

8-Mavzu.Ma'lumotlar bazasini boshqarish tizimlarida xavfsizlik auditini o'tkazish xususiyatlari

Axborot tizimi yoki axborot texnologiyasining auditida deganda tizimning, texnologiyaning joriy holati, unda kechuvchi texnologiyalar va hodisalar xususidagi obyektiv ma'lumotlarni olish va baholashning, ularning ma'lum mezonlarga moslik darajasini o'rnatish va natijalarni buyurtmachi taqdim etishning tizimli jarayoni tushuniladi.

Audit o'tkazilishi axborot texnologiyasining joriy xavfsizligini baholashga, xavf-xatarlarni baholash va boshqarishga, ularning tashkilot biznes jarayonlariga ta'sirini boshqarishga, tashkilot axborot aktivlarining xavfsizligini ta'minlash masalasiga to'g'ri va asoslangan yondashishga imkon beradi.

Tashkilotning asosiy aktivlari

Tashkilotning asosiy aktivlari quyidagilar:

- g'oyalar;
- bilimlar;
- loyihalar;
- ichki tekshirish natijalari.

Auditning umumiy tushunchasi

1844-yili Angliyada aksioner shirkatlar xususida qonun qabul qilingan. Ushbu qonunga binoan shirkat boshqarmasi har yili aksionerlar oldida hisob berishi lozim. Buning ustiga hisobot maxsus kishi mustaqil auditor tomonidan tekshirilishi va tasdiqlanishi shart. Ushbu yil auditning tug'ilgan yili hisoblanadi.

Hozirda audit o'z rivojining bir necha bosqichlarini o'tib, davlatlar xo'jalik hayotining qismiga aylandi. Aksionerlik shirkatining buxgalteriya hisoblarini alohida professional auditorlar tekshirishidan boshlab audit tarkibida professional auditorlar va auditorlik firmalar ko'rsatuvchi qator xizmatlar (buxgalteriya hisobotini tekshirish, moliyaviy tahlil, maslahat berish) bo'lgan kompleks tushunchagacha rivojlandi. Bunday firmalarning orasida o'nlab xodimlari bo'lgan katta bo'lmaganlari va minglab xodimlari bo'lgan – juda kattalari mavjud.

Tashqi audit

Axborot tizimi xavfsizligining auditini odatda, tashqi va ichkilariga ajratishadi. *Tashqi audit* asosan tashkilotdan tashqarida va odatda, axborot xavfsizligi auditida bilan shug'ullanuvchi ixtisoslashgan tashkilotlar tomonidan o'tkaziladi. Bunda tashqi hujumlar natijasidagi xavf xatar o'chamlari tahlillanadi (xatto tashkilot tarmoqlararo ekranlar bilan himoyalangan bo'lsa ham). Tashqi auditni o'tkazishdoportlarni skanerlash, tarmoq va tatbiqiy dasturiy taininot zaifliklarini qidirish amalga oshiriladi. Web - serverlari, pochta va fayl serverlari bilan o'zaro bog'lanishga hamda tashkilot lokal tarmogiga kirishga urinishlar amalga oshiriladi. Tashkilot rahbariyatining xohishi bilan tashqi auditning maxsus turi - Ethical Hacking o'tkazilishi mumkin. Bunda maxsus tashkilot (bunday tashkilot maxsus

Tiger Team nomiga ega) tashkilot serverlariga, saytlariga va xostlariga tanlab olingan hujumlarni amalga oshiradi. Bunday hujumlar tashkilot axborot tizimining zaifliklarini namoyish etishi mumkin.

Ichki audit

Ichki audit odatda tashkilot xodimlaridan tashkil topgan maxsus komanda tomonidan o'tkaziladi. Ichki auditning vazifasi mavjud axborot tizimi texnologiyasidan foydalanishdagi xavf xatarni baholash hisoblanadi. Auditning bu turi qandaydir standartni amalga oshiruvchi auditni avtomatlashtirish vositasini jalb qilish orqali bajariladi. Ichki audit tashkilotning tarmoqlararo ekran bilan chegaralangan tarmoq muhitining ichida o'tkaziladi. Tashkilotning ichki xost portlarini va zaifliklarini skanerlash ham uning vazifasi hisoblanadi. Undan tashqari, tashkilotda o'rnatilgan xavfsizlik siyosatining bajarilishi, resurslardan foydalanishning nazorati va boshqarilishi, tashkilot xodimining parol siyosati va uning bajarilishi tahlillanadi. Auditning bu turi audit o'tkazishning standart usulini tarmoq zaifliklarini mukammal ko'rib chiqish bilan todiradi.

Ma'lumotlar bazasi xavfsizligi auditini o'tkazish

Oracle MBBT misolida ma'lumotlar bazasi xavfsizligi auditini

o'tkazish. Oracle MBBT - fiinksonai rivojlangan mahsulot hisoblanadi va unda audit o'tkazishning bir necha imkoniyatlari mavjud. Oracle auditini ma'lumotlar bazasi tarkibidagi axborotdan avtorizatsiyalanmagan foydalanishni yoki axborotning ichki suiiste'mol qilinishini aniqlashda yordam berishi mumkin.

Oracle dagi audit uchta qismga ajratilgan:

- CREATE TABLE yoki CREATE SESSION kabi iboralarning auditini;
- ALTER USER imtiyozlar auditini;
- SELECT TABLE obyekt sathidagi obyektga audit.

Asosiy konfiguratsiya

Audit yozuvini ma'lumotlar bazasining auditorlik jadvaliga yoki operatsion tizimning auditorlik jumaliga joylashtirish mumkin. Audit yozuvining operatsion tizim jumaliga yozilishi ba'zi hollarda himoyalashning yuqoriroq darajasini ta'minlashda, ushbu imkoniyat barcha platformalar uchun mumkin emas.

Ma'lumotlar bazasiga yozishda audit init.ora fayliga quyidagi qatomi qo'shish orqali ishga tushiriladi.

```
audit_trail=db
```

Auditning komanda formatlari

Oracle hujjatlari bo'yicha auditning barcha komanda formatlari quyidagi ko'rinishga ega:

```
audit { statement_option/privilege_option }  
[by user] [by { session/access }] [whenever  
{ successful/unsuccesful } ]
```

Ushbu ifodaning faqat statement option va privilege_option qismlari majburiy hisoblanadi, qolgan qismlari qo'shimcha parametrlar bo'lib, ularning ishlatilishi auditni yanada o'ziga xos bo'lishiga imkon beradi.

Foydalanuvchi audit komandasini berishi uchun unda "AUDIT SYSTEM" imtiyozi bo'lishi shart. MDSYS, CTXSYS va WKSYS foydalanuvchilar hujumchi uchun yaxshigina nishon bo'lishlari mumkin, chunki qilinuvchi har qanday harakatlarni bekitish uchun foydalanuvchilarning ixtiyoriy biri tomonidan auditning har qanday harakati to'xtatilishi mumkin. Endi qandaydir foydalanuvchi tizimga kirib, o'z ishini boshlasa, audit foydalanishning barcha urinishlarini kuzatadi.

Ma'lumotlar bazasida o'zgarishlarni nazorat qilishga auditni o'rnatish misoli.

Ushbu misolda qisqalikni ta'minlash maqsadida, ma'lumotlar bazasi obyektlaridagi barcha o'zgarishlar kuzatilmaydi. Garchand, umuman, ma'lumotlar bazasida har qanday obyektlaridagi o'zgarishlarni kuzatish mumkin. Mazkur misolda audit obyektlarning tanlangan guruhiga ishga tushiriladi.

Auditni sozlash quyida ko'rsatilganidek, ikki bosqichda, audit komandalarini yaratish va bajarish uchun ishga tushirish orqali amalga oshirilishi mumkin:

```
set head off
```

```
set feed off
```

```
set pages 0
```

```
spool aud.lis
```

```
select 'audit 7/name/7;'
```

```
from system_privilege_map
```

```
where (name like 'CREATE%TABLE%')
```

Ushbu skript/ssenariy audit komandalarini bajarish uchun ishga tushiriladi.

Audit uchun qandaydir imtiyozlar yoki iboralar ishlatilganligini tekshirishda quyidagilar ishlatiladi:

```
SQL>select from dba_stm_audit_opts
```

```
2 union
```

```
3 select from dba_priv_audit_opts
```

```
no rows selected
```

```
SQL.
```

Qanday obyektlar nazoratlanganligini aniqlash uchun dba_obj_audit_opts topshiriqini so'rash lozim.

Predmetli sohada loyihadan oldingi tadqiqotlar

Loyiha oldi tadqiqotlarning maqsadi bo'lg'usi biror narsaga mo'ljallangan dasturiy ta'minot haqidagi umumiy noravshan bilimlarni unga nisbatan aniq talablarga aylantirishdan iborat.

Noaniqlikning ikki varianti mavjud:

- shakllantirilayotgan vazipredmeti hal etish usullari noma'lum, odatda ilmiy-texnik vazifalarni hal etishda shunday tipdagi noaniqliklar yuzaga keladi;
- avtomatlashtiriluvchi axborot jarayonlarning tuzilmasi noma'lum, odatda korxonalar boshqaruvining avtomatlashgan tizimlarini barpo etishda uchraydi.

Birinchi holat

Birinchi holatda loyihadan oldingi tadqiqotlar vaqtida qo'yilgan vazipredmeti hal etish imkoniyati va talab qilinuvchi natijani olishga imkon beradigan usullar belgilanadi, bu esa ham fundamental, ham amaliy mazmundagi tegishli ilmiy-tadqiqotlarni, real dunyo obyektlari yangi modellafining ishlab chiqilishi hamda tadqiq qilinishini talab etishi mumkin.

Ikkinchi holat

Ikkinchi holatda quyidagilar belgilanadi:

- avtomatlashtiruvchi va axborot jarayonlarining tuzilmasi hamda o'zaro aloqasi;
- inson va tizim o'rtasidagi, shuningdek apparatlar va dasturiy ta'minot o'rtasidagi funksiyalar taqsimoti;
- dasturiy ta'minot funksiyalari: uning ham foydalanuvchilar bilan, ham zaruratga ko'ra apparatlar bilan birga ishlashining tashqi shart-sharoitlari va uni interfeysining xususiyatlari;
- dasturiy va axborot komponentlariga talablar, zarur apparat resurslari, ma'lumotlar bazalariga talablar hamda dasturiy komponentlarning fizik tavsifnomalari. Predmetli sohaning loyihadan oldingi tadqiqotlari natijalaridan texnik topshiriqni ishlab chiqish jarayonida foydalaniladi.

Texnik topshiriqni ishlab chiqish

Texnik topshiriq ishlab chiqishning asosiy maqsadlari, dasturiy mahsulotga talablar shakllantirilgan, ishlab chiqish muddatlari va bosqichlari belgilangan hamda qabul, topshirish sinovlari jarayoni reglamentlangan hujjatni o'zida ifoda etadi.

Texnik topshiriqni ishlab chiqishda buyurtmachi vakillari ham, ijrochi vakillari ham ishtirok etishadi. Ushbu hujjatning asosida buyurtmachining boshlang'ich talablari, texnika ilg'or yutuqlarining tahlili, ilmiy-tadqiqot ishlarining, loyihadan oldingi tadqiqotlarning, ilmiy bashorat (prognozlash)ning natijalari va hokazolar turadi.

Sxematik omillar

Ishlab chiqiluvchi dasturiy ta'minot tavsiflarini belgilovchi omillar sxematik ravishda ko'rsatilgan. Bunday omillar quyidagilardir:

- dastur yoki tizim funksiyalarini belgilaydigan boshlang'ich ma'lumotlar va talab qilinuvchi natijalar;
- funksiyalar bajarish muhjt (dasturiy va apparatli) yuklatilishi ham, texnik topshiriqda ko'rsatilgan parametrlarni ta'minlash uchun tanlanishi ham mumkin;

- boshqa dasturiy ta'minot yoki maxsus texnik vositalar bilan mumkin bo'lgan o'zaro harakatlanishi ham, bajariluvchi funksiyalar majmuidan kelib chiqqan holda tanlanishi ham mumkin.

Texnik topshiriqni ishlab chiqish quyidagicha davomiylilikda bajariladi. Eng avvalo, bajariluvchi funksiyalar majmui tayinlanadi, so'ngra natijalar ro'yxati, ularning tavsifnomalari va taqdim etish usullari belgilanadi. Keyin esa dasturiy ta'minot funksiyalari bajarilishining muhiti aniqlanadi; texnik vositalarning muayyan komplektlashuvi va parametrlari, foydalaniluvchi operatsion tizim versiyasi hamda bo'lg'usi dasturiy mahsulot o'zaro harakatlanishi joiz bo'lgan boshqa o'rnatilgan dasturiy ta'minlash ning versiyalari va parametrlari belgilanadi.

GOST standarti

Ishlab chiqilayotgan dasturiy ta'minot ayrim axborotni to'playdigan va saqlaydigan yoki boshqaruvga qandaydir texnik jarayon bilan qo'yiladigan hollarda ham uskuna, ham energiya ta'minoti izdan chiqishlari holatlarida dastur ishlashini aniq reglamentlash zarur.

Texnik topshiriq uchun GOST «Texnik topshiriq. Mazmun va rasmiylashtirishga-talablar» standarti mavjud. Ushbu standartga muvofiq texnik topshiriq quyidagi bo'limlardan iborat bo'lishi shart:

- kirish;
- ishlab chiqish uchun asos;
- ishlab chiqish ahamiyati;
- dasturga yoki dasturiy jihozga talablar;
- dasturiy hujjatlarga talablar;
- texnik-iqtisodiy ko'rsatkichlar;
- ishlab chiqish davrlari (stadiya) va bosqichlari;
- nazorat va qabul tartibi.

“Kirish”

Zaruratga ko'ra texnik topshiriqqa ilovalar qo'shishga yo'l qo'yiladi.

Kirish dastur yoki dasturiy mahsulot nomini va ularning o'llanish sohasi qisqacha tavsifnomasini, shuningdek ulardan foydalanish ko'zda tutilayotgan obyekt (masalan tizim) nomi hamda tavsifnomasini qamrab olishi shart. Kirishning asosiy ahamiyati tegishli ishlab chiqish dolzarbligini namoyish etish va ushbu ishlab chiqish, o'xshash ishlab chiqishlar qatorida qanday o'rin egallashini ko'rsatish. Ishlab chiqish uchun asos bo'limi ishlab chiqish yuritilishiga asos hujjat va ushbu hujjatni tasdiqlagan tashkilot nomini hamda ishlab chiqish mavzusining nomini yoki shartli ifodasini o'z ichiga olishi darkor. Reja, buyruq, shartnoma va hokazolar asos hujjat tariqasida xizmat qilishi mumkin.

Ishlab chiqish ahamiyati foydalanuvchilar toifalari ko'rsatilgan holda dasturiy mahsulot funksional va tasarrufiy ahamiyatining bayonini qamrab olishi shart.

Funksional tavsiflarning talablari

Dasturga yoki dasturiy jihozga talablar bo'lishi quyidagi tagbo'limlarni o'z ichiga olishi dardkor:

- funksional tavsiflarga talablar;
- ishonchlilikka talablar;
- ishlatish sharoitlari;
- texnik vositalar tarkibiga va parametrlariga talablar;
- axborot va dasturiy mutanosiblikka talablar;
- rusumlashtirish va o'ramlashtirishga talablar;
- transportlashtirish va saqlashga talablar;
- maxsus talablar.

Yuqorida sanalganlar qatorida funksional tavsiflarga talablar tag bo'limi eng muhimdir. Ushbu bo'limda bajariluvchi funksiyala sanalishi va boshlang'ich ma'lumotlar hamda natijalarning tarkibi, tavsifnomalari, taqdim etish shakllari bayon etilishi shart.

Xuddi shu bo'limda zaruratga ko'ra samaradorlik mezonlari: tizim javobining maksimal mumkin bo'lgan vaqti, foydalaniluvchi operativ yoki tashqi xotira hajmi va boshqalar ko'rsatiladi.

Eslatma !!!

Eslatma. Agar ishlab chiqilgan dasturiy ta'minlash texnik topshiriqda ko'rsatilgan funksiyalarni bajara olmasa, u texnik topshiriqqa nomuvofiq, ya'ni sifat mezonlari nuqtayi nazaridan noto'g'ri hisoblanadi. Bo'lg'usi loyihaning universalligi ham odatda maxsus ta'kidlab o'tilmaydi, lekin ko'zda tutiladi.

Ishonchlilikka talablar tag bo'limida ishlab chiqilayotgan tizim ta'minlashi shart bo'lgan ishonchlilik darajasi va izdan chiqishdan so'ng tizimning tiklanish vaqti ko'rsatiladi. Tizimlar uchun ishonchlilikka odatiy talablar bilan birga ushbu bo'limda ba'zan ishlab chiqilayotgan mahsulotning natijalar ishonchligini orttirish bo'yicha harakatlari (kirish va chiqish axboroti nazorati, oraliq natijalarning zaxira nusxalarini yaratish va hokazo) reglamentlanadi.

Ishlatish sharti tag bo'limida ishlatish shartlariga qo'yiladigan maxsus talablar: atrof-muhit haroratiga, havoning nisbiy namligiga va hokazolarga alohida talablar ko'rsatiladi. Qoidaga ko'ra, agar ishlab chiqilayotgan tizim nostandart shart-sharoitlarda tasarruf qilinadigan yoki maxsus tashqi qurilmalardan, masalan axborotni saqlash uchun qurilmadan foydalaniladigan bo'lsa, shunday talablar qo'yiladi. Xuddi shu joyda xizmat ko'rsatish turi, xodimlarning zaruriy miqdori va malakasi ko'rsatiladi. Aks holda talablar qo'yilmayotganligini ko'rsatishga ruxsat etiladi.

Texnik vositalarning tavsiflari

Texnik vositalarning tarkibi va parametrlariga talablar tag bo'limida texnik vositalarning zaruriy tarkibi ularning quyidagi asosiy texnik tavsiflari aniqlanadi: mikroprotsessor tipi, xotira hajmi,

tashqi qurilmalar mavjudligi va boshqalar qayd etilgan holda ko'rsatiladi, bunda ko'pincha konfiguratsiyaning

ikki varianti: minimal hamda tavsiya etiluvchi konfiguratsiya ko'rsatiladi.

Axborot va dasturiy mutanosiblikka talablar tag bo'limida zaruratga ko'ra hal etish usullarini berish, ishlab chiqish uchun dasturlash tilini yoki muhitni belgilash, shuningdek foydalaniluvchi operatsion tizimni hamda ishlab chiqilayotgan dasturiy ta'minot o'zaro harakatlanishi shart bo'lgan boshqa tizimiy va foydalanuvchilik dasturiy vositalarini ham belgilash mumkin. Xuddi shu bo'limda zaruratga ko'ra axborot himoyasining qanday darajasini ko'zda tutish zarurligi ham ko'rsatiladi. Dasturiy hujjatlarga talablar bo'limida dasturchi qo'llanmasi, tushuntirish yozuvnomasi va boshqalar mavjudligining zarurligi ko'rsatiladi. Hujjatlarning ushbu barcha tiplariga ham GOSTlar mavjud. Texnik-iqtisodiy ko'rsatkichlar bo'limida taxminiy iqtisodiy samaradorlik, ko'zda tutilayotgan yillik ehtiyoj va mavjud analoglarga qiyosan iqtisodiy afzalliklar ko'rsatilishi tavsiya etiladi. Ishlab chiqish stadiyalari va bosqichlari bo'limida ishlab chiqish stadiyalari, bosqichlari va yillar mazmuni ishlab chiqish muddatlari hamda ijrochilar ko'rsatilgan holda bo'ladi.

Nazorat va qabul tartibi bo'limida sinovlar turlari, ish qabuliga umumiy talablar ko'rsatiladi. Ilovalarda zaruriyatga ko'ra quyidagilar keltiriladi: ishlab chiqishni asoslovchi ilmiy-tadqiqot ishlari ro'yxati; ishlab chiqishda foydalanish lozim bo'lgan algoritmlar sxemalari, jadvallar, bayonlar, asosnomalar, hisob-kitoblar va boshqa hujjatlar.

Ishlab chiqilayotgan mahsulot o'ziga xosliklariga bog'liq ravishda bo'lishlar mazmunini aniqlashtirishga, ya'ni tag bo'limlardan foydalanishga, yangi bo'limlar kiritishga yoki ularni birlashtirishga ruxsat etiladi.

Texnik topshiriqda nazarda tutilgan qandaydir talablarni buyurtmachi qo'ymagan holatlarda tegishli joyda «Talablar qo'yilmadi» deb ko'rsatish lozim.

Texnik topshiriqni ishlab chiqish

Texnik topshiriqni ishlab chiqish — muayyan ko'nikmalarni talab qiluvchi sermehnat jarayon qoidaga ko'ra, asosiy bo'limlarni: kirish, dasturiy mahsulot ahamiyati va dasturiy mahsulotga talablarni aniq shakllantirish eng murakkabdir. Misol tariqasida qisqartirilgan sxema bo'yicha tuzilgan kurs loyihalashini bajarishga ikkita texnik topshiriqni hamda davlat budjetidagi ilmiytadqiqot ishini bajarishga nisbatan to'liq texnik topshiriqni ko'rib chiqishimiz mumkin.

Loyihalashtirish boshlang'ich bosqichlarining prinsipial yechimlari

Loyihalashtirishning boshlang'ich bosqichlarida ushbu jarayonni, ishlanma sifati va murakkabligini belgilovchi muhim qarorlar qabul qilinishi lozim. Bu quyidagi qarorlardir:

- dasturiy ta'minot arxitekturasini tanlash;
- foydalanuvchining interfeys turini va hujjatlar bilan ishlash texnologiyasini tanlash;

- ishlanmaga tuzilmaviy va obyektiv yondashuvni tanlash;
- dasturlash tili va muhitini tanlash;

Boshqacha aytganda, bu qarorlar qanday iste'mol tavsifi va qanday vositalar bilan loyihalashtirilishini belgilab beradi.

Qarorlarning bir qismi texnik vazifalarda, texnologik talablar guruhini tashkil etgan holda belgilanishi, qolganlari esa loyihalashtirish jarayoni uchun boshlang'ich ma'lumotlar bo'lgani sababli oldinroq qabul qilinishi lozim.

Dasturiy ta'minot arxitekturasi

Dasturiy ta'minot arxitekturasini tanlash. *Dasturiy ta'minot arxitekturasi* deb, uning tuzilishining bazaviy konsepsiyalari rna'jmuiga aytiladi. U yengiladigan vazifalar murakkabligi, ishlab chiqiladigan dasturiy ta'minot universalligi darajasi va uning bir nusxasi bilan ishlovchi foydalanuvchilar soni bilan belgilanadi va quyidagilarni farqlaydi:

- bitta foydalanuvchi arxitekturasini.

Bunda dasturiy ta'minot personal kompyuterda ishlovchi bitta foydalanuvchiga mo'ljallangan:

- lokal yoki global tarmoqda ishlashga mo'ljallangan ko'p foydalanuvchi arxitekturasi.

Bundan tashqari bitta foydalanuvchi arxitekturasi quyidagilarni farqlaydi:

- dasturlar;
- dasturlar paketi;
- dasturiy komplekslar;
- dasturiy tizimlar.

Ko'p foydalanuvchili arxitektura <<mijoz-server>> tamoyili bo'yicha tuzilgan tizimlarni amalga oshiradi.

“Dastur”

Dastur deb aniq bir masalani yechish uchun bajarilish ketmaketligi kompyuterda aniq va ravshan bayon etilgan instruksiya (yo'riqnoma)lar to'plamiga aytiladi.

Dastur tuzilmali (strukturali) yondashuvda aniq bir masalani yechish jarayonida bir-birini chaqiruvchi dastur ostilari (tag, dastur (dastur qismi)lari) iyerarxiyasini namoyon etadi, obyektli yondashuvda esa, realizatsiya uchun maxsus yaratilgan sinflardan iborat obyektlarning xabarlar bilan almashinuvi majmuasidan iborat.

Dastur bunday holatda alohida jamlanuvchi dasturiy birlikni namoyon etadi. U standart kutubxona dastur ostisidan foydalanishi mumkin, ammo o'zini tashkil etmaydi. Bu eng sodda arxitektura turi bo'lib, odatda kichik vazifalarni yechishda foydalaniladi.

Dasturlar paketi

Dasturlar paketi ayrim amaliy sohada vazifalarni hal etuvchi dasturlar majmuidan iborat. Masalan, grafik dasturlar paketi, matematik dasturlar paketi. Bunday paketlar dasturi bir-biri bilan ma'lum bir amaliy sohaga mansubligi bilan bog'liq bo'ladi. Dasturlar paketi alohida dasturlar to'plami sifatida amalga oshiriladi.

Ularning har biri kerakli ma'lumotlarni kiritadi va natijalarni chiqaradi. Mohiyatan, dasturlar paketi bu dasturning o'ziga xos kutubxonasidir.

Dasturiy komplekslar

Dasturiy komplekslar dasturlar majmuidan iborat bo'lib, birgalikda biror amaliy sohada murakkab vazifalarning kichikroq sinfini hal etishni ta'minlaydi. Bunday masalalarni hal etish uchun ketma-ket komplekslar dasturini chaqirgan hamda bir nechta vazipredmetini bajarishga to'g'ri kelishi mumkin. Dasturiy kompleksda dasturni chaqirish maxsus dastur — dispatcher orqali amalga oshiriladi. U foydalanuvchi bilan murakkab bo'lmagan interfeys orqali bog'lanadi va ayrim ma'lumotnomalarni taqdim etishi mumkin. Dasturiy kompleks dasturlar paketidan yana shunisi bilan farqlanadiki, bir nechta dastur navbati bilan yoki davriy shaklda bitta vazipredmeti hal etish uchun chaqirilishi mumkin.

Dasturiy tizimlar

Dasturiy tizimlar ayrim amaliy sohada keng ko'lamdagi vazifalarni hal etish imkonini beruvchi dasturlar (tizim osti)ning uyushgan majmuini ifodalaydi. Dasturiy komplekslardan farqli ravishda dasturiy tizimga kiruvchi dasturlar umumiy ma'lumotlar orqali o'zaro harakat qiladi. Dasturiy tizimlar odatda, rivojlangan foydalanuvchi va ichki interfeysga ega bo'ladi. Bu ularni puxta loyihalashni talab etadi.

Ko'p foydalanuvchi dasturlar tizim

Ko'p foydalanuvchi dasturlar tizim odatdagi dasturiy tizimlardan farqli ravishda dasturiy ta'minotning alohida komponentlari tarmoq ta'sirini tashkil etishi lozim. Bu uni ishlab chiqish jarayonini yanada murakkablashtiradi. Bunday dasturiy ta'minotni ishlab chiqish uchun maxsus texnologiya yoki CORBA, COM, Java va hokazo texnologiyalar platformasidan foydalaniladi. Foydalanuvchi interfeysi turini tanlash. Foydalanuvchi interfeysining to'rtta turi farqlanadi. Bular:

- *primitiv (sodda)* — yagona ish ssenariysini amalga oshiradi, masalan, ma'lumotlarni kiritish — natijalarni chiqarish;
- *menyu* — ko'plab ish ssenariylarini amalga oshiradi. Ularning operatsiyalari iyerarxik strukturada tashkil etilgan. Masalan, <<qo'yish>>, <<faylni qo'yish>>, <<simvolni qo'yish>> va hokazo.
- *erkin navigatsiyali* — ko'plab ssenariylarni amalga oshiradi. Ularning operatsiyalari iyerarxiya darajalariga bog'liq va ma'lum bir ish bosqichida ko'plab operatsiyalarni belgilab beradi. Bunday shakl interfeyslari asosan Windows-illovadan foydalanadi;
- *bevosita manipulyatsiyalash* — obyektlar ustidagi operatsiyalarda taqdim etilgan ko'plab ssenariylarni amalga oshiradi.

<<Sichqon>> obyektleri piktogrammasi

Asosiy operatsiyalar <<sichqon>> obyektleri piktogrammasi ko'chishi bilan amalga

oshriladi. Bu shakl Windows operatsiya tizimining interfeysida, erkin navigatsiyali interfeysga muqobil ravishda amalga oshirilgan. Foydalanuvchi interfeys turi ishlanmaning murakkabligi va ko'p mehnat talab etishini belgilab beradi. Oxirgi ma'lumotlarga ko'ra, dasturiy kodning 80 foizigacha aynan foydalanuvchining interfeysi amalga oshirilishi mumkin. Shu sababli dasturlashni o'rganishning boshlang'ich bosqichida, asosan, sodda interfeys va menyulardan foydalaniladi, garchi ular foydalanuvchi uchun noqulay bo'lsa ham. Dasturiy vositalarni ishlab chiqishning obyektga yo'naltirilgan vizual muhitini paydo bo'lishi muhim ahamiyatga ega bo'ladi. Bunday yo'l dasturlashda hodisali yondashuvga asoslangan va erkin navigatsiyali interfeyslarni yaratishga mo'ljallangan bo'lib, monand interfeyslarni yaratish qiyinchiliklarini yetarlicha kamaytirdi hamda to'g'ridan to'g'ri manipulyatsiya qilish interfeyslarini realizatsiyasini soddalashtirdi. Shunday qilib, oxirgi ikkita interfeys turini tanlash dasturiy ta'minotni ishlab chiqishning vizual muhitlaridan biridan foydalanishni nazarda tutadi.

Interfeys

Interfeys turini tanlash o'z ichiga hujjatlar bilan ishlash texnologiyasini tanlashni oladi. Ikkita texnologiya farqlanadi:

- bitta hujjatli interfeys (SDI — Single Document Interface) ni mo'ljallaydi;
- ko'p hujjatli — ko'p hujjatli interfeysga (MDI — Multiple Document Interface) mo'ljallangan.

Ko'p hujjatli texnologiya agar, dasturiy ta'minot bir vaqtning o'zida bir necha hujjatlar bilan ishlashi shart bo'lsa (masalan, bir necha matn yoki tasvir bilan) foydalaniladi. Bitta hujjatli texnologiyada esa bir vaqtning o'zida bir necha hujjatlar bilan ishlash shart emas. Zamonaviy kutubxonadan foydalangan holda ko'p hujjatli interfeysni amalga oshirish ish hajmi birinchisiga qaraganda 3—5% ko'p. Interfeys turini tanlash ish hajmiga jiddiy ta'sir ko'rsatadi. Ishlanmaga yondashuvni tanlash. Agar erkin navigatsiyali yoki bevosita manipulyatsiyalovchi interfeys tanlangan bo'lsa, yuqorida qayd etilganidek, bu vaziyatga ko'ra dasturlash va obyektiv yondashuvdan foydalanishni nazarda tutadi. Chunki Visual C++, Delphi, Bulder C++ kabi zamonaviy vizual dasturlash muhiti aynan kutubxona sinfi obyektlari ko'rinishidagi interfeys komponentlarini taqdim etadi. Bunda predmet sohaning murakkabligiga bog'liq ravishda dasturiy ta'minot obyektlar (mos ravishda sinflar) dan foydalanilgan holda yoki faqat protsedurali holdan foydalanib realizatsiya qilinishi mumkin. Istisno tariqasida Perl ga o'xshash Internet-qo'llanmani maxsus tillarini yaratishning natijasini ko'rsatish mumkin. Bu til umuman olganda boshqacha tamoyilga asosan qurilgan.

Sodda interfeys

Sodda interfeys va menyuga o'xshash interfeys ham strukturali, ham ishlanmaga obyektli yondashuvga muvofiq keladi. Shu bois yondashuvni tanlash qo'shimcha axborotdan foydalanish orqali amalga oshiriladi.

Tajriba shuni ko'rsatadiki, obyektiv yondashuv juda katta dasturiy tizimni (dasturlash universal tilining 100 000 dan ortiq operatori) ishlab chiqishda hamda predmet sohasining obyekt strukturasi yorqin ifodalanganda samaralidir. Shuni ham hisobga olish kerakki, ishlab chiqiladigan dasturiy ta'minot samarasi qat'iy cheklanganda obyektli yondashuvdan ehtiyotkorona yondashish kerak. Boshqa hamma holatlarda yondashuvni tanlash foydalanuvchi ixtiyoriga bog'liq.

Dasturlash tilini tanlash

Ko'pgina hollarda dasturlash tilini tanlash yuzasidan hech bir muammo mavjud emas. Til quyidagilar tomonidan belgilanishi mumkin:

- ishlanmani olib boruvchi tashkilot tomonidan. Masalan, agar firma C++Builder litsenziya variantiga ega bo'lsa, u ko'proq mazkur muhitda ishlanmani olib boradi;
- har doim imkoniyatga qarab, yaxshi tildan foydalanuvchi dasturchi tomonidan;
- qat'iy fikr asosida («barcha bunday ishlanmalar C++ da yoki Java yoki ... da bajarilishi kerak») va hokazo.

Agar, til tanlash imkoni bo'lsa dasturlashning barcha mavjud tillarini quyidagi guruhlarga bo'lish mumkinligini hisobga olish kerak:

- yuqori darajadagi universal til;
- dasturiy ta'minotni ishlab chiquvchining ixtisoslashgan tillari;
- foydalanuvchining ixtisoslashgan tillari;
- quyi darajadagi tillar.

Universal tillar

Universal tillarning yuqori darajasi guruhida so'zsiz ravishda bugungi kunda C (C++ bilan birga) dasturlash tili yetakchi sanaladi. Darhaqiqat C va C++ning turli versiyalari juda muhim afzalliklarga ega:

- ko'p platformali - bugungi kunda barcha foydalanuvchilar uchun C va C++ tilli kompilyatorlar mavjud;
- asosiy tuzilmaviy algoritimli konstruksiyalarni (shartli qayta ishlash, sikllarning barcha turi) amalga oshiruvchi operatorlar mavjudligi;
- tezkor xotira manzillaridan foydalangan holda quyi (tizimli) darajada dasturlash imkoniyatini;
- dastur osti va sinflarga oid katta kutubxonalar.

C va C++ ning operatsiya tizimi

Bular bari C va C++ ning operatsiya tizimini yaratish uchun foydalaniladigan asosiy tillarga aylantirdi. Biroq C va C++ jiddiy kamchiliklarga ega:

- ma'lumotlarni to'laqonli ichki tuzilmali (strukturali) turlarining yo'qligi (adresli (manzilli) arifmetikada foydalanadigan psevdotuzilmasi (psevdostrukturali) ma'lumotlarning mavjudligi bu ma'lumotlar ustida ko'p amallarni nazorat qilish uchun yetarli darajada qat'iy aniqlangan. Bu esa o'z navbatida faqat dasturni rostdash jarayonida ko'p sondagi xatoliklarni aniqlashga olib keladi);
- sintaktik jihatdan bir qiymatlilik emasligini (turli xillikning) mavjudligi. Bu ham kompilyatorga dastur to'g'riligini nazorat qilish imkonini bermaydi;

- tag dastur (dastur qismi)ga uzatiladigan parametrlarning cheklangan nazorati. Bu faqat dasturni yopish jarayonida aniqlanadi va hokazo.

Pascal

Amaliy dasturlash ta'minotini yaratish uchun foydalaniladigan universal dasturlash tillari o'rtasida C va C++ ga muqobil bu Pascal sanaladi. Uning sintaksisi aniq bo'lgan bo'is kompilyatorlari sintaksisdan tashqari ko'plab semantik xatolarni ham aniqlaydi. Delphi muhitida foydalangan Object Pascal versiyasi professional kutubxona yordamida kuzatib boriladi. Bu katta ishlanmalarni kiritishni osonlashtiradi va Windows ilovasini yaratish uchun Delphini yetarlicha samarali muhit bilan ta'minlaydi. Bu tillardan tashqari universal guruhga Basic, Modula, Ada va ayrim tillar kiradi. Ular ham C++ va Pascal singari o'zining qo'llanish sohasiga hamda xususiyatlariga ega.

Ixtisoslashgan tillar

Ishlanma yaratuvchining ixtisoslashgan tillaridan dasturiy ta'minotning konkret turlarini yaratish uchun foydalaniladi.

Ularga quyidagilar kiradi:

- ma'lumotlar bazalari tillari;
- tarmoq ilovalarni yaratish tillari;
- sun'iy intellekt tizimini yaratish tillari va h.k.

Bu tillar maxsus kurslarda o'rganiladi va mazkur darslikda ko'rib chiqilmaydi. Foydalanuvchining ixtisoslashgan tillari odatda foydalanuvchi professional muhitining bir qismi sanaladi. U tor yo'nalishli bo'lib, dasturiy ta'minotni ishlab chiquvchi tomonidan foydalanilmaydi.

Quyi darajadagi tillar dasturlashni amalda mashina komandalari darajasida amalga oshirish imkonini beradi. Bunda ham bajarish vaqti nuqtayi nazaridan, ham dasturning zarur xotirasi hajmi nuqtayi nazaridan eng maqbul yo'l tanlanadi. Biroq bu tillar kata dasturlarni yaratish uchun yaramaydi. Quyi darajadagi tillar strukturali dasturlash tamoyillariga ega emas. Bu ishlab chiqiladigan dastur texnologiyasini yomonlashtiradi.

Assembler tipidagi tillar

Hozirda Assembler tipidagi tillardan quyidagi holatlarda foydalaniladi:

- texnik vositalar bilan (masalan, drayver) bevosita o'zaro ta'sir ko'rsatuvchi oddiy dasturlarni yozishda, chunki bu holatda tegishli jihozni ter to'kib sozlash kerak bo'ladi.

Yuqori darajadagi dasturlash tillar ustunligi muhim bo'lmaydi:

- yuqori darajadagi tillarda dasturga qo'shimcha vosita shaklida, masalan, katta miqdordagi takrorlash bilan sikllarda ma'lumotlar hosil bo'lishini tezlashtirish uchun.

Dasturlash muhitini tanlash

Dasturlash muhiti deb dasturiy kompleksga aytiladi. U o'z ichiga ixtisoslashgan matnli muharrir, kompilyator, komponovlovchi, ma'lumotnoma tizimi va boshqa dasturlashlarni oladi.

So'nggi vaqtlarda yuqorida qayd etilgan vizual dasturlash keng tarqaldi. Bunga ko'ra dasturchi komponentlarning maxsus kutubxonalaridagi ayrim kodlar dasturiga vizual bog'lanislriskoniga ega bo'ladi.

Borland firmasining (Inprige Corparation) Delphi, C++ Builder, Microsoft firmasining Visual C++, Visual Basic va IBM firmasining Visual Ada vizual muhitlaridan nisbatan ko'p foydalanilmoqda.

Ushbu firmalarning Delphi, C++ Builder va Visual C++ asosiy vizual muhitlari o'rtasida jiddiy farq mavjud. Microsoft firmasining vizual muhitlari «Windows» ostida dasturlashning past darajasini ta'minlaydi. Bu ularning ham yutug'i, ham kamchiligi sanaladi. Yutug'i shundan iboratki, u «nostandart» vaziyatlar yuzaga kelish ehtimolini kamaytiradi. Kamchiligi esa bu Delphi yoki C++ bilan ishlovchi dasturchini eskicha ishga undaydi.

«Quyi darajali dasturlash»

Shuningdek, «quyi darajali dasturlash» ga mo'ljallangan Visual C++ interfeysi ham ko'plab norozilikka sabab bo'ladi. Umuman olganda, agar gap ushbu muhitlardan birini tanlash ustida ketsa, u ko'proq loyiha tavsifi bilan belgilanadi. Ishlanma standartlarini tanlash yoki shakllantirish. Har qanday loyihalash texnologiyasini qo'llash barcha loyiha ishtirokchilari rioya etishi shart bo'lgan qator standartlar shakllantirish yoki tanlashni talab etadi.

- loyihalashtirish standarti;
- loyiha hujjatlashtirishini rasmiylashtirish standarti;
- foydalanuvchi interfeysi standarti.

Loyihalashtirish standarti

Loyihalashtirish standarti quyidagilarni aniqlashi lozim:

- loyihalashning har bir bosqichida zarur modellarni (sxema, diagramma) tanlash va ularni detallashtirish darajasi;
- diagrammalardagi loyiha qarorlarini qayd etish qoidasini, shuningdek, obyektlar va atamalar bo'yicha kelishuvni nomlash, barcha obyektlar uchun atributlar to'plami va har bir bosqichda ularni to'ldirish qoidasi, diagrammani rasmiylashtirish qoidasi, jumladan obyektlar shakli va o'lchamiga bo'lgan talablar;
- ishlanma muammolarining ish o'rni konfiguratsiyasiga bo'lgan talablar jumladan operatsiya tizimi va foydalanayotgan

CASE vositasini sozlash

- loyiha ustida hamkorlikda ishlashni ta'minlash mexanizmi, jumladan, loyiha tizim ostini integratsiyalash qoidasi hamda loyiha qarorlari tahlili.

Loyiha hujjatlarini rasmiylashtirish standarti quyidagilarni rejalashtirishi lozim:

- har bir bosqichda hujjatlashtirish jamlanmasi, tarkibi va strukturasi;

- uni saqlab turish va rasmiylashtirishga nisbatan talablar;
- hujjatlarni tayyorlash, kocrib chiqish, kelishish va tasdiqlash qoidalari.

Foydalanuvchi interfeysining standarti

Foydalanuvchi interfeysining standarti quyidagilarni aniqlashi lozim:

- ekranlarni rasmiylashtirish qoidasi (shriftlar va rang palitrasi), derazalar va boshqaruv elementi tarkibi hamda joylashuvi;
- klaviatura va <<sichqon>>dan foydalanish qoidasi;
- yordam ko'rsatish matnlarini rasmiylashtirish qoidasi;
- standart xabarlar ro'yxati;
- foydalanuvchi reaksiyasini qayta ishlash qoidasi.

Yuqorida bayon etilgan loyiha qarorlari ishlanmaning ish hajmi va murakkabligiga ta'sir etadi. Faqat ularni qabul qilgandan so'ng loyihalashtirishning dasturiy ta'minotiga o'tish zarur.

Foydalanilgan adabiyotlar

Ramez Elmasri, Shamkant B.Navathe. Fundamentals of Database Systems (7th Edition). Pearson. USA, 2015.

1. M.T.Azimjanova, Muradova, M.Pazilova. Informatika va axborot texnologiyalari. O'quv qo'llanma. T.: "O'zbekiston faylasuflari milliy jamiyati", 2013 y.
2. Holmatov T.X., N.I.Tayloqov. Amaliy matematika, dasturlash va kompyuteming dasturiy ta'minoti. O'quv qo'llanma. T.: "Mehnat", 2000 y.
3. S.S.Qosimov Axborot texnologiyalar O'quv qo'llanma. T.: "Aloqachi", 2006 y.
4. M Aripov, B.Begalov va boshqalar. Axborot texnologiyalari. O'quv qo'llanma. T.: "Noshir", 2009 y.
5. A.Sattorov. Ma'lumotlar bazasini boshqarish sistemasi Access (Windows 9x/2006) O'quv qo'llanma. T.: "Fan va texnologiya", 2006 y.
6. Axborot xavfsizligi asoslari, I.M.Karimov Tashkent, 2019
7. Информационная безопасность телекоммуникационных систем (технические аспекты), Кулаков М.В., Гарашин А.В. Учеб. Пособие для вузов. М.: Радио и связь, 2014
8. Информационная безопасность государственных организаций и коммерческих фирм. Справочное пособие. М.: 2017.
9. Зегжда Д.П. Основы безопасности информационных систем: Учеб. пособие для студ. вузов. – М.: Горячая линия – Телеком, 2016.