy dasturlash va standart jarayonlar qulay modullarga misoldir.

Buni ham yuqoridan pastga qarab bajariladi. Faqat unda vazifalarni boshqarish tilini biladi. Odatda loyiha "tanasini" ni bir kishi bajarish mumkin. Bu esa loyihani butunligini ta'minlaydi va dasturni tushunish oson bo'ladi.

Modulni printcipi qo'llanilaetganligi sababli asosiy dastur qisqa bo'lish va ko'maklashuvchi kichik dasturlarni yaratuvchi va modullovchi kichik va modullarni chiqaruvchi bo'lishi kerak.

Ko'maklashuvchidastur haqiqiy dastur yaralgunga qadar o'rnini bosuvchi qisqa buyruqlar ketma - ketlikdan iborat.

Ko'maklashuvchi dastur 2 ga bo'linadi.

Soxta va almashinuvchan modullar.

Soxta modul xech qanday ish bajarilmaydi, u faqat chaqiruvchi modullarga boshqaruvni kaytarib beradi. Vaqtincha almashinuvchi modulni murakkab modul yaratulgunga qadar oddiy qayta ishlashni bajaradi.

Almashinuvchi modul natijani bermagunga qadar chiqaruvchi modul ishini davom ettirmasligi kerak. Ko'maklashuvchi dasturlar dasturning boshqa bo'g'inarini (segmentga) testdan o'tkazish uchun ham xizmat qiladi.

Bosh dasturni 2 ta muvaffakiyatli loyihalar ishlab chiqarilgandan so'ng tarkibiy dasturlarning hamma usullari birlashtirilib BDB (bosh dastur birgadasi) tuziladi.

1. Texnik boshqarishda loyihani texnik ekspertlarni nazorat qilish.
2. Qo'l ostidagilarni boshqarishda odamlarni hisobotligini va tartibligini nazorat qilish.
3. Shartnoma shartlarini bajarishda buyuruvchi bilan muomolani saqlab turish.

Loyihani qo'llab - quvatlovchi kutubxona (BPR) tizimga birlashtirilgan va testdan utkazilgan brigada tarkibiy dasturlarini foydalanish uchun hamda saqlash uchun xizmat qiladi.

Ko'p masalalarda tez - tez qo'llanilib turadigan modullar, standart dasturlar bir xil qoida asosida tuziladi (rasmiylashtiriladi), ya'ni ularga murojat qilish, foydalanish va ulardan natija olish bir xil qoida asosida tashkil etiladi. Bu esa dastur tuzishda ulardan osonlik bilan foydalanish imkonini beradi.

Modullardan ikki xil usulda: asosiy dasturning zarur bo'lgan qismiga modullarni joylashtirish yo'li bilan: har bir mashinaning o'ziga xos buyruqlari mavjud bo'lib, bu buyruqlar yordamida modullarga murojat qilish yo'li bilan foydalanish mumkin. Ikkinchi usul ko'proq qo'llaniladi. Chunki bu xolda operativ xotirada modullar kutubxona shaklida joylashadi va bu modullardan istilgan vaqtda foydalanish mumkin. Shu sababli modullarni ishdan ozod qilish uch xil adres qo'llaniladi:

1. Absolyut adreslarning qiymatlari modullarning tutgan o'rniga qarab o'zgarmaydi;
2. Ichki adreslar modullarning joylashgan eriga bog'liq holda hisoblanadi;
3. Tashqi adreslar boshqa modullarning tutgan o'rniga qarab o'zgaradi;

Absolyut adreslar o'zgarmas ish yacheykalari yoki mashina registrari, ba'zi bir maxsus mashina komandalarining adres qismlaridan iborat bo'ladi. Masalan, surish buyrug'i hamda komanda shaklida yozilgan o'zgarmas sonlarning adres qismlari.

Ichki adreslar boshqa modullarga o'tish komandalari va standart dasturning o'zida o'zgarmaslar yozilgan joyda uchrashi mumkin.

Tashqi adreslar boshqa modullarga o'tish buyruqlarida uchraydi.

Mashina xotirasida doim saqlanadigan modullar to'plami modullar kutubxonasini tashkil etadi. Kutubxona tarkibiga bir necha o'nlab modullardan bir necha yuzlab modullargacha kiritish mumkin. Umuman mashinada bajariladigan ishlarning soni dastur shaklida yozib yig'ilgan bo'ladi. Bu ishlarning ba'zi birlari mashina zimmasiga yuklansa, mashina qurilmasini murakkablashtirib yuboradi. Shuning uchun kutubxona tuzish maqsadga muvofiqdir.

Kutubxona operativ xotiraga chaqirish va ularni xohlagan shaklda joylashtirish uchun ikki xil: interpretatsiya va kompilyatsiya usullari mavjud. Kompilyatsiya usuli qo'llanilganda modullarning chaqirish va joylashtirish uchun kompilyatsiya qiluvchi tizim, interpretatsiya usuli qo'llanilganda esa interpretatsiya qiluvchi tizim ishlaydi.

Ikkala usulda ham standart dasturlar kutubxonasiga, kutubxona katalogiga va boshqaruvchi dasturga murojat kilinadi. Interpretatsiya usulning afzalligi shundan iboratki, asosiy yoki modul dasturlar hajmi jixatdan katta bo'lganda xotiradan bemalol foydalanish mumkin. Kompilyatsiya usulidan esa barcha modullar va asosiy dastur operativ xotiraga joylashgan vaqtda foydalaniladi.

Kompilyatsiya usuli bilan ishlaganda kutubxonadan barcha kerakli modullar operativ xotiraga chaqirib, jamlab qo'yiladi va dastur ishlaganda operativ xotiradagi modullardan avtomatik ravishda foydalaniladi. Kam vaqt sarflash bu usulning afzalligi, operativ xotiraning ko'p sarflanishi esa kamchiligi hisoblanadi.

Nazorat savollari:

1. Modullarga tushuncha bering
2. Yuklovchi modullar- bu nima?
3. Kompilyator va ShEXM to'g'risida ma'lumotlarni qo'llash?

10- ma`ruza. Dasturning umumiy tuzilishi va ulaning yozilishi.

Izohlar.

R E J A:

- 1. Masalalarni tavsiflash. Algoritmni tanlashdagi xato. Umumiy xatoliklar.**
- 2. Dasturlarni to'g'rilash.**
- 3. Sintaksiz xatolar.**

Adabiyotlar

1. Van Tassel D. Stil, razrabotka, effektivnost, otladka i ispytanie programm. M. , Mir, 1991.
2. Linger R. , Teoriya i praktika strukturnogo programmirovaniya M. , Mir, 1985.
3. V. V. Lipaev. "Proektirovanie programmných sredstv", M. : "VSh", 1991.
4. Foks Dj. "Programmnoe obespechenie i ego razrabotka". Per. s angl. M. : Mir, 1985.
5. S. S. Gulomov. "Iqtisodiy informatika". Toshkent 1999 y.

Tayanch so'zlar: *Masalalarni tavsiflash, algoritmni tanlash, taxminiy xatoliklari. Umumiy xatoliklar, xatosiz dasturlashtirish. Dasturlarni to'g'irlash. Sintaksiz xatolar.*

Dasturlarni sozlash.

Xatolarni borligini aniqlab ularni to'g'rilash - *sozlash* deb ataladi. Dasturlarni u yoki bu xatolarda sozlash majburiydir. Aks holda biz uni testdan o'tkazishimiz kerak bo'ladi. Jarayonni sozlash dasturni ishlash usuliga bog'liq, ya'ni foydalaniladigan mashinaga, jarayon tizimiga, dasturlash tiliga, beriladigan vazifa tarkibiga va xattoki dasturni muayyan xususiyatiga ham bog'lik bo'ladi. Yana shuni aniq aytish mumkinki, har bir xil qurilma va mashinalar dastur kamchiliklari dasturlash xatolari bilan uzluksiz bog'liq.

Masalan: Sintaksis xatolar bo'lganda dasturlashni aniq tili orqali oldindan bilib yoki aniqlab olinadi. Xozirgi vaqtda dasturlarni hajmi katta va murakkab bo'lib bormokda, lekin xatolar o'shaligicha qolmoqda.

Masalalarni tavsiflashdagi xatolar.

Odatda dastur yozib bo'lingandan so'ng foydalanuvchi olinayotgan ma'lumotlarni kerakli ma'lumotlardan farq qilishini bilib oladi.

Dasturga qo'yiladigan talabni sifatsizligi keyinchalik tayyor dasturni noto'g'ri ko'rsatilgan va berilgan vazifalarni to'g'ri ishlashiga olib keladi.

Biron bir buyuruvchi uchun loyihalashayotgan dasturni uning talablariga mos kelmasligini belgilaridan biri, quyiladigan vazifani tushunmayotganligidandir.

Shuning uchun biron bir dasturni loyihalashdan oldin buyuruvchidan qo'yiladigan talabni yozma ravishda olishimiz kerak. Bu esa buyuruvchini fikrini bir joyga to'plashga va talabni aniq, ravshan va tushunarli qilib yozib berishiga olib keladi.

Algoritmni tanlashdagi xatolar.

Vazifa oxiri aniq bo'lgandan so'ng, dasturchi unga mos keluvchi algoritm yoki echish dasturlarini qidiradi. Sifatsiz, noto'g'ri tanlangan algoritm sifatida biz vazifani to'g'ri echadigan lekin hisoblashga uzoq vaqt sarflaydigan misolni ko'rsatishimiz mumkin.

Algoritmi noto'g'riligini uni sinab ko'rilgandan so'ng aniqlash mumkin. Shuning uchun dasturni boshqatdan tekshirib chiqish oldini olish uchun algoritmgaga alohida ahamiyat berish kerak.

Tahlil qilishdagi xatolar.

Bunday xatolar sodir bo'ladigan xatolarni to'liq hisobga olmaganligi va vazifani noto'g'ri echilishida hosil bo'ladi. Birinchi holatga misol tariqasida katta va kichik kattaliklarni, o'zgaruvchilarni manfiy qiymatda hosil bo'lishiga etiborsiz qaralishi natijasida hosil bo'lishidir.

Ikkinchi holatda odatda yirik va kichik mantiqiy xatolar hisoblanadi. Ulardan:

- O'zgaruvchilarni boshlangich qiymatini vazifasini yo'qligi.
- Tcikl yakunining noto'g'ri sharti.
- Tciklni noto'g'ri indektsatciyalash.
- Initsirlashgan tciklni shartlarini vazifasini yo'qligi.

- Vazifani echish jarayonini davom ettirish uchun berilgan algoritmni shoxlarini noto'g'ri ko'rsatilishi.

Sozlashni tashkillashtirishni eng oson yo'li sozlashdan kam foydalanishga harakat qilish ya'ni xatolarga yo'l qo'ymaslikka erishish.

Umumiy ko'rinishdagi xatolar. Qancha urinmang baribir tanlangan tilga bog'liq bo'lmagan holda dasturlashda xatolarga yo'l qo'yiladi. Bularga quyidagilar kiradi:

-Dasturchi tomonidan mashinani yoki dasturlash tilini bilmasligi sababli bo'ladigan xatolar.

-Algoritmni loyihalashda, bunda dasturda foydalanilgan operatorlar algoritm tomonidan qo'yilgan ketma-ketliklarni noto'g'ri bajarishda kelib chiqadigan xatolar.

-Sintaktik xatolar.

-Sintaktik to'g'ri operatorlarni bajarishda kelib chiqadigan xatolar. Ya'ni nolga bo'lishda yoki manfiy sondan kvadrat ildiz olishda va noto'g'ri ma'lumot berishda. Barcha ko'rsatilgan xatolar sintaktik xatodan tashqari testdan o'tkazish orqali aniqlanadi. Buning natijasida dastur ustida olib boriladigan ish sozlash bosqichiga o'tadi.

Dasturdagi umumiy xatolar.

1. Vazifani noto'g'ri qo'yilishi Noto'g'ri berilgan vazifani to'g'ri echish.
2. Noto'g'ri algoritm Vazifani noto'g'ri yoki samarasiz echishga olib keluvchi algoritm.
3. Tahlildagi xato. Algoritmni noto'g'ri dasturlash.
4. Semantiq xato. Buyruqni tansiflash tartibini tushunmaslik.
5. Sintaktik xato. Dasturlash tili yordamida aniqlangan qoidani buzish.
6. Jarayon bahsdagi xatolar. Hisoblashdagi chegaralovchi shartlarga ko'rsatmalarning yo'qligi.
7. Malumotdagi xatolar. Ma'lumotda hosil bo'luvchi o'zgarishlarni noto'g'ri aniqlash.
8. Hujjatdagi xatolar. Foydalanuvchining hujjatlari xaqiqiy dasturga mos kelmasligi.

Sodda dasturlash.

Dasturni qoniqarli sozlashda kodlashni oddiyligi va to'g'riligi katta ta'sir ko'rsatadi. Shuning uchun har xil ramz belgiyu va h. k. lardan foydalanmaslik kerak. Chunki ular dasturni sozlashda katta to'sqinlar hosil qiladi. Murakkab dasturning boshlangich bosqichlarida oddiy va kichkina bloklar, uni soddalashtirish uchun qo'shib yoziladi. Dasturni qulay o'qishligi sozlashni ham osonlashtiradi.

Xamma dasturlar mantiqan olib qaraganda ayrim bir ma'lumotlar ta'sir ko'rsatuvchi xududlarni aniq ko'rsatishi, ya'ni dastur ish bajarish qobiliyatiga ega bo'lishi, malumotlarni ko'rsatilgan chegaralarda turganligini aniqlash uchun operatorlarni kiritish imkoniga ega bo'lishi kerak.

Dasturni foydalanishga berishdan oldin uni to'g'riligigi ishonch hosil qilish kerak. Noto'g'rilikni aniqlashni ikkita usuli bor:

1. Dasturni konstruktciyasi sintaktik xato.
2. Dastur noto'g'ri natijalar ko'rsatmoqda.

Sintaktik xato.

Traslyator yordamida sintaktik xatolarni aniqlash dasturni sozlashda eng kerakli va muhim o'rinlarni egallaydi. chunki qancha ko'p xato bo'lsa shu bosqichda aniqlanib to'g'rilansa, keyinchalik sozlashda va testdan o'tkazishda ish oson bo'ladi. Yana bu xatolarni aniqlashning muximligi shundaki, agar ular aniqlanmasa. keyinchalik dasturni bajarishda katta qiyinchiliklarga olib keladi.

Kompilyator uchun bu xatolar dasturni sinash uchun ko'rib o'tishdan oldin bartaraf etilishi lozim. Berilgan operatorlardagi sintaksis xatolar sifatida quyidagilarni misol qilib ko'rsatishimiz mumkin:

- Tinish belgilarini qo'yishda kerakli belgilarni tashlab ketish.
- Kelishilmagan holda qavslarni qo'yish.
- Kerakli qavslarni tashlab ketish.
- Operatorni noto'g'ri shakllantirish.
- O'zgaruvchilar nomini noto'g'ri shakllash.
- Arifmetik operatorlardan noto'g'ri foydalanish.
- Olib qo'yilgan yoki ajratib qo'yilgan so'zlarni noto'g'ri yozish.

Ikkita yoki undan ortiq operatorlarni egallab oluvchi sintaktik xatolarga quyidagilar kiradi:

- Karamaqarshi buyruq.
- Tcikl oxirini ko'rsatuvchi shartlar yo'qligi
- Belgilar yo'qligi.
- Massvlarni tavsifini yo'qligi.
- Taqiqlangan o'tishlar.

Agar komplyator ikkita yoki undan ortiq buyruqlarni birma-bir tekshirib chiqmasa unda yuqorida ko'rsatilgan xatolar aniqlanmay qolib ketadi. Bir xilgi vaqtda yaxshi sozlovchi komplyator tekshirishga ketadigan vaqtni teng yarmini meyorlab qoladi. Maxsus sozlovchi ko'ramalar oddiysiga nisbatan sintaktik xatolarni yaxshi aniqlaydi.

Chunki u o'zaro ta'sir etuvchi buyruqlarni va sintaktik tizimni birma-bir tekshirib chiqadi. Sozlovchi ko'ramalar EVM bilan birgalikda keltirilmaydi. Ularni alohida sotib olish kerak.

Agar sozlash dasturchining ishchi vaqtini 70% ni va mashina vaqtini ko'p qismni egallasa, demak, sozlovchi zarur va u tez orada o'z-o'zini qoplaydi.

Barcha asosiy dasturlash tillarini sozlovchi komplyator bilan ta'minlangan bo'ladi. Agar dasturda mos keluvchi operatorlar to'g'ri shakllantirilgan bo'lsa, u xolda komplyator ayrim xatolarni aniqlay olmaydi.

Bir qator komplyator dastur ob'ektini yordamchi bloklarini keltirib chiqaradi. (Masalan, diapazonlarni va indekslarni o'zlashtirish tekshiruvchi bloklar). Bu bloklar dasturni bajarishda ayrim tafsifga ega bo'lgan xatolarni nazorat qiladi. Bunday bloklar qanchalik ko'p bo'lsa xatolarni ushlash jarayoni shuncha oson kechadi.

Nazorat savollari:

1. Nima uchun dasturlar sozlanadi?
2. Dasturni sozlash deganda nimani tushunasiz?
3. Qanday turdagi xatoliklarga yo'l qo'yish mumkin?
4. Dasturdagi umumiy xatoliklarga tushuncha bering?

11-ma`ruza. Foydalanuvchi bilan mulokotni ta`minlash.
Foydalanuvchi interfeysini yaratish. Asosiy koidalar.

R E J A :

- 1. Sozlash turlari.**
- 2. Sozlash vositalari. Umumiy tavsiyalar.**
- 3. Interaktiv holatda sozlash.**

Adabiyotlar

1. Van Tassel D. Stil, razrabotka, effektivnost, otladka i ispytanie programm. M. , Mir, 1991.
2. Linger R. , Teoriya i praktika strukturnogo programmirovaniya M. , Mir, 1985.
3. V. V. Lipaev. "Proektirovanie programmných sredstv", M. : "VSh", 1991.
4. Foks Dj. "Programmnoe obespechenie i ego razrabotka". Per. s angl. M. : Mir, 1985.
5. S. S. Gulomov. "Iqtisodiy informatika". Toshkent 1999 y.

Tayanch so'zlar: *turlari. Sozlash vositalari. Umumiy tavsiyalar. Interaktiv holatda sozlash. Dasturlarni tekshirish uchun sozlash modullari. Dasturchiga maslaxatlar.*

Sozlash turlari.

Sintaktik xatolar xaqida xabarlar to'xtashi bilanoq sozlash boshlanadi. Sozlash jarayoni boshida oddiy test ma'lumotlaridan foydalanish kerak. Agar bunda to'g'ri natijalar hosil bo'lsa unda juda ko'p murakkab ma'lumotlar bilan ishlab ko'rish kerak. Agar natijalar noto'g'ri bo'lsa unda quyidagi hollar bo'lishi mumkin:

1. Sintaktik xato yo'q, lekin dastur kompilyatciya bo'lmagan.
2. Dastur ishlayapti, kompilyatciya bo'lgan, lekin natija chiqarmayapti.
3. Dastur kompilyatciya bo'lgan ishlayapti, lekin kutilmagan uzilishlar bo'lyapti.
4. Dastur kompilyatciya bo'lgan , ishlayapti, lekin noto'g'ri natija chiqaryapti.
5. Dastur tcikl ichida ishlab to'xtab qolmoqda.

Quyidagi har bir holatni ko'rib chiqamiz:

1 - holat. Bu holatlarni tuzatishda tizimni yaxshi biladigan xodimlardan maslahat so'rashi kerak.

2 - holat. Bu holatlar qandaydir mantiqiy va tizimli holatlar orqali bo'lgan bo'lishi mumkin. Masalan, algoritm boshlanishi bilan qiymatiga qarab ketishi mumkin.

Tuzatish yo'llaridan natija hosil qilish segmentini qayta dasturlash usulidir.

3 - holat. Dastur ko'zda tutilgandan oldinroq to'xtab qolishi ogir holatlardan biridir. Bu holatlarni topish uchun xatolarni topish usullaridan foydalanish ma'qul.

4 - holat. Bu holat dastur to'g'ri tuzilganligini, lekin unda xatolik borligini ko'rsatadi.

5- Bu holatda qaysi tcikldan to'xtash bo'lganligini topish kerak. Buning uchun gumon qilingan tcikldan oldin va keyin chiqarish operatorlarini quyib tekshirish kerak. Yuqoridagilarni hisobga olib quyidagi umumiy takliflarni berish mumkin:

- Dastur qancha ko'p yozilsa shuncha ko'p xato bo'lishi mumkin.
- tuzatish jarayonini dasturni yozish bosqichida o'ylash kerak.
- Tuzatish jarayonida ishlatiladigan o'zgaruvchi konstantalarning ro'yxati bo'lishi kerak.
- Topilgan xatolarni albatta ketma - ket tuzatish kerak.
- har doim chiqarilgan natijalarni yaxshilab ko'rib tahlil qilish zarur.
- Dastur variantlarni sanasi bilan alohida saqlanishni tahlil qilish kerak.

Sozlash vositalari.

Dasturni yozish vaqtida uning ichiga kiritiladigan sozlash vositalari eng samarali hisoblanadi. Bu holatda dasturchi tomonidan xatolarni aniqlash tez va aniq bo'ladi.

Dasturlashda quyidagi sozlash vositalari qo'llaniladi:

1. Xotira tarkibini tahlil qilish.
2. Algoritm bajarilishini nazorat qilish.
3. O'zgaruvchilardan foydalanishni nazorat qilish.
4. Indeksni tekshirish.
5. Kichik dasturlardan foydalanishni nazorat qilish.
6. O'zgaruvchilarni belgilarini tasvirlash.

Dasturni hozirgi holatini uni bajarilishiga qadar hisobga olishga yoki belgilashga yordam beradi. Yozuvdagi ma'lumotlar amaliyotda foydalaniladigan holatiga umuman mos kelmasligi sababli, sozlash moslamasiga intilish hosil bo'ladi va bu moslama ma'lumotni qulay holatga (o'qish uchun) keltirib boradi.

Algoritmni bajarish yo'llarini nazorat qilish.

Dasturni bajarilishini mantiqiy yo'lini hisobga oladi. Undan dasturchilar tomonidan berilgan jarayonni bajarilish ketma-ketligini va o'zgaruvchilarni xozirgi vaqtdagi belgisini bajarilishini va hisobga olishini tekshirishda foydalaniladi. Bunday kuzatishlarning uchta turi mavjud:

1. O'zgaruvchilarni belgilarini nazorat qilish.
2. Kichik dasturlarni chiqarishni hisobga olishni nazorat qilish.
3. Dasturdagi buyruqlarni etkazishni nazorat qilish.

Ikkinchi turdagi kuzatish asosan kichik dasturlardan foydalanilgan paytda sozlashda qo'llaniladi. Xar bir muomalada ezuvga kichik dasturni nomi chiqadi, undan chiqilganda-asosiy dasturga kaytilganligi xaqida ma'lumot chiqadi.

Bu turdagi tekshirish dasturchiga dasturiy xatolarni aniqlashdagi barcha kerakli ma'lumot bilan ta'minlab turadi. Uning kamchiligi mashina vaqtini ko'p sarf bo'lishidir, hamda berilayotgan ma'lumotlar bir necha ming qatordan iborat bo'lishidadir. Shuning uchun yuqoridagi kamchiliklardan voqif bo'lish uchun dasturni qisimlarga ajratib belgilangan joylarda o'chirib -yokib tekshiriladi.

O'zgaruvchilardan foydalanishda nazorat qilish moslamasi.

Nazorat moslamasi shunday tuziladiki, unda o'zgaruvchilar birdaniga emas, balki aniq ro'yxat bo'yicha ko'rsatilgandek tartib bilan yoziladi.

Indekslarni tekshirish. Massivlarni e'lon qilingan chegaralarni va ularni indekslni solishtirish orqali nomlangan massivlarni indeksatciyalanganligini to'g'riligini tekshirish.

O'zgaruvchilarni mazmunini takribiy chiqarish.

Display nomli sozlovchi buyruq orkali bajariladi. Chiqariladigan ma'lumotni to'liq chiqishini ta'minlash uchun o'zgaruvchini nomini va uni hozirgi holat mazmunini ko'rsatilgan buyruq orkali beriladi.

Interaktiv rejimida sozlash.

Juda ko'p holatlarda, dasturlar dasturchini mashina bilan o'zaro ta'siri ostida bajariladi, shuning uchun bu holatlarda qulayliklardan kengroq foydalanish ma'qulroq, ya'ni sozlash jarayonini samaradorligini oshirish uchun terminaldan kengroq foydalanish kerak. Paketli sozlash tizimida qo'llaniladigan usullarni ko'p qismni vaqti taksimlangan tizimda ham foydalanish mumkin, lekin interaktiv jarayon harakatida qo'shimcha faktorlarni va yangi sozlovchi usullarga yondashishga ham ahamiyat berish kerak .

Vaqti taksimlangan tizimlarda ko'pincha maxsus sozlovchi moslamalardan tashkil topgan bo'ladi. Shuning uchun quyida ko'rsatilgan qulayliklarni o'z ichiga olgan sozlovchi tizim terminaliga ega bo'lish kerak.

1. To'xtatish yoki uchirish kaliti ma'lum sharoitlarda dasturni bajarilishini to'xtatish, uni keyingi holatini qayd qilish va boshqaruvchi foydalanuvchiga uzatish.

2. Ruksat etilmagan jarayonlarni bajarish uchun harakat bo'layotgan holatlarda foydalanuvchiga boshqaruvchi uzatish uchun xatolar xaqidagi signal yordamida to'xtatib qo'yish.

3. Har qanday o'zgaruvchilarni mazmunini tartiblash yoki sozlashni xatolik signali yoki uzuq kalit orkali dastur uzilganda, unga o'zgartirish kiritish yoki holatini analiz qilish.

4. Qayta ishga tushirish, ya'ni uzilish sodir etilgan operatoridan yoki har qanday operatorlardan boshlab ishga tushirib dastur bajarilishini davom ettirish.

5. Qo'shimcha yoki alohida qatorlarni o'zgartirish, o'chirish uchun kerak bo'lgan modifitsirlangan dastur: bu xarakat sozlash jarayoni davom ettirilishi uchun zudlik bilan ishga tushirilishi kerak.

Nazorat savollari:

1. Qanday sozlash turlarni bilasiz?
2. Sozlash vositalarini vazifasi?
3. Interaktiv holatda sozlash - bu nima?

12-ma`ruza. Dastur anikligi. Ichki spetsifikatciya hujjatlari.

R E J A :

1. Testni loyihalash.
2. Testlash usullari.
3. Oq va qora qutilarni uslubiyoti.

Adabiyotlar

1. Van Tassel D. Stil, razrabotka, effektivnost, otladka i ispytanie programm. M. , Mir, 1991.
2. Linger R. , Teoriya i praktika strukturnogo programmirovaniya M. , Mir, 1985.
3. V. V. Lipaev. "Proektirovanie programmných sredstv", M. : "VSh", 1991.
4. Foks Dj. "Programmnoe obespechenie i ego razrabotka". Per. s angl. M. : Mir, 1985.
5. S. S. Gulomov. "Iqtisodiy informatika". Toshkent 1999 y.

Tayanch so'zlar: *Testni loyihalash. Testlash usullari. "Qora va Ok" quti usullari uslubologiyasi.*

Dasturni sozlash va testdan o'tkazish. Dastur loyihalashda murakkab jarayonlar sifatida sozlash bosqichi va testdan o'tkazish jarayoni hisoblanadi. Dasturni testdan o'tkazishning maqsadi, ya'ni dasturni sinashdan maqsad, undagi mavjud va engil xosil bo'lgan xatolarni aniqlashdir. Sozlashdan maqsad xatolarni hosil bo'lish sababini aniqlash va ularni bartaraf etish.

Dasturni sozlash uchun ishni testlash rejasini tuzishdan boshlanadi. Rejani tuzishda xatoni kelib chiqishi va uni xarakteriga yondashiladi. Asosiy sabablaridan biri matematik modellarni chuqur ishlab chiqilmaganligi va vazifani bajarish algoritmi to'liq qurilmaganligidir;

Dasturiy blankadagi boshlangich ma'lumotlar xaqidagi 1-tasavvur kiritish moslamasidagi klaviatura yordamida boshlangich ma'lumotni va dasturni tekshirishda loqaydlilik.

Xatolarni kelib chiqish sabablari turli xilligini hisobga olgan holda ularni 2 ta turga ajratiladi:

1. Sintaktik
2. Semantik

Sintaktik xatolar - dastur tilini tarkibini yozishdagi xatolar (sonlar, o'zgaruvchilar, funktsiyalar, operatorlar, belgi va kichik dasturlar).

Semantik xatolar - kattaliklarni ruxsat etilmagan kiymatlaridan foydalanishda va harakatni noto'g'ri tarkibi bilan bog'liq bo'lgan xatolar.

Asosiy dasturlash tizimlarida sintaktik xatolarni aniqlash avtomatlashtirilgan (fortran, paskalp, beysik, pl/1 va x. k).

Semantiq xatolarni aniqlash yaxshi yo'lga quyilmagan.

Testlash rejasiga odatda quyidagi tiplar kiradi:

1. Dasturni algoritm sxemasi bilan solishtirish
2. Display ekranida dasturni vizual nazorat qilish yoki dasturiy blankadagi originalni dastur raspechatkasini o'rganish, solishtirishni vizual urganish.
3. Dasturni mashina tiliga olib ko'rsatish. Bu bosqichda sintaktik xatolar aniqlanadi. Dastur listingidagi sintaktik xatolarni fortran, paskalp tilidagi ko'ramalar deagnostik xabar beradi.

Xatolar xaqidagi xabarlar har xil EVM larda va tizim versiyalarida (fortran, paskalp, beysik) shakliga qarab har xil bo'lishi mumkin.

Interpretatciyalovchi beysik - navbatdagi operatorni bajarishdaxalakit kilaetgan ham sintaktik, ham semantiq xatolarni birdaniga diagnostika qiladi.

4. Dasturni kompanovkalash va tashqi aloqalarni taxrirlash.

Tashqi aloqalarni taxrirlash jarayonida tashki aloqalar taxriri dasturi yoki vazifalarini tartiblovchi shunday sintaktik xatolarni aniqlaydiki, bu xatolarda kichik dasturlarda yozilgan prametrklar qiymati bir biriga mos kelmaydi, aslida yo'q bo'lgan standart dasturlarni chiqaradi. Masalan SIN o'rniga SIH va shunga o'xshash xatolar.

5. Dasturni bajarilishi.

Translyator va tashqi aloqalar taxriri tomonidan barcha xatolar bartaraf etilgandan so'ng keyingi - EHM da dasturni mashina tilida bajarish bosqichiga utiladi: dastur operativ xotiraga yuklatiladi va boshlangich ma'lumotlar kiritilgandan so'ng boshlanadi. Xato

borligi haqida belgi paydo bo'lishi sozlash o'tkazishna sabab bo'ladi: belgi yo'qligi dasturda xato yo'qligini bildiradi.

Testlash rejasi boshlangich ma'lumotlarning ruxsat etilgan kiymatlarini xulosasini to'g'riligini ham tekshirishni uz ichiga jamlaydi.

Testlash uslublari (Usullari).

Odatda dasturni testdan o'tkazishda bosqichlarga bo'lib urganiladi. Bunda har bir modulni tekshirishdan tortib, to butun tizimni yakuniy tekshirishlar kabi bosqichlarni oladi. Agar bunda biron bir ishonchli ketma - ketlikka endashilmasa, ishonchli ta'minlovchi dastur olish juda qiyindir. Testlash strategiyasi ikkita usuldan birortasiga asosan bajariladi: odatiy quyidan - yuqoriga qarab testlash, yoki zamonaviy yuqoridan - pastga qarab testlash.

Quyidan - yuqoriga qarab testlash.

Bu usul keng tarkalgan usul bo'lib, unda eng kuyu pog'onadagi

boshlangich yozilgan modullar tekshiriladi. So'ngra yuqori qatlamdagi elementlar dasturlanadi va testlanadi. Bu jarayon to yozilgan dastur butunlay yakunlanmaguncha davom etadi. Quyidan - yuqoriga qarab testlash usuli xozirgi vaqtda yuqoridan - pastga qarab testlovchi va dasturchilvr tomonidan kulanmayapti. Ularni fikricha bu usulda interfeys va algoritmdagi ko'pgina xatolar aniqlanmay kolib ketmokda. Bu esa dasturni kayta-kayta o'zgartirishdan so'ng buzishga olib keladi.

Ikkinchi kamchiligi esa: har xil pog'onadagi elementlarni tes-

tdan o'tkazishda yangidan yangi testlovchi moslamalarni, drayverlarni va testlovchi ma'lumotlarni talab qilmoqda. Bu esa o'z-o'zigadan dasturlashda katta hajmda mexnat talab qiladi.

Yuqoridan pastga qarab testlash.

Bu testlash usuli yuqoridan pastga qarab dasturlashni, yuqoridan pastga qarab kodlashni qo'shimcha bosqichi hisoblanadi. Bu usulda oldin asosiy dastur eziladi va so'ngra past pog'onadagi loyihalanmagan elementlar urin bosuvchi dasturlar bilan almashtiriladi. Bunday skeletli dastur chaqiriluvchi dastur va har qanday malumotlar yo'qligida ham uz ishini davom ettiradi. Bu tekshirish natijasida bazi xollarda bemani bo'lgan xatolar ham aniqlanadi. Keyingi qadam modul kushilishidan iborat bo'lib, unda bu

modullar kiruvchi modullarni ko'paytiruvchi bo'lishi ham mumkin, -bu esa kiritish moduli, bazi bir yordamchi modul (oxirgini dasturlash tugash dakikasiga qadar) bo'lishi mumkin. Bu tekshirishdan so'ng sinash oddiy bir sodda kiruvchi malumotlar bilan o'tkazish mumkin.

Favqulotda xollarda tekshirish.

Dasturni testlashning oxirgi bosqichi o'zgartirishlarsiz ruxsat

etilan chegarasidan tashqarida joylashgan kiymatlaridan iborat bo'lgan malumotlardan foydalangan xolda bajariladi. Dasturlarni favqulotda va qutilmagan xollarda tekshirishda bir necha maslaxatlar beriladi. Bu maslaxatlar tekshirish vaqtini tayerlashga yordam beradi.

Avvalambor taxrirlangan malumotlarni to'g'riligini tekshirish kerak bo'lsa, oldin ruxsat etilgan oblastni ikkala chegarasida etgan kiymatni oling, chunki shu kiymatlar dasturni ishlashida kiynchiliklar tugdirish extimoli yuqoriroq. Bir xilgi holatlarda qiyinchiliklarni bir nechta xatolarni jamlagan malumotlar tugdirishi mumkin. Xatoning yana bir turi - bu ishlov berilaetgan malumotning birligi yoki keyingi elementidagi xatolardir. Shuning uchun dastur o'z ishini odatiy xolga o'xshab tugatmasligi muxumdur. Buning uchun teslashda yagona bir faylni formalansa etadi. Chunki bu fayl noto'g'ri elementdan tashkil topadi. Agar bu ishlash testlash bosqichida bajarilgan bo'lsa, dasturni eksplutatciya qilishda qiyinchiliklar yuzaga keladi.

" Qora " va "Oq" quti usullari uslubologiyasi.

Testdan o'tkazish dasturlarini normal holatda ishlashini garantlovchi jarayondir. Qora quti usuli testdan o'tkazish jarayonida ishlatiladigan usullardan biridir unda dastur kibernetik ob`ekt sifatida qabul kilinadi va uning kirish kiymatlari va chiqarish kiymatlari bor deb xolos.

Quti ichida qanday jarayon ketaetganligini hisobga olinmaydi. Quti kirishiga kiymatlar beriladi va chiqish qismida ma`lum natija olishni kuzda tutiladi.

Ikkinchi usullaridan biri bu ok quti usulidir. Bunda dasturni kiritish va chiqarish bo'lgan ob`ekt sifatida qabul kilinadi va kirish kiymatlari kanyday qilib chiqarish natijalari aylanish mexanizm (dastur matni) ham ma`lum bo'ladi.

Nazorat savollari:

1. Dasturni tesdan o'tkazishning maqsadi
2. Qanday testlash uslublarini (usullarini) bilasiz va ularning vazifalariga xarakteristika bering?
3. "Qora" va "Ok" quti usullari metologiyasiga tushushcha bering?
4. Dasturni tesdan o'tkazishning maqsadi
5. Qanday testlash uslublarini (usullarini) bilasiz va ularning vazifalariga xarakteristika bering?
6. "Qora" va "Ok" quti usullari metologiyasiga tushushcha bering?

13-ma`ruza. Xatoliklar. Tizimli xatolik, masalani tavsiflashdagi xatolik. Algoritm tanlashdagi xatolik.

R E J A :

- 1. Testlash vositalari.**
- 2. Test o'tkazish bosqichlari.**
- 3. Testlarni rasmiylashtirish.**

Adabiyotlar

1. Van Tassel D. Stil, razrabotka, effektivnost, otladka i ispytanie programm. M. , Mir, 1991.
2. Linger R. , Teoriya i praktika strukturnogo programmirovaniya M. , Mir, 1985.
3. V. V. Lipaev. "Proektirovanie programmných sredstv", M. : "VSh", 1991.
4. Foks Dj. "Programmnoe obespechenie i ego razrabotka". Per. s angl. M. : Mir, 1985.
5. S. S. Gulomov. "Iqtisodiy informatika". Toshkent 1999 y.

Tayanch so'zlar: *Testlash vositalari. Test o'tkazish bosqichlari. Testlarni rasmiylashtirish, testni loyihalash, . testlash usullari. "Qora va Ok" quti usullari uslubologiyasi.*

Dasturni testlash jarayonida muxum shartlardan biri shuki dasturlashga qancha vaqt sarf bo'lgan bo'lsa testlashga ham shuncha vaqt ajratish kerak. Chunki dasturlashda rejalashtirish ishlariga intilish (tendentciya) mavjud bo'lib, shuning uchun barcha loyxalarni muddatlarini buzgan xolda bajarilishini tushinish mumkin. Bu esa ishlab chiqishga, kodlashga, sozlashga va dasturni testlashga aloxida vaqtlar ajratishni va biron bir jarayonni bajarish vaqtini biror haftaga surilsa, kolgan bosqichlarni bajarish mudatlari ham surilishini kuz oliga keltirishi kerak.

Test o'tkazish bosqichlari:

1. Normal sharoitda tekshirish.
2. Ekstremal sharoitda tekshirish.
3. Favqulot holatlarda tekshirish.

Normal sharoitda tekshirish.

Dasturni ishlash sharoitidan chiqan xolda olingan kiymatlar asosida tekshirish.
Dastur normal sharoitda to'g'ri qiymatlar chiqarishini ko'rsatish kerak.

Ekstremal sharoitda tekshirish.

Bu tekshirishda dasturdagi o'zgaruvchilarning chegaraviy kiy matlari asosida test utkaziladi. Masalan: Eng kichik kiymatdan maksimal kiymatga ega (sonli o'zgaruvchi uchun) hamma belgilarni chop qilishi (belgilik o'zgaruvchilari uchun). Bu tekshirishda o'zgaruvchilarni nolg' kiymati ham bo'ladi.

Favqulot holatda tekshirish.

Bu tekshirish o'zgaruvchining kiymatlar soxasidan tashqarida etgan kiymatlar asosida tekshirish qiladi.

Rejalashtirish.

Testdan o'tkazish jarayoni vaqtida dastur yozish bilan bir bo'lganligi hisobga olib testdan o'tkazishni ham rejalashtiriladi.

Testlash rejasiga odatda quyidagi tiplar kiradi.

1. Dasturni algoritm sxemasi bilan solishtirish.
2. Displey ekranida dasturni vizualg' nazorat qilish yoki dasturiy blankadagi originalni dastur raspechatkasini urganish, solishtirishni vizualg' urganish.

3. Dasturni mashina tiliga olib ko'rsatish. Bu bosqichda sintaktik xatolar aniqlanadi. Dastur listingidagi sintaktik xatolarni fortran, paskalp tilidagi ko'ramalar deagnostik xabar beradi.

Xatolar xaqidagi xabarlar har xil EVM larda va tizim versiyalarida (fortran, paskalp, beysik) shakliga qarab har xil bo'lishi mumkin.

Interpretatciyalovchi beysik - navbatdagi operatorni bajarishda xalakit kilaetgan ham sintaktik, ham semantiq xatolarni birdaniga diagnostika qiladi.

4. Dasturni kompanovkalash va tashki aloqalarni taxrirlash.

Tashki aloqalarni taxrirlash jarayonida tashki aloqalar taxriri dasturi yoki vazifalarini tartiblovchi shunday sintaktik xatolarni aniqlaydiki, bu xatolarda kichik dasturlarda yozilgan prametrklar kiymati bir biriga mos kelmaydi, aslida yo'q bo'lgan standart dasturlarni chiqaradi. Masalan SIN o'rniga SIH va shunga o'xshash xatolar.

5. Dasturni bajarilishi.

Translyator va tashki aloqalar taxriri tomonidan barcha xatolar bartaraf etilgandan so'ng keyingi - EHM da dasturni mashina tilida bajarish bosqichiga utiladi: dastur operativ xotiraga yuklatiladi va boshlangich ma'lumotlar kiritilgandan so'ng boshlanadi. Xato borligi xaqida belgi paydo bo'lishi sozlash o'tkazishna sabab bo'ladi: belgi yo'qligi dasturda xato yo'qligini bildiradi.

Testlash rejasi boshlangich ma'lumotlarning ruxsat etilgan kiymatlarini xulosasini to'g'riligini ham tekshirishni uz ichiga jamlaydi. Dasturni testlashning oxirgi bosqichi o'zgartirishlarsiz ruxsat etilan chegarasidan tashqarida joylashgan kiymatlaridan iborat bo'lgan malumotlardan foydalangan xolda bajariladi. Dasturlarni favqulotda va qutilmagan xollarda tekshirishda bir necha maslaxatlar beriladi. Bu maslaxatlar tekshirish vaqtini tayerlashga yordam beradi.

Avvalambor taxrirlangan malumotlarni to'g'riligini tekshirish kerak bo'lsa, oldin ruxsat etilgan oblastni ikkala chegarasida etgan kiymatni oling, chunki shu kiymatlar dasturni ishlashida kiynchiliklar tugdirish extimoli yuqoriroq. Bir xilgi holatlarda qiyinchiliklarni bir nechta xatolarni jamlagan malumotlar tugdirishi mumkin. Xatoning yana bir turi - bu ishlov berilaetgan malumotning birligi yoki keyingi elementidagi xatolardir. Shuning uchun dastur uz ishini odatiy xolga o'xshab tugatmasligi muxumdur. Buning uchun teslashda yagona bir faylni formalansa etadi. Chunki bu fayl noto'g'ri elementdan tashkil topadi. Agar bu ishlash testlash bosqichida bajarilgan bo'lsa, dasturni eksplutatciya qilishda qiyinchiliklar yuzaga keladi.

Dasturni to'g'riligi.

Xamma dasturlar mantiqan olib karaganda ayrim bir ma'lumotlar ta'sir kursatuvchi xududlarni aniq ko'rsatishi, ya'ni dastur ish bajarish kobiliyatiga ega bo'lishi, malumotlarni kursatilgan chegaralarda turganligini aniqlash uchui operatorlarni kiritish imkoniga ega bo'lishi kerak.

Dasturni foydalanishga berishdan oldin uni to'g'riligigi ishonch xosil qilish kerak. Noto'g'rilikni aniqlashni ikkita usuli bor:

1. Dasturni konstruktciyasi sintaktik xato.
2. Dastur noto'g'ri natijalar ko'rsatmoqda.

Odatda dasturni testdan o'tkazishda bosqichlarga bo'lib urganiladi. Bunda har bir modulni tekshirishdan tortib, to butun tizimni yakuniy tekshirishlar kabi bosqichlarni oladi. Agar bunda biron bir ishonchli ketma - ketlikka endashilmasa, ishonchli

ta'minlovchi dastur olish juda qiyindir. Testlash strategiyasi ikkita usuldan birortasiga asosan bajariladi: odatiy quyidan - yuqoriga qarab testlash, yoki zamonaviy yuqoridan - pastga qarab testlash. Quyidan - yuqoriga qarab testlash.

Bu usul keng tarkalgan usul bo'lib, unda eng kuyu pog'onadagi

boshlangich yozilgan modullar tekshiriladi. So'ngra yuqori qatlamdagi elementlar dasturlanadi va testlanadi. Bu jarayon to yozilgan dastur butunlay yakunlanmaguncha davom etadi. Quyidan - yuqoriga qarab testlash usuli xozirgi vaqtda yuqoridan - pastga qarab testlovchi va dasturchilvr tomonidan kulanmayapti. Ularni fikricha bu usulda interfeys va algoritmdagi ko'pgina xatolar aniqlanmay kolib ketmokda. Bu esa dasturni kayta-kayta o'zgartirishdan so'ng buzishga olib keladi.

Dasturchiga maslaxatlar.

- Nazorat qilish usullarining eng oz mikdordagisidan foydalarning.
- Dasturni sinashda xech qanday kamchiliklarga yo'l kuymagan holda, aqliy ravishda o'tkazing.
- Testlashni iloji boricha ertaroq boshlang.
- Tekshirishda kul mexnatidan foydalaning.
- Tizimni kurilish printciplarini to'g'riligini tekshirishda oddiy ko'rinishda qo'llashga xarakat qiling.
- Testlarni yuqoridan pastga qarab usulidan foydalanishga xarakat qiling.
- Navbatdagi har qanday testni yangi ma'lumotlar sinfidan foydalaning.
- Dasturni oddiy, eksteremal va favkulodda holatlarda tekshiring.
- Sinashga ketadigan vaqtni oldindan rejalashtiring.
- Dasturga kiritilgan o'zgartirishlardan so'ng uni yana testdan o'tkazing.

Nazorat savollari:

1. Testlar vositalarini vazifasi
2. Test o'tkazish nechta bosqichdan iborat?
3. Testlarni rasmiylashtirish deganda nimani tushunasiz?
4. Dasturni tesdan o'tkazishning maqsadi
5. Qanday testlash uslublarini (usullarini) bilasiz va ularning vazifalariga xarakteristika bering?

14-ma`ruza. Dasturning asosiy xarakteristikolari. Ishonchlik, samaradorlik, inson faktorini hisobi, tushunishlik.

R E J A :

- 1. Ishonchlik.**
- 2. Samaradorlik.**
- 3. Inson faktorini hisobga olish.**

Adabiyotlar

1. Van Tassel D. Stil, razrabotka, effektivnost, otladka i ispytanie programm. M. , Mir, 1991.
2. Linger R. , Teoriya i praktika strukturnogo programmirovaniya M. , Mir, 1985.
3. V. V. Lipaev. "Proektirovanie programmných sredstv", M. : "VSh", 1991.
4. Foks Dj. "Programmnoe obespechenie i ego razrabotka". Per. s angl. M. : Mir, 1985.
5. S. S. Gulomov. "Iqtisodiy informatika". Toshkent 1999 y.

Tayanch so'zlar: *Ishonchlik, samaradorlik, inson faktorini hisobi, tushunishlik.*

Ishonchlik.

Dasturning har xil muhitda va har xil qiymatlar bilan ishlash ishonchlik deb ataladi.

Samaradorlik.

Programani oz hajmi bilan ishlash samaradorlik deb ataladi.

Lekin samaradorlik 2-bosqich bo'lib, birinchi bosqichda dasturni to'g'ri ishlashini tashkil qilish kerak.

Ma`lumotlarni tafsiflash.

Yaxshi tafsiflangan ma`lumotlar dasturni ancha kiskartiradi, demak ma`lumotlar massivini tanlashda zarur bo'lganicha qo'llash kerak. Boshqacha usulda tayanch va kursatkichlardan foydalanish kerak.

Ma'lumotlar xaqida kurilaetgan vazifaga mos xolda taosurotga ega bo'lish kerak. Hisobda har bir ma'lumot tarkibini hamma tilga o'zgartirish mumkin. Lekin siz muljallagan ma'lumot tarkibini mujassamlagan dasturlash tilidan foydalanishingiz maoqulroq.

Dasturiy foydalanuvchilar extiyojlarini qondirish, keng tarqatish va sotish uchun mo'ljallangan.

Xozirgi paytda dasturiy maxsulotlarni ochik (legal) tarqatishning boshqa variantlari ham mavjud, ular yalpi (global) va mintaqaviy kommunikatsiyalardan foydalanish bilan yuzaga keladi.

Freeware-erkin tarkatiladigan foydalanuvchining uzi kul-lab-kuvvatlaydigan bepul dasturlar, u bularga zaruro'zgartirishlar kiritishga xakli. Shareware-notijorat (shartli-tulovsiz) dasturlar, ulardan odatdatulovsiz foydalanish mumkin. Bunday maxsulotlardan doimiy foydalanilganda muayyan summa badal tulanadi.

Bir qator ishlab chiqaruvchilar OEM-dasturlar (Original Equipment Manufacturer), ya'ni kompyuterlarga o'rnatilgan yoki hisoblash texnikasi bilan birgalikda keltirilgan maxsus dasturlardan foydalaniladi.

Dasturiy maxsulot foydalanishga tegishli ravishda tayerlanishi zarur texnik hujjatlariga ega bo'lishi, shuningdek davlat ro'yxati kodi mavjud bo'lishi lozim. Faqat shunday sharoitlardagina yaratilgan dasturiy majmua dasturiy maxsulot deb nomlanishi mumkin.

Dasturiy maxsulot-sanoat maxsulotining istalgan turi kabi realizatsiyaga tayerlangan ommaviy extiyojli muayyan muammo (vazifa) ni xal etish uchun o'zaro bog'langan dasturlar majmuasidir.

Dasturiy maxsulotlar quyidagtcha yaratilishi mumkin:

buyuormaga ko'ra individual ishlanma;

foydalanuvchilar orasida ommaviy tarqatish uchun ishlanma.

Individual ishlanmada firma-ishlab chiqaruvchi muayyan buyurtmachi uchun ma'lumotlarni kayta ishlash o'ziga xosligini hisobga oluvchi dasturiy maxsulotni yaratadi.

Ommaviy tarqatish uchunishlanmani yaratishda firma-ishlab chiqariuvchi, bir tomondan, ma'lumotlarni kayta-ishlashni bajarilaetgan funktciyalar universalligi, boshqa tomondan, muayyan bir qo'llash sharoitida dasturiy maxsulotning moslanishi va

sozlanishini ta'minlash lozim. Dasturiy maxsulotlarning ajralib turuvchi xususiyati uning tizimlilik-jamlikda qo'llanilgan xolda amalga oshiriladigan kayta ishlash vazifalarining funktsional tlakonligi va tugalligi bo'lmogi lozim. Dasturiy maxsulot dasturlashtirishning zamonaviy asbobsozlik vositalari qo'llangan xolda loyiha ishlarini bajarish sanoat texnologiyasi asosida ishlab chiqiladi. Uning o'ziga xaosligi axborot va asbobsozlik vositalaridan foydalanishni kayta ishlash xususiyatiga bog'liq xolda algoritm va dasturlarni ishlab chiqish jarayonining noobligidadir. Dasturiy maxsulotlarni yaratishga ko'plab mexnat, moddiy, moliyaviy zaxiralar talab etiladi; yuqori malakali mutaxassislar zarur.

Dasturiy maxsulotni tayerlash (kuzatish) -dasturiy maxsulot ishga laekatliligini qo'llab-kuvvatlash, unga yangi versiyalar, o'zgartirishlar kiritish, topilgan xatolarni tugralash va xokazolarni uz ichiga oladi.

Dasturiy maxsulotlar an'anaviy dasturiy maxsulotlardan farkli ravishda dasturlarni yaratishda beriladigan sifat xususiyatlarining kat'iy belgilangan turkumiga ega emas yoki bu xususiyatlarni oldindan aniq ko'rsatish yoki baholash mumkin emas, chunki dasturiy vosita ta'minlaydigan bir xil kayta ishlash vazifalari turli ichki ishlanmalarga ega bo'lishi mumkin. Xatto dasturiy maxsulotlarni ishlab chiqishga sarflanadigan vaqt va hujjatlarni ham oldindan katta aniqlikda belgilash mumkin emas.

Dasturlarni asosiy tavsiflari quyidagilar;

- * algoritmik murakkablik(axborotni qayta ishlash algoritm mantiqi);
- * kayta ishlashning amalga oshirilgan vazifalari ishlanmalarning tarkibi va chuqurligi;
- * kayta ishlash vazifalarining to'laqlonligi va tizimlilik;
- * dasturlar fayllari hajmi;
- * dasturiy vosita tomonidan kayta ishlashning operatcion tizimi va texnik vositalarga talablar;
- * diskli xotira hajmi;
- * dasturni tushurish uchun operativ xotira o'lchami;
- * protcessorlar turlari;
- * operatcion tizimlar versiyalari;
- * hisoblash tarmoklarining mavjudligi va boshqadar.

Dasturiy maxsulotlarning sifat kursatkichlari xilma-xil, ular quyidagi jixatlarda aks etadi:

- * dasturiy maxsulotni qanchalik yaxshi (oddiy, ishonchli, samara li)foydalanish mumkinligi;
- * dasturiy maxsulotdan qanchalik engil foydalanish mumkinligi;
- * dasturiy maxsulotni qullashda sharoit o'zgarganda undan foydalanish mumkinligi yoki yo'qligi va boshqadar.

Xarakatchanlik-dasturiy maxsulotlarda ma'lumotlarni kayta ishlash tizimi texnik majmuasi, operatsion muxit, ma'lumotlarni kayta ishlashning tarmok texnologiyasi predmetli soxa o'ziga xosligi va xokazolardan mustakillikni anglatadi. Xarakatchan (ko'p platformali)dasturiy maxsulot hisoblash tarmogi sharoitida foydalanishga hech qanday cheklanishlarsiz kompyuterlar va operatsion tiximlarningturli modellariga urnatilishi mumkin. Bunday dasturiy maxsulotlarni kayta ishlash funktsiyalari biron-bir o'zgarishsiz ommaviy foydalanish uchun makbuldir.

Ishonchlilik-dasturiy maxsulot ishida uzliksizlik va barkarorlik, kayta ishlashni bajarishning aniqligi, ish jarayonidagi xatolarni oldindan bilish bilan belgilanadi.

Samaradorlik-dasturiy maxsulot faoliyatida uning ham bevosita vazifasi-foydalanuvchi talabi, ham foydalanish uchun zarur bo'lgan hisoblash zaxiralari xarajatlari nukta'i nazaridan baholanadi.

Inson omilini hisobga olish oxirgi foydalanuvchi uchun dustona interfeysni ta'minlash, dasturiy vosita tarkibida kontekstli bog'liq xolda aytib beruvchi yoki ukituvchi tizim, dasturiy vositaga kiritilgan funktsional imkoniyatlarni uzlashtirish va ulardan foydalanish uchun yaxshi hujjatlar mavjudligini anglatadi.

Nazorat savollari:

1. Dastur qanday asosiy xarakteristikalarga ega?
2. "Ishonchlik, samaradorlik, inson faktorini xislobi, tushunishlik"-tushunchalarni vazifasini ko'rsating.
3. Dasturni tesdan o'tkazishning maqsadi
4. Qanday testlash uslublarini (usullarini) bilasiz va ularning vazifalariga xarakteristika bering?

15-ma`ruza. Programmaning asosiy xarakteristikalari. Dasturni loyihalashda texnologik jarayoni.

R E J A :

- 1. Dastur tuzish usullari.**
- 2. Dastur tuzishning texnologik jarayoni.**

Adabiyotlar

1. Van Tassel D. Stil, razrabotka, effektivnost, otladka i ispytanie programm. M. , Mir, 1991.
2. Linger R. , Teoriya i praktika strukturnogo programmirovaniya M. , Mir, 1985.
3. V. V. Lipaev. "Proektirovanie programmných sredstv", M. : "VSh", 1991.
4. Foks Dj. "Programmnoe obespechenie i ego razrabotka". Per. s angl. M. : Mir, 1985.
5. S. S. Gulomov. "Iqtisodiy informatika". Toshkent 1999 y.

Tayanch so'zlar: *Dasturni loyihalashda texnologik jarayoni, prosedura, fakt.*

Dastur tuzish usullari va vositalari.

Dastur tuzishninig eng asosiy usullaridan bir bu strukturaliy dastur tuzishdir. Bu usulda dastur tuzish uchun uchta qism mavjud:

1. Yuqoridan pastga dasturm tuzish.
2. Modul dasturlashtirish.
3. Strukturali kodlash.

Yuqoridan pastga dastur tuzishda dasturning yuqori qismdan boshlanadi. Dasturning asosiy qismi tuzilib, quyi qismidagi modullar esa vaqtinchalik fakt nomlari bilan atalgan proseduralar bilan almashtiriladi. Dasturni asosiy moduli tuzilib, testidan utkazilgandan so'ng ketma-ket vaqtincha yozilga modullarni yozish bilan dastur tuzish davom ettiriladi.

Modul proqrammalashtirishda dasturni mantiqiy kisimlariga bo'linadi. Bu modullar dasturda protceduralar va funktsiyalar orkali amalga oshiriladi.

Strukturali kodlash deganda xarbir modulni gorizantal va vertikal qatorlarda to'g'ri nomlanishiga aytiladi. Bu usul yordamida modullardan tuzilgan dasturlar ishlaydigan testidan o'tkazishi qulay mdifikattciya qilish uchun qulay dasturlar yaratish mumkun.

Dasturchi vaz'ifani tahlil qilib kerakli bo'lgan algoritumni tanlaydi. Tanlangan algoritm to'liq tahlil kilinadi va uning blok-tcxemasi chiziladi.

- Dasturlashtirish bosqichi. Dastur yaratish tili tanlanadi.

Dastur qabul qilingan algoritimda tuziladi.

- Dasturni tuzatish bosqichi.

- Dasturni testidan o'tkazish bosqichi.

EXMdan foydalanib masalani echish- yaratilgan algoritimga asoslangan xolda dastlabki ma'lumotlar ustida avtomatik tarzida amallar bajarib izlardan natija ko'rinishiga keltirish demakdir.

Informatika fani uch tarkibiy qismdan iborat:

EXMdan foydalanib "ilmiy- texnik masalani echish" tushunchasi keng maonodagi so'z bo'lib, quyidagi bosqichlarga bo'linadi. Maqsadimiz bosqichlarni qaysi birlarini mutaxassis EXMdan foydalanmasdan va qaysi bosqichlani EXMdan foydalanib bojarishni aniqlash, hamda bosqichlarni tula urganib chiqishdan iborat.

Ilmiy texnik masallarni EXMdan foydalanib echish bosqichlari:

1. Masalaning qo'yilishi va maqsadining aniqlanishi;
2. Masalani matematik ifodalash;
3. Masalani echish uslubini ishlab chiqish, sonliy usullarni tanlash;
4. Masalani echish algoritimini ishlab chiqish;
5. Ma'lumotlarni taerlash va tarkibini aniqlash (tanlash);
6. Dasturlash;
7. Dastur matnini va ma'lumotlarni axborot tashuvchiga o'tkazish;
8. Dastur xatolarini tuzatish;
9. Dasturni avtomatik tarzda EXMda bojarilishi;
10. Olingan natijalarni izohlash, tahlil qilish va dasturdan foydalanish uchun ko'rsatma yozish;

"Informatika" kursida 1-4 bosqichlar qisqa maonoda, xususiyy xatolar, ko'p uchraydigan murakkab bo'lmagan malumotlar uchun tushutiriladi. Bu bosqichlar tom

maonoda to'raligicha mutaxassislikni egallash davomida maxsus kurslar vositasida o'rgatiladi.

8 va 9-bosqichlarni bajarishda mutaxassis (EXMdan foydalanuvchi) EXMdan foydalanadi.

7-bosqichda EXMdan foydalanish ham, foydalanmaslik ham mumkin ITMni EXMda echish bosqichlarini aloxida ko'rib chiqamiz.

1-bosqich. Masalaning qo'yilishi va maqsadni aniqlanishi. Xalk xujaligining muayan soxasi (texnika, iktisod, lingivistika, ta'lim va x. k) buycha ishlayotgan malakaliy va etakchi mutaxassis tomonidan bajariladigan ish.

Masalani maqsadni amalga oshirish uchun kerakli ma'lumotlartarkibi (strukturasi), tuzilishi, ifodalanishi aniqlangan bo'lib, ular orasidagi boglanishlar aniq ifodalangan bo'lsa masala quyilgan deb aytiladi.

2-Bosqich. Masalani matematik ifodalash.

Bu bosqichda masalani echish uchun kerakli va etarli bo'lgan dastlabki ma'lumotlarni tarkibi, tavsifi, turi, tuzilishi hisobga olingan xolda matematik terminlardan ifodalanadi, hamda masalani echishning matematik terminlarda ifodalanadi, hamda masalani echimining matematik terminlarda ifodalanadi, hamda masalani echishning matematik modeli yaratiladi. Buning uchun har xil matematik aparat ishlatilishi mumkun. Masalani iktisod soxasidagi mutaxassislar - chiziklidasturlash, dinamik dasturlash, stoxastik dasturlash, bashoorat qilish bilan bog'liq masalalarni echish mamtematik aparatini bildirish kerak; texnik soxasidagi mutaxassisliklar oddiy differentcial tenglamalar va ularning tizimlari, mexanikaning chetki masalalarni, gaz dinamikasiga oid maslalarni, integral ko'rinishidagi masalani ifodalash va echish uchun ishlatiladigan matematik aparatni to'liq tushunib etgan bo'lishi kerak.

3- bosqich Masalani echish usulini ishlab chiqish, sonli usulni tanlash.

4- bosqich Masalani echish algoritmin yaratish.

5- bosqich Ma'lumotlarni tayerlash va tarkibini aniqlash.

6- bosqich dasturlash.

7- bosqich dastur matnini va ma'lumotlarni axborot tashuvchiga o'tkazish.

8- bosqich dasturni xatosini tuzatish.

9- bosqich dasturni avtomatik tarzda exmda

BAJARILISHI.

10- bosqich olingan ma`lumotlarni izohlash, tahlil qilish va dasturdan foydalanish uchun yuriknoma yozish.

Nazorat savollari:

1. Dasturni tayirlash bosqichlarini ko'rsating.
2. Dasturlashning maksadini ko'rsating?
3. Universal dastur vazifasini ko'rsating
4. Dasturni tesdan o'tkazishning maqsadi
5. Qanday testlash uslublarini (usullarini) bilasiz va ularning vazifalariga xarakteristika bering?

16-ma`ruza. Dasturiy ta`minot maxsulot sifatida. Dasturiy maxsulotni loyخالash usullari va komponentlari.

R E J A :

1. Maxsulot kontseptciyasi.

2. Dasturiy maxsulotni loyخالash usullari.

Adabiyotlar

1. Van Tassel D. Stil, razrabotka, effektivnost, otladka i ispytanie programm. M. , Mir, 1991.
2. Linger R. , Teoriya i praktika strukturnogo programmirovaniya M. , Mir, 1985.
3. V. V. Lipaev. "Proektirovanie programmných sredstv", M. : "VSh", 1991.
4. Foks Dj. "Programmnoe obespechenie i ego razrabotka". Per. s angl. M. : Mir, 1985.
5. S. S. Gulomov. "Iqtisodiy informatika". Toshkent 1999 y.

Tayanch so`zlar: *Dasturiy maxsulotni loyخالash usullari va komponentlari.*

Maxsulot kontseptciyasi.

Dastur maxsulot sifatida xukuk tomonidan ximoyalangan tovardir. Dasturni xukuk tomonidan Uzbekiston Respublikasining tegishli konunlari ximoya qiladi.

Tuzilgan dastur maxsulot sifatini olish uchun va uning muallifi himoyalanishi uchun Respublikada tuzilgan "Algoritm va dasturlar fondiga" murojat qilish va undan tuzilgan dasturni qayd etish zarur.

Dastur mahsulotini tayorlash vositalari va komponentlari.

Dastur maxsulotini tayorlashda quyidagilar zarur:

- Taqdimnoma
- Diskda yozilgan bajariluvchi fayl (exe va com fayllarida)
- Dasturni matni
- Tushuntirish xati
- Testdan o`tkazish kiymatlari
- Test natijalari

- Modifikatsiya qilish yo'llari

Dasturni va dasturiy komplekslarni loyihalash kiska vaqt ichida loyihalovchilar yordamisiz foydalaniladigan dasturiy maxsulot yaratishni taminlashi lozim. Dasturiy maxsulot sifatida dasturning teksti, mashinada ifodalovchi sifati ko'rinishi (magnit disk) va dasturni hujjatlari sifatida kuriladi. Dasturiy maxsulotlar dastur va algoritm fondida hisoba olinadi. Ularning vazifasiga foydalanuvchi uchun dastur va uning hujjatlarini nusxalarini chiqarib berish kerak.

Dasturiy maxsulot sifatiga quyidagilar talab kilinadi: Funktsionalligi, ishonchligi, foydalanishga qulayligi (kerakli hujjatlarni yaratishda va dasturiy kompleksni loyihalashni to'g'ri bajarish hisobiga) Murakkab dastur va dasturiy komplekslar uchun dastur loyihalash algoritmi va ma'lumotlar tarkibini loyihalash bilan birgalikda bajariladi.

Loyihalashda vazmfani bir nechta kichik vazifalarga bo'lib, ularni echish uchun dasturiy modul xosil kilinadi. Bu holatda dastur ko'p moduli struktura sifatida loyihalanadi. Ko'p modulli dasturlarni ikkita loyihalash usuli mavjud: kutariluvchi va pasayuvchi. Kutariluvchi loyihalash (pastdan yuqoriga qarab loyihalash). Bunda bir nechta yirik modullarga ajratib ularni ayrim funktsiyalari umumiy dasturda amalga oshiriladi.

Modullarni ajratiladigan vaqtda foydalaniladigan funktsiyalarni tushinish uchun osonligicha, ma'lumotlar tarkibini oddiylikicha, berilgan funktsiyalarni amalga oshirish uchun modul va dasturlarni tayer holatda borligicha, yangi maksdlarda tayer dasturni o'zgartirishga imkoniyat borligicha va kelajakdagi modulning o'lchami va ahamiyatiga endashadi. Kutariluvchi loyihalashda har bir modul avtonom ravishda dasturlanadi, testdan utkaziladi va sozlanadi. Shundan keyin aloxida bo'lgan modullar podtizimga boshqaruvchi modullar yordamida birlashtiriladi. Bu holatda modullar chaqirish ketma ketligi, kiritish, chiqarish, natija va ma'lumotlar nazorati yordamida birlashtiriladi. Shunga asosan uz navbatida podtizimlar murakkab tizim va umumiy dasturiy kompleksga birlashtiriladi. Bunda modullararo aloqalar to'g'riligiga kompleks sozlashdan utkaziladi.

Bunday endashishlar uncha murakkab bo'lmagan dasturlarni loyihalashda foydalanish maslahat beriladi. Agar kichik dasturlarning o'lchamlari katta bo'lmasa, unda kichik dasturlarni ajratib ularni loyihalashni boshlagan maqulokdir.

Kutariluvchi loyihalashning asosiy nuksonlari sifatida modullarni yagona tizimga birlashtirishda, loyihalashning boshida yo'l quyilgan xatolarni aniqlash va ularni bartaraf etishni kurish mumkin. Undan tashqari aloxida bo'lgan modullarni butun tizimni tarkibini

tasavur etmagan xolda yaratish mumkin. Bu esa birlashtirishni murakkablashtiradi. Juda murakkab bo'lgan dasturlarni loyihalashda pasayivchi loyihalash maslaxat beriladi. Bu usulda echilgan masalani umumlashtirish pogonalariga buysunishiga asoslanadi. Umumlashtirish pogonalarini buysunish sxemasi dasturini fikrini oldin nima qilish kerak va undan so'ng qanday qilish kerakligiga to'g'rilaydi. Asosiy dastur yuqori pog'onadagi dastur, chaqiriklarni boshqaruvchi ancha quyi pog'onaga eziladi. Bu quyi pog'onadagi modullar uz navbvtida yana ham quyi pog'onada bo'lgan modullar chaqiriklarni boshqaradi. Bunday loyihalash maydoni murakkab va nozik bo'lgan dasturlarni uncha katta bo'lmagan oddiy modullardan yaratishga yordam beradi: vazifa o'lchami umumlashtiruvchi modullar sonida tasvirlanadi.

Dasturni yuqori sifatiga birinchi urinda loyihalash vaqtida algoritm sxemasini chukur ishlab chiqish hisobiga erishiladi. Bu avvalambor dasturni bexatoligi, dasturchini xatolar yo'qligiga ishonch va foydalanuvchini dasturni to'g'rililigiga ishonchidir. Dasturni bu xatoligiga ishonchlilik, uni tushunarligi, odiyligi, oson o'qilishi, mualiflar va foydalanuvchilar tomonidan oson interpretatsiyalashdir. Chunki xatolar uni yaratishda va foydalanishda ham paydo bo'lishi mumkin.

Loyihalash vaqtida xato kilmaslikni iloji yo'q, shuning uchun ularni oldini olish uchun quyidagi maslaxatlar beramiz. Bu maslaxatlar yordamida siz tezda xatolarni va ularni sabablarini aniqlashingiz, hamda oldini olishingiz mumkin.

1. Quyilgan vazifani, uni matematik modelini yaratish vaqtidaek chukur, to'liq va yaxshi tushunishga intilish kerak. Bunday intilishlar dastur ob'ektlari o'rtasidagi mantiqiy o'zaro tasirlarni tushunishga olib keladi.

2. Vazifalarni echish algoritmi EVM da hisoblash jarayonlarini kobiliyatini to'liq hisobga olgan xolda ishlab chiqiladi.

3. Algoritmni loyihalash vaqtida iloji boricha tushunarli va sodda bo'lishiga intilish kerak. Bu dasturni yozishdagi dasturlash tili shakliga ham bogligdir. Tarkibiy dasturlashda qo'llaniladigan standart usullar, dasturni tushunarli qiladi, ammo bir xilgi xollarda u nozik va samarasiz bo'lib kolish xollari ham uchraydi. Shuning uchun sodda va tushunarli dastur samarali dasturga nisbatan ustun turadi. Tarkibiy dasturlash tizimli endashishning asosiy printcplariga asoslanadi: dastur mayda qadamlardan tashkil topishi kerak. Qadam o'lchamlari dasturchi tomonidan shu qadamda qo'llaktgan echimlar soniga asosan ulchanadi. Murakkab vazifalar etarli ravishda sdda, qabul qilishga oson bo'lgan

blokka bo'linadi. Ularning har biri bittadan kirish va chiqishga ega bo'ladi. Dasturning mantiqiy oddiy tarkibiy boshqaruvchi bazalarning eng oz soniga tayanadi. Tarkibiy dastur bittadan kirish va chiqishdan iborat bo'lgan ketma-ket yoki bir-biriga kiydirilgan bloklar tizimidan tashkil topgan. Bunda bloklar o'lchami dasturlash tili (operator) ning elementar tallifi bugimigacha etadi. Eng sodda dasturga ega bo'lish uchun, siz loyihalash vaqtida ma'lumotlar tarkibini aniqlab olishingiz kerak. Chunki bu tarkiblar ustidan ishlagandan so'ng siz intilgan natijaga erishasiz. Dastur ma'lumotlar bilan algoritmlar o'rtasidagi teskari tasvir sifatida ko'rinishi kerak. Dasturda kiritish chiqarish jarayonlari, tashkil etishga, dastur tashqarisidan kiritiladigan boshlangich ma'lumotlarni raspechatkasi va yozishda ma'lumotlar tarkibini va ularni interpretatsiyasini tablitca6 grafika va x. k sifatida ko'rsatish kerak. O'zgaruvchilarni initializatsiyalash ni ulardan foydalanishdan oldin bajarish lozim. Ma'lumotlarni to'g'riligini, ularni algoritmgga kiritishda tekshirish dasturni sifatini oshiradi. O'zgaruvchilarni solishtiruvchilari maonoli yuklashlardan iborat bo'lishi kerak. Juda ko'p xollarda o'zgaruvchilarni tasvirlovchilar dasturda oson rasshifrovkalovchi so'zlarning bosh harflaridan tashkil topgan bo'ladi.

Boshlangich ma'lumotlarni tubanlashishida dasturni to'g'ri bajarilishiga intilish kerak. Bunda saralovchi dastur bitti elementdan tashkil topgan ro'yxatni saralashi kerak; matritcalar bilan ishlovchi dastur matritca bitta elementdan tashkil topganda kayta shakllanishi kerak; noto'g'ri boshlangich ma'lumotga ega bo'lgan dastur xatolar borligi xaqida xabar beruvchi vazifa bilan tugashi kerak. Dasturni o'qish oson kechishi uchun aloxida so'zlar va jumlar urtasida joy(probel) qoldirish maslaxat beriladi.

Dasturni tarkibiy qismida dastur to'g'rismda ma'lumot beruvchi ilovalar bo'lishi kerak.

Nazorat savollari:

1. Dastur maxsulotini tayyorlash vositalari va komponentlariga tushuncha bering.
2. Dasturiy maxsulotni loyihalash usullarini vazifasini ko'rsating.
3. Dasturni tesdan o'tkazishning maqsadi
4. Qanday testlash uslublarini (usullarini) bilasiz va ularning vazifalariga xarakteristika bering?

17-ma`ruza. Dasturiy maxsulotni foydalanisga chiqarish.

R E J A :

1. Dasturiy maxsulotlarni konstruktciyalash.

2. Dasturiy maxsulotlarni baholash.

3. Ishlatilishi.

Adabiyotlar

1. Van Tassel D. Stil, razrabotka, effektivnost, otladka i ispytanie programm. M. , Mir, 1991.
2. Linger R. , Teoriya i praktika strukturnogo programmirovaniya M. , Mir, 1985.
3. V. V. Lipaev. "Proektirovanie programmných sredstv", M. : "VSh", 1991.
4. Foks Dj. "Programmnoe obespechenie i ego razrabotka". Per. s angl. M. : Mir, 1985.
5. S. S. Gulomov. "Iqtisodiy informatika". Toshkent 1999 y.

Tayanch so'zlar: *Dasturiy maxsulotni loyihalash va analiz qilish usullari. Kichik xajimdagi dasturlarni ishlab chiqarish usullari. Katta xajimdagi dastur loyihalarni tadbik qilish yo'llari.*

Foydalanish xususiyati va foydalanuvchilar kategoriyalariga ko'ra barcha dasturlarni ikki sinfga - utilitar dasturlar va dasturiy maxsulotlarga bo'lish mumkin.

Utilitar dasturlar - shu dasturlarni ishlab chiqaruvchilar extiejini qondirish uchun mo'ljallangan. Ko'proq utilitar dasturlar ma'lumotlarni kayta ishlash texnologiyasida servis rolini bajaradi yoki keng tarkalish uchun muljallanmagan funktsional vazifalarni xal etish dasturlari bo'ladi.

Dasturiy maxsulotlar foydalanuvchilar extiyojlarini qondirish, keng tarqatish va sotish uchun mo'ljallangan.

Xozirgi paytda dasturiy maxsulotlarni ochik (legal) tarqatishning boshqa variantlari ham mavjud, ular yalpi (global) va mintaqaviy kommunikatsiyalardan foydalanish bilan yuzaga keladi.

Freeware-erkin tarkatiladigan foydalanuvchining uzi kullab-kuvvatlaydigan bepul dasturlar, u bularga zaruro'zgartirishlar kiritishga xakli.

Shareware-notijorat (shartli-tulovsiz) dasturlar, ulardan odatdatulovsiz foydalanish mumkin. Bunday maxsulotlardan doimiy foydalanilganda muayyan summa badal tulanadi.

Bir qator ishlab chiqaruvchilar OEM-dasturlar (Original

Equipment Manufacturer), ya'ni kompyuterlarga o'rnatilgan e hisoblash texnikasi bilan birgalikda keltirilgan maxsus dasturlardan foydalaniladi.

Dasturiy maxsulot foydalanishga tegishli ravishda tayerlanishi zarur texnik hujjatlariga ega bo'lishi, shuningdek davlat ro'yxati kodi mavjud bo'lishi lozim. Faqat shunday sharoitlardagina yaratilgan dasturiy majmua dasturiy maxsulot deb nomlanishi mumkin.

Dasturiy maxsulot-sanoat maxsulotining istalgan turi kabi realizatciyaga tayerlangan ommaviy extiyojli muayyan muammo (vazifa)ni xal etish uchun o'zaro bog'langan dasturlar majmuasidir.

Dasturiy maxsulotlar quyidagtcha yaratilishi mumkin:

buyuormaga ko'ra individual ishlanma;

foydalanuvchilar orasida ommaviy tarqatish uchun ishlanma.

Individual ishlanmada firma-ishlab chiqaruvchi muayyan buyurtmachi uchun ma'lumotlarni kayta ishlash o'ziga xosligini hisobga oluvchi dasturiy maxsulotni yaratadi.

Ommaviy tarqatish uchunishlanmani yaratishda firma-ishlab chiqariuvchi, bir tomondan, ma'lumotlarni kayta-ishlashni bajarilaetgan funkctiyalar universalligi, boshqa tomondan, muayyan bir qo'llash sharoitida dasturiy maxsulotning moslanishi va sozlanishini ta'minlash lozim. Dasturiy maxsulotlarning ajralib turuvchi xususiyati uning tizimlili-gi-jamlikda qo'llanilgan xolda amalga oshiriladigan kayta ishlash vazifalarining funktcional tlakonligi va tugalligi bo'lmogi lozim.

Dasturiy maxsulot dasturlashtirishning zamonaviy asbobsozlik vositalari qo'llangan xolda loyiha ishlarini bajarish sanoat texnologiyasi asosida ishlab chiqiladi. Uning o'ziga xaosligi axborot va asbobsozlik vositalaridan foydalanishni kayta ishlash xususiyatiga bog'liq xolda algoritm va dasturlarni ishlab chiqish jarayonining noebligidadir. Dasturiy maxsulotlarni yaratishga ko'plab mexnat, moddiy, moliyaviy zaxiralar talab etiladi; yuqori malakali mutaxasislar zarur.

Dasturiy maxsulotni tayyorlash (kuzatish) -dasturiy maxsulot ishga layoqatligini qo'llab-quvvatlash, unga yangi versiyalar, o'zgartirishlar kiritish, topilgan xatolarni tugralash va xokazolarni uz ichiga oladi.

Dasturiy maxsulotlar an'anaviy dasturiy maxsulotlardan farkli ravishda dasturlarni yaratishda beriladigan sifat xususiyatlarining kat'iy belgilangan turkumiga ega emas yoki bu xususiyatlarni oldindan aniq ko'rsatish yoki baholash mumkin emas, chunki dasturiy vosita ta'minlaydigan bir xil kayta ishlash vazifalari turli ichki ishlanmalarga ega bo'lishi mumkin. Xatto dasturiy maxsulotlarni ishlab chiqishga sarflanadigan vaqt va hujjatlarni ham oldindan katta aniqlikda belgilash mumkin emas.

Dasturlarni asosiy tavsiflariquyidagilar;

- * algoritmik murakkablik(axborotni kayta ishlash algoritm mantiqi);
- * kayta ishlashning amalga oshirilgan vazifalari ishlanmalar-ning tarkibi va chuqurligi;
- * kayta ishlash vazifalarining to'laqonligi va tizimlilik;
- * dasturlar fayllari hajmi;
- * dasturiy vosita tomonidan kayta ishlashning operayion tizimi va texnik vositalarga talblar;
- * diskli xotira hajmi;
- * dasturni tushurish uchun operativ xotira o'lchami;
- * protcessorlar turlari;
- * operatcion tizimlar versiyalari;
- * hisoblash tarmoklarining mavjudligi va boshqadar.

dasturiy maxsulotlarning sifat kursatkichlari xilma-xil, ular quyidagi jixatlarda aks etadi:

- * dasturiy maxsulotni qanchalik yaxshi (oddiy, ishonchli, samarali) foydalanish mumkinligi;
- * dasturiy maxsulotdan qanchalik engil foydalanish mumkinligi;
- * dasturiy maxsulotni qullashda sharoit o'zgarganda undan foydalanish mumkinligi yoki yo'qligi va boshqadar.

Xarakatchanlik-dasturiy maxsulotlarda ma'lumotlarni kayta ishlash tizimi texnik majmuasi, opertcion muxit, ma'lumotlarni kayta ishllashning tarmok texnologiyasi predmetli soxa o'ziga xosligi va xokazolardan mustakillikni anglatadi. Xarakatchan (ko'p

platformali) dasturiy maxsulot hisoblash tarmogi sharoitida foydalanishga xech qanday cheklanishlarsiz kompyuterlar va operatcion tizimlarning turli modellariga urnatilishi mumkin. Bunday dasturiy maxsulotlarni kayta ishlash funktsiyalari biron-bir o'zgarishsiz ommaviy foydalanish uchun makbuldir.

Ishonchlilik-dasturiy maxsulot ishida uzliksizlik va barkarorlik, kayta ishlashni bajarishning aniqligi, ish jarayonidagi xatolarni oldindan bilish bilan belgilanadi.

Samaradorlik-dasturiy maxsulot faoliyatida uning ham bevosita vazifasi-foydalanuvchi talabi, ham foydalanish uchun zarur bo'lgan hisoblash zaxiralari xarajatlari nuqta`i nazaridan baholanadi.

Inson omilini hisobga olish oxirgi foydalanuvchi uchun dustona interfeysni ta`minlash, dasturiy vosita tarkibida kontekstli-bog'liq xolda aytib beruvchi yoki ukituvchi tizim, dasturiy vositaga kiritilgan funktsional imkoniyatlarni uzlashtirish va ulardan foydalanish uchun yaxshi hujjatlar mavjudligini anglatadi.

Modifikatsiyalanganlik-dasturiy maxsulot ishida o'zgartirishlar kiritish, masalan, kayta ishlash funktsiyalarini kenggaytirish, kayta ishlashning boshqa texnik bazasiga o'tish va boshqadarga kobillikni anglatadi.

Kommunikativlik-dasturiy maxsulot ishida boshqa dasturlar bilan eng ko'p integratsiyasi, umumiy takdim etish shakllarda ma`lumot almashuvini ta`minlash(ma`lumotlar bazasining eksport va importi, kayta ishlash ob`ektlarini tadbik etish yoki boglash va boshqadarga asoslangan).

Dasturiy maxsulotlar bozori mavjudligi sharoitlarida muxim xususiyatlar quyidagilardir:

- * narx;
- * sotilgan maxsulot soni;
- * bozorda turish vaqti(savdo davomiyligi);
- * firma-ishlab chiqaruvchi va dasturlari mashxurligi;
- * xuddi shunday dasturiy maxsulotlarning mavjudligi.

Ommaviy tarkatiladigan dasturiy maxsulotlar bozor extiyoji va kon`yunkturasini (rakiblar dasturlari mavjudligi va narxlari) hisobga oluvchi narxlarda sotiladi. Firma olib boradigan marketing ishlari katta ahamiyatga ega, u quyidagilarni uz ichiga oladi:

- * bozorni egallash uchun narx-navo siyosatini shakllantirish;

- * dasturiy maxsulotni keng reklama qilish;
- * dasturiy maxsulotni sotish uchun saovdo tarmoklarini barpo qilish(dilerlik va distribyuterlik markazlari deb ataladi);
- * dasturiy maxsulot foydalanuvchilarga kafolatli xizmat ko'rsatishni ta'minlash, tezkor liniyani yaratish(dasturiy maxsulotlardan foydalanish jarayonida yuzaga kelgan savollarga tezkor javob);
- * dasturiy maxsulot foydalanuvchilarini ukitish.

Nazorat savollari:

1. Foydalanish xususiyati va foydalanuvchilar kategoriyaga ko'ra barcha dasturlar necha sinfga bo'linadi?
2. Dasturiy maxsulot bu nima?
3. Dasturlarni asosiy tavsiflarini ko'rsating.
4. Dasturni tesdan o'tkazishning maqsadi
5. Qanday testlash uslublarini (usullarini) bilasiz va ularning vazifalariga xarakteristika bering?

18-ma`ruza. Dasturlarni hujjatlashtirish.

R E J A :

1. Dastur kuzatuvchini hujjatlarini asosiy turlari.

2. Hujjatlashtirish tashki spetsifikatciyasi.

Adabiyotlar

1. Van Tassel D. Stil, razrabotka, effektivnost, otladka i ispytanie programm. M. , Mir, 1991.
2. Linger R. , Teoriya i praktika strukturnogo programmirovaniya M. , Mir, 1985.
3. V. V. Lipaev. "Proektirovanie programmných sredstv", M. : "VSh", 1991.
4. Foks Dj. "Programmnoe obespechenie i ego razrabotka". Per. s angl. M. : Mir, 1985.
5. S. S. Gulomov. "Iqtisodiy informatika". Toshkent 1999 y.

Tayanch so'zlar: *Dasturiy maxsulotni loyihalash va analiz qilish usullari. bosqich, real dastur.*

Hujjatlashtirish, misol keltirishni bosqich boshidan bir vaqtda dastur ishlab chiqarishdan boshlanishi lozim. Ishlab chiqarilgan dastur dan foydalanish uchun kerakli qo'llanma utilishi kerak. Yomon hujjatlashgan dastur bir guruh mutaxasislar orkali tekshirilishi kerak. Ular chala hujjatlarni kushishlari va eski raspechatka sifatidagi hujjatlarni urganishlari kerak.

Dasturlarni ko'chirish.

Ko'chirish samarasi va tez sozlanuvchi dasturni tayerlashga yordam beradi. Katta dasturlarda ishlashda, ulaprning asosiy qismni modellashtirish uchun kichik dasturlarni yozish maqsadga muvofik bo'ladi.

Katta dasturlar uchun keng kiritish-chiqarish va dasturlash bilan testdan o'tkazish uchun ko'p vaqt kerak bo'lishi mumkin. Modelda kiritishning bo'lmasligi va soddalashtirilgan chiqarishlar bo'lishi mumkin. Bu o'rtacha dastur boshqacha ishlovchi real dasturga karaganda ikki tartibga ozroq bo'lishi kerak bo'ladi. Yakuniy echim yakini topilganda ham, model asosiy yordam berishi kerak bo'ladi. Model dasturni har qanday tilda yozishi mumkin va loyiha hamda algoritim tekshirishda foydalanish mumkin.

Dasturni kayta yozishda yoki boshqa xollarda, tekshirish usullaridan tizimni nazorat qilishni qo'llashdir, ya'ni har bir dastur uning avtoriga va kamida bitta dasturchiga tushunarli bo'lishi kerak.

Xar bir dastur loyihalanaetgan dasturiy buyumga hujjatlar ham birgalikda bajarilishi kerak. Hujjatlarni va dasturga kerakli bo'lgan talablarni unifikatsiyalash maqsadida ular yagona dasturiy hujjatlar tizimsiga (ESPD) standartlashtirilgan. Uning tarkibiga turi va shaklini aniqlab beruvchi standartlar, hujjatlar tarkibi, uni kuzatib borish, yana qo'shimcha sifatida dasturiy hujjatlarni avtomatlashtirilishini ta'minlovchi standartlar kiradi.

Dastur uchun bajarilgan hujjatlar dasturni o'tkazish hamda uzatish uchun foydalaniladi va dastur tayyorlashda qo'llaniladi. Dasturiy hujjatlar tarkibiga spetsifikatsiyalar, asosiy nusxalar verimosti, dastur teksti, dastur tarkibi, dasturchi boshqarishi, til tarkibi, texnik xizmat boshqarishi.

Spetsifikatsiya - dastur tarkibini va uning hujjatlarni uz ichiga oladi. Asosiy nusxalar egasining verimostlarida dasturiy hujjatlar saklovchi korxonalar tarkibi kursatilgan. Kerakli tushuntirishlarga ega bo'lgan dasturiy tekst ham dasturiy hujjatlar hisoblanadi. Bu hujjat bir yoki bir necha qismli tekstlarni uz ichiga oladi. Uning har birida dastur qaysi bir algoritmik tilning belgilari yordamida yozilgan bo'ladi. Dastur tarkibi quyidagilarni uz ichiga oladi. Umumiy ma'lumotlar, funktsional vazifasi, mantiqiy tarkibini tartibitashki ichki aloqalar, tikrovchi ma'lumotlarni chiqarish va yuklanish buyicha instruktsiyalar, chiquvchi ma'lumotlar.

Sinash uslubikasi va dasturini nomli hujjat sinashdagi teshshirishlarni nazorat kishlash usllari va taktlari, dasturiy tavsifiga talablarni uz ichiga oladi.

Texnik vazifa dasturi kushilaetgan talablarni aniqlanib beriladi. Tushuntirish hajmida algoritm tartibi dasturini funktsiyalashtirish kursatilgan.

Kolgan hujjatlar ekspluatatsiya kilinuvchi qatorga kiradi. Qo'llanish tartibi dasturni qo'llanish chegaralari xaqida, vazifani echish sinfi va usuli xaqida , dasturni vazifasi to'g'risidagi ma'lumotlarni uz ichiga oladi. Tartibiy dasturchi boshqaruv dasturi tarkibi xaqidagi ma'lumotlarni, aniq bir qo'llanish sharoitlarga sozlash xaqida ma'lumotlar kursatilishi, sozlash vaqtida chiqarilaetgan xabarlar teksti, dastur tekshirilganligi xaqida ma'lumot. Dasturchining boshqaruv dasturni ishlatishda kerak bo'lgan ma'lumotlarni hamda bajarilishi vaqtidagi shartlar(operativ xotira hajmi, pereferiyani moslama o'lchamlari va tarkibiy kushgan talablar, dasturiy ta'minlashga quyilgan talablar) umumiy tavsiflarni tarkibiy va dastur xususiyatlari(vaqtincha tavsifi, ishlash rejimi, dasturni to'g'ri

bajarilishini nazoratg' qilish usullari, ma'lumot o'lchamlarini uzatishchikish va kirish jarayonlarini tashkil etish tarkibi kiradi.

Operator yuriknomasi dasturni bajarish vaqtida EXM bilan operator o'rtasidagi mulokot shartlarini uz ichiga oaldi. Ya'ni apparat vositalari tartibi, yuklatish vaqtida operator xarakatlari ketma-ketligi, yoki dasturni bajarish va tuzatish, operator buyruqlarni taxminiy formalar, dasturni buyruqlarga bo'lgan javoblari, operatorga bo'lgan xabarlar teksti va uning mumkin bo'lgan xarakatlari tartibi. Dasturlash tilida ezilishning maxsus standart, vazifalarini boshlanish, hisoblash jarayonini tashkil etish va xokazo). Dastur bilan foydalanuvchi o'rtasidagi mulokot tilni aniqlab beradi va tilni tavsifini, sintaksizini ma'lumotlar simantiqni tilning tarkibiy elementlarini, standart funktsiyalarni, tildagi dasturni sozlash vositalar tarkibini nazarda tutadi.

Texnik xizmat qilish buyicha qo'llanma texnik xizmat kursatuvchi vositalardan qo'llaniladigan tekstli va diagnostikali dasturlarni uz ichiga oladi va faqat shu dasturlarda qo'llaniladi.

Spesifikatsiyadan tashqari u yoki bu hujjatga bo'lgan muxtojlik dasturga quyilagn texnik vazifada aniqlanadi.

Dasturiy hujjatlar har xil ko'rinishda(kogozda, magnit lentalarda, magnit disklarda va x. k)bo'ladi va ular quyidagi qismlardan iborat bo'ladi: titulli, ma'lumotli, anotatsiya, xulosa asosiy va qism registratsiyasi.

O'z-o'zigani hujjatlashtiruvchi dasturlar boshida qo'llash va kuzatib borish instruktsiyalarini uz ichiga oladi. Bu instuktsiyalarda kiruvchi va chiquvchi kattaliklar, massiv va o'zgaruvchilar belgilanuvchi va ularni maonosi beriladi. Ayrim xollarda dastur tarkibini algoritm tilida ko'rsatish ham maqsadga muvofikdir.

Nazorat savollari:

1. Dasturni xujatlashtirish qaysi vaqtdan boshlanishi kerak?
2. Dastur kuzatuvchini xujatlarini asosiy turlarini ko'rsating?
3. Dasturiy hujjatlar qanday ko'rinishda bo'ladi va ular qanday qismlardan iborat bo'ladi?
4. Dastur maxsulotini tayyorlash vositalari va komponentlariga tushuinchcha bering.
5. Dasturmiviy maxsulotni loyihalash usullarini vazifasini ko'rsating.

