

**O'ZBEKISTON RESPUBLIKASI AXBOROT TEXNOLOGIYALARI VA
KOMMUNIKATSIYALARINI RIVOJLANTIRISH VAZIRLIGI
MUHAMMAD AL-XORAZMIY NOMIDAGI TOSHKENT AXBOROT
TEXNOLOGIYALARI UNIVERSITETI SAMARQAND FILIALI
KOMPYUTER INJINIRINGI FAKULTETI
AXBOROT TEXNOLOGIYALARI KAFEDRASI**

5330500 – Kompyuter injiniringi (Kompyuter injiniringi, AT-Servisi)

Himoyaga ruxsat

Kafedra mudiri: dots.Maxmudov Z.M.

_____ «___» _____ 2018 yil

Shifoxonalarda bemorlarni ro'yxatga olishni avtomatlashtiruvchi dasturiy vosita
yaratish
mavzusida

BITIRUV MALAKAVIY ISHI

Bitiruvchi: _____ *406-14 KI –gr.Husaynov N.*

Rahbar: _____ *Ass.Raximov R.T.*

Taqrizchi: _____ *Ass.Qarshiyev H.*

HFX maslahatchi: _____ *Kat.o'q. Kurbanniyazov A.S.*

SAMARQAND – 2018

MUNDARIJA

KIRISH.....	3
I Bob. RO'YXATGA OLISHNI AVTOMATLASHTIRISHNING BA'ZI USULLARI	6
1.1. Ruyxatga olish uchun kerakli ma'lumotlar	6
1.2. Avtomatlashtirilgan axborot tizimlari	9
II Bob. SHIFOXONAGA OID MA'LUMOTLAR BAZASINI YARATISH BOSQICHLARI.....	13
2.1. Ma'lumotlarni tashkil qilish.....	13
2.2. Ma'lumotlarni mantiqiy tashkil etish.....	13
2.3. Fayllar modeli.....	14
2.4. Ma'lumotlarni tashkil qilishning tarmoqli va iyerarxik modellari.....	16
2.5. C++ Builder dasturida ilovalar bilan ishlash	16
2.6. Masalaning algoritmi.....	33
III Bob. DASTURIY TA'MINOT VA UNDAN FOYDALANISH.....	35
3.1. Dasturiy ta'minotni ishlatish va natija olish.....	35
3.2. Shifoxona xodimlarini favqulodda vaziyat sharoitida himoyalash.....	37
XULOSA.....	48
FOYDALANILGAN ADABIYOTLAR RO'YXATI.....	49
ILOVA.....	51

KIRISH

Mavzuning dolzarbligi. O'zbekiston Respublikasi Prezidentining 2002 yil 30 maydagi «Kompyuterlashtirishning yanada rivojlantirish va axborot-kommunikatsiya texnologiyalarini joriy etish to'g'risida»gi farmoni va uni bajarilishini ta'minlash yo'lida Vazirlar Mahkamasining 2002 yil 6 iyundagi maxsus qarori hamda Vazirlar mahkamasining «Joylarda kompyuterlashtirish va axborot kommunikatsiya texnologiyalarini yanada rivojlantirish uchun shart-sharoitlar yaratish chora-tadbirlari to'g'risida»gi qarorlari Respublikamizda axborot texnologiyalarini rivojlantirishda yana bir muhim bosqich bo'lib qoladi[1].

Kompyuterlashtirishni yanada rivojlantirish yo'lida hozirgi kunda ko'plab ilmiy-amaliy ishlar amalga oshirilmoqda. Jumladan bir qator oliy ta'lim dargohlarida axborot texnologiyalari sohasida ko'plab ilmiy tadqiqot ishlari olib borilmoqda va ularning samaralari yaqqol ko'zga tashlanmoqda.

«Jahon sivilizatsiyasiga daxldor bo'lgan eng zamonaviy ilmlarni egallamay turib, mamlakat taraqqiyotini ta'minlash qiyin», - degan edilar birinchi prezidentimiz I.A.Karimov. O'zbekistonning iqtisodiy va ijtimoiy sohalarda yuqori natijalarga erishishi, jahon iqtisodiy tizimida to'laqonli sheriklik o'rnini egallay borishi, inson faoliyatining barcha jabhalarida zamonaviy axborot texnologiyalaridan yuqori darajada foydalanishning ko'lamlari qanday bo'lishiga hamda bu texnologiyalar ijtimoiy mehnat samaradorligini oshishida qanday rol o'ynashiga bog'liq[2].

Birinchi Prezidentimiz Islom Karimovning asarlaridagi fikr-mulohazalariga tayanib yaratilgan «Milliy istiqlol g'oyasi: asosiy tushuncha va tamoyillar» nomli risolada ta'lim va tarbiya jarayonining ixtiyoriy bosqichlarida amal qilish lozim bo'lgan quyidagi mezonlar va talablar keltirilgan:

- o'quv mashg'ulotlarini olib borishda talabalarning yoshi, tafakkuri, dunyoqarashi va qiziqishlarini hisobga olish;
- ta'lim-tarbiyaning ilg'or, ta'sirchan vositalaridan, zamonaviy o'qitish texnologiyasi imkoniyatlaridan keng foydalanish;
- ayrim tushunchalarni haddan ziyod soddalashtirish, ta'limning eskicha uslub va tamoyillarini qo'llash natijasida fanning qadrsizlanishiga yo'l qo'ymaslik;
- ta'lim jarayonida tazyiq o'tkazmasdan ma'rifiy asosda ish tutish, yoshlarning mustaqil va erkin fikrlash, bahs-munozara yuritish ko'nikmalarini oshirishga e'tibor qaratish;
- o'qituvchi va tinglovchilar orasida o'zaro hamfikrlik va hamkorlik muhitini shakllantirish, mavzuning tushuncha va tamoyillarini sharhlashda hayotiy misollar, bugungi dunyoda ro'y berayotgan voqyealar tahlilidan, matbuot materiallaridan keng foydalanish;
- yoshlarda g'oyalar, o'z ma'no-mohiyatiga ko'ra bunyodkor yoki vayronkor bo'lishi haqidagi hayotiy va haqqoniy tasavvurlarni shakllantirish;
- milliy istiqlol g'oyasining insonparvarlik mohiyatini ko'rsatish asosida mustaqillik biz uchun eng oliy qadriyat, uni asrab-avaylash esa har birimizning muqaddas burchimiz ekanini talabalarning qalbi va ongiga singdirish.

Yuqorida aytilgan mezon va talablarga rioya qilgan holda Respublikamizda, zamonaviy hisoblash texnikasi vositalaridan samarali foydalanishni uddalay oladigan, zamonaviy kompyuterlardan amaliy ish faoliyatida keng foydalana oladigan yetuk kadrlar tayyorlash dolzarb vazifalardan hisoblanadi. Shuning uchun, kadrlar tayyorlash milliy dasturining ikkinchi bosqichida yuqori malaka, raqobatbardosh kadrlar tayyorlash uchun sifatli, jahon andozalariga mos darsliklar, o'quv qo'llanmalari va ma'ruza matnlarini tayyorlab, chop ettirish masalasiga juda katta e'tibor berilgan[4].

Ma'lumki, bugungi kunda, ya'ni jahon moliyaviy – iqtisodiy inqirozi sharoitida iqtisodiy rivojlanishning innovatsion yo'li juda dolzarb hisoblanadi. Bu borada birinchi Prezidentimizning “Jahon moliyaviy – iqtisodiy inqirozi, O'zbekiston sharoitida uni bartaraf etishning yo'llari va choralari” kitobida ham quyidagi fikrlari keltirilgan: “... innovasion texnologiyalarni keng joriy etish - O'zbekiston uchun inqirozni bartaraf etish va jahon bozorida yangi marralarga chiqishning ishonchli yo'lidir”[3].

Ushbu bitiruv malakaviy ishida shifoxonalarda bemorlarni ro'yxatga olishni avtomatlashtiruvchi dasturiy vosita yaratishga qaratilgan.

Bitiruv Malakaviy ishi kirish, masalaning qo'yilishi, uchta bob, xulosa, foydalanilgan adabiyotlar ro'yxati va ilovadan iborat. BMI mavzusining asosiy qismi birinchi 3 ta bobda bayon qilingan. Hayot faoliyati xavfsizligini ta'minlash chora-tadbirlari uchinchi bobda qarab chiqilgan. Yaratilgan dasturiy vositaning matni ilova qilinadi.

Shifoxonalarda bemorlarni ro'yxatga olishni avtomatlashtiruvchi dasturiy vosita yaratish va ulai amaliy jihatdan muhim ishlarni amalga oshirishda qo'llash shifoxona xodimlari uchun dolzarb masalalaridan biri bo'lib hisoblanadi.

I BOB. RO'YXATGA OLISHNI AVTOMATLASHTIRISHNING BA'ZI USULLARI

1.1. Ruyxatga olish uchun kerakli ma'lumotlar

Kasalxona - bemorlarni davolash va (yoki) kasalliklarni asosan statsionar sharoitlarda chuqur differentsial tashhislashga qaratilgan majmua hisoblanadi. Hozirgi vaqtda bemorlar bemorlarning kartalarini ro'yxatdan o'tkazib, har bir qabul qilingan holat bo'yicha ishni olib borish orqali kasalxonada ro'yxatga olinadi. Kasallik tarixida barcha kuzatuvlar, protseduralar, tahlillar va boshqalar yoziladi. Bemorni kasalxonada qolish vaqtida (agar u doimiy ravishda davolansa) yoki tashrif buyurishi (ambulatoriya bilan davolansa). Shubhasiz, shifoxonada ishlaydigan ma'lumotlarning soni juda ko'p bo'ladi. Hozirgi kunda shifoxonalarda bemorlarni davolash bo'yicha barcha ma'lumotlar va ro'yxatga olish arxivlarda qog'oz shaklida saqlanadi. Elektron shaklda faqatgina bemor haqida ma'lumot (ism, yashash joyi, yoshi, va hokazo), shifokorlar, bo'limlar va buyurtma kuponlari to'g'risida ma'lumot taqdim etiladi. Kasalxonalardagi bemorlar hisobini avtomatlashtirishga imkon beruvchi dasturiy vositani yaratish, ya'ni ularning pasport ma'lumotlarini emas, balki ularning kelib tushishi, shikoyati, kasallikning rivojlanishi va davolanish natijalari haqida ma'lumot olishni ta'minlayd i, ma'lumotni izlashni, qog'ozni, bo'sh joyni, ish vaqtini tejashni ancha osonlashtirad i. Bu yong'in natijasida ma'lumotlar yo'qolishiga oldini olishi mumkin, masalan, ma'lumotlar bazasining zaxira nusxalarini yaratish, axborotdan foydalanishni cheklash va h.k.

1.2. Avtomatlashtirilgan axborot tizimlari

Avtomatlashtirilgan axborot tizimini ishlab chiqishdan maqsad — tashkiliy loyixalashtirish, texnologik va hokazo jihatlarini hisobga olgan holda tizim faoliyatining samaradorligini oshirishdir.

Umumiy holatda tizim, deganda ular orasidagi va ularning hususiyatlari o'rtasidagi aloqalar majmuiga ega bo'lgan, ya'ni bir-

biriga chambarchas bog'langan qismlardan iborat butun bir ob'ektlar majmuasi tushuniladi. Bunday ta'rifdagi tizimga quyidagilarni misol qilib keltirish mumkin: detallar va tutashtiruvchi kurilmalardan yig'ilgan mashina hujayralarning butun majmuini tashkil etuvchi tirik organizm; turli resurslar, bir-biri bilan bog'langan ko'plab ishlab chiqarish jarayonlari va kishilar jamoalari yahlitligida yuzaga kelgan korxonalar va xokazo. Bunday hollarda ob'ektlar yagona tizim sifatida ishlaydi, ya'ni har bir ob'ekt, kenja tizimlar umumiy tizim oldidagi yagona maqsad uchun harakat qiladi.

Tizimni aniqlashga quyidagi atamalar kiradi: «Ob'ektlar», «Aloqalar», «Xususiyatlar».

Ob'ektlar — tizimning bir bo'lagi yoki komponentlari bo'lib, jismoniy, matematik o'zgaruvchan tenglamalar, qoida va qonunlar, texnologik jarayonlar, axborot jarayonlari, ishlab chiqarish bo'linmalari kabi ko'plab cheklanmagan qismlarga ega.

Xususiyatlar — bu ob'ektning sifatini ifodalovchi parametrlardir. Xususiyat tizimning ma'lum bir o'lchamga ega ob'ektlarini belgilab miqdoriy jihatdan bayon etish imkonini beradi.

Ob'ektlarning xususiyatlari tizim harakati natijasida o'zgarishi mumkin.

Aloqalir ob'ektlar va ularning xususiyatlarini tizim jarayonida yagona yaxlitlikka birlashtiradi. Bunda barcha tizim elementlarining kenja tizimlari va tizimlar o'rtasida aloqa bo'lishi nazarda tutiladi. Ayrim umumiy qonuniyatlar, qoidalar yoki tamoyillar bilan birlashuvchilar o'rtasida aloqaning mavjud bo'lishi tizimning asosiy tushunchasi sanaladi. Boshqalar bilan biror-bir aloqaga ega bo'lmagan element ko'rib chiqilayotgan tizimga kirmaydi. Tizimning hususiyatlari quyidagilar sanaladi: elementlar murakkabligi, maqsadga

qaratilganligi, turli - tumanligi hamda ular tabiati, tarkiblashganligi, bo'linishligidir.

Tizimlar tarkibi hamda asosiy maqsadlariga farqlanadi. Tashkiliy murakkablik tizimning asosiy hususiyati sanaladi va u elementlar o'rtasidagi o'zaro aloqalar (o'zaro harakatlar) miqdori bilan aniqlanadi. Elementlar o'rtasidagi chatishib, ko'shilib ketgan o'zaro aloqalar shunday tuzilganki, u birorta parametr aloqasining o'zgarishiga olib keladi.

Tashkiliy murakkablik elementlar tizimini tashkil etuvchi tavsiflar miqdori bo'lmagan, yahlit holda, faqat tizimga tegishli tavsiflarni aniqlaydi. Umuman olganda, tizim uni tashkil etuvchi elementlardan boshqacharoq tavsiflarga ega bo'libgina qolmay, balki uning barcha qismlaridan sifat jihatidan farqlanadi. Shuningdek elementlar ega bo'lmagan boshi vazifalarni ham bajarish hususiyatiga ega.

Tizim butunligining o'ziga xosligi bilan aniqlanadigan yangi xususiyatlarning paydo bo'lishi ba'zan emerjentlik (inglizcha «Emergent» — yuzaga keluvchi, paydo bo'luvchi) deb ataladi. Tizimlarni qismlarga, ayniqsa o'zi tarkib topadigan elementlarga bo'lganda bunday vazifalar yoki tavsiflar o'z-o'zidan yo'q bo'aladi.

Maqsadga karatilganlik. Tizim umumiy hususiyatga ega, ya'ni u umumiy maqsadga erishishga harakat qilishga qaratilgan. Tizimning maqsadga yo'naltirganligini ifodalovchi barcha elementlar uchun umumiy bo'lgan o'zaro aloqalarning maqsadli qoidalari maqsadning mavjudligini belgilaydi.

Tizimning tarkiblashganligi — bu tizimning alohida elementlari va ularning tashqi muhit bilan o'zaro harakati o'rtasidagi ichki aloqalarning doimiy tarkibidir. Tizim tarkibi — uning faoliyati

samaradorligini ko'p jihatdan belgilovchi muhim tavsiflardan biri sanaladi.

Tizimning bo'linishi — bu uning maqsadlar va vazifalarga javob beruvchi ma'lum belgilar bo'yicha ajratilgan elementlar yoki bir qator kenja tizimlardan tuzilganligini anglatadi. Kenja tizimlar bunday ajratilishning asosini tashkil etib, bunda elementlar o'rtasidagi aloqalar ko'proq, kenja tizimlar o'rtasida esa kamroq bo'ladi.

Tizim tushunchasi shu ma'noda nisbiyki, tizim elementining o'zi ham murakkab tizim bo'lishi mumkin. Biror belgi bo'yicha ajratilgan tizim o'ziga nisbatan yuqori darajadagi tizim elementi bo'lishi mumkin.

Tashqi muhit. “Tizim” tushunchasi tizimga kiruvchi bir qator elementlarni cheklaydi: shartli ravishda cheklangan chegara o'rganadi, undan tashqaridagi elementlar esa ushbu tizimga kirmay qoladi. Bundan anglashiladiki, tizim o'z-o'zidan emas, balki boshqa ko'plab elementlar qurshovida mavjud bo'ladi. Ayrim masalalarni hal etishda bizni bu tashqi muhitning barcha elementlari emas, balki ushbu masala nuqtainazaridan tashqi muhitni tashkil etuvchi, ko'rib chiqilayotgan tizimga biror-bir aloqasi bo'lgan elementlargina qiziqitiradi. Tashqi muhit — bu qurilayotgan tizimga ta'sir ko'rsatuvchi yoki qurilayotgan masala sharoitida uning ta'siri ostida bo'lgan, tizimdan tashqaridagi har qanday tabiat elementlaridir. Chunki, real sharoitlarda tizimlarning har biri alohida emas, balki boshqalari yonida, bir-biriga bog'liq, holda ishlaydi. Tizimlarni taxlil va sintez qilish chog'ida aloqalarning ikki xil turi ajralib turadi: ichki va tashki aloqa. Tashki aloqaga ega tizimlar ochiq deb, unga ega emaslar esa yopiq aloqa deb ataladi.

Tizimlar tasnifi. Tizimlarni qiyoslash va farqlash, ularning bir-biriga o'xshashlari va farqlilarini ajratish orqali tasniflash amalga oshiriladi.

Tasniflash — bu faqat borliq, modeli va uni turli belgilar, ya'ni, kirish va chiqish jarayonlarining bayoni, ularning kelib chiqishi, boshqaruv turi, boshqaruvning resurslari bilan ta'minlanganligi va hakoza bo'yicha amalga oshirish mumkin. Bizni tizimning kelib chiqishi bo'yicha tasniflash sezdiradi. Sun'iy tizimlar — bu inson tomonidan yaratilgan tizimlardir.

Tabiiy tizimlar bu tabiatda yoki jamiyatda inson ishtirokisiz yuzaga kelgan tizimlar.

Eralash tizimlar ta'biy va sun'iy tizimlarni o'z ichiga oladi.

Ergonomik tizimlar — bu, «mashina — inson- operator» majmui.

Biotexnik tizimlar — tirik organizmlar va texnik qurilmalar kiradigan tizimlardir.

Tashkiliy tizimlar — bu, zaruriy vositalar bilan jixozlangan kishilar jamoasidan tashkil topgan tizimlar sanaladi.

Tashkiliy tizim — boshqarish, shuningdek, tashkiliy tuzilma, maqsadlar, boshqarish samaradorligi va xodimlarni rag'batlantirish qoidalari mezonlari uchun foydalanadigan, hodimlarning yurishturishi va texnik vositalarning ishlatilish tartibini belgilovchi qoidalar yig'indisidir.

Tashkiliy tizimlar ishlab chiqarish vositalaridan foydalanuvchi kishilar jamoasining ishlab chiqarish faoliyatini boshqarish uchun mo'ljallangan. Oxirgisi ancha muhim holat hisoblanadi, chunki tashkiliy tizimlar texnik vositalarning o'ziga xosligini, xususan, boshqaruv vositalarini hisobga olishi lozim.

Tizimda boshqaruv ob'ekti — bu muayyan moddiy zaxiralarga ega va aniq mahsulotni olishga yo'naltirilgan ishlab chiqarish operatsiyalarini bajaruvchi vazirlik, idora, korxonalar, sex, ishlab chiqarish, uchastkalar, ijrochilar jamoasi yoki ayrim shaxslardir. Boshqaruv ob'ektining faoliyati ishlab chiqarish jarayoni chog'idagi turli holatlardagi vazifalarni amalga oshirishga bo'ysindirilgan.

Boshqaruv organi ob'ektni boshqarish uchun tashkiliy tizimdan foydalanuvchi shaxs yoki shaxslar guruhi sanaladi.

Tashkiliy tizimlar avtomatlashtirilgan yoki avtomatlashtirilmagan bo'lishi mumkin.

Tashkiliy tizimlar bir qator o'ziga xos xususiyatlarga ega. Dastlabki o'ziga xosligi shuki, tizimning asosiy elementi murakkab, faol tizim bo'lgan insondir. Inson yurish-turishi, hulqi jihatlarining amaliy talablarini bayon etuvchi norasmiy modellarini tuzish juda murakkab, ba'zan esa iloji yo'q. Ayni paytda inson tashkiliy tizimlarda qaror qabul qiluvchi shaxs (QQSh) hisoblanadi.

Tashkiliy tizimlarning ikkinchi o'ziga hosligi ko'p maqsadli ishlash qususiyatidir. Ushbu tizimlar faoliyatining samaradorligi umuman olganda ham uning kichik tizim va elementlarini tashkil etuvchilariga ko'plab miqdordagi texnik, iqtisodiy va ijtimoiy ko'rsatkichlar bilan belgilanadi. Samaradorlikni baholashning ko'pqirraligi ko'pgina o'zaro bog'liq jihatlar bo'yicha boshqarishni tashkil etish zaruriyatiga olib keladi. Bunda tizimning boshi elementlari bilan moddiy va axborot jihatdan o'zaro ta'sirini tashkil etish talab etiladi.

Uchinchi o'ziga xoslik — tashkiliy tizimlarning uzluksiz rivojlanishini o'z ichiga oladi, u yangi ehtiyojlar paydo bo'lishi, bu

extiyojlarni tashqi va ichki shart-sharoit hamda o'zgarishlar bilan bog'liq holda qondirish yo'llarini takomillashtirishdan iborat. Oqibatda, ob'ektlar tarmotspari doimiy o'zgaradi, uning elementlari o'rtasida yangi aloqalar paydo bo'ladi. Shuningdek, ham alohida ob'ekt, ham umuman tizim sifatida boshqarish tizimi o'zgaradi.

II Bob. SHIFOXONAGA OID MA'LUMOTLAR BAZASINI YARATISH BOSQICHLARI

2.1. Ma'lumotlarni tashkil qilish

Mashina muhitida ma'lumotlarni tashkil etish mantiqiy va fizik bosqichlar bilan tasvirlanadi. Ma'lumotlarni bevosita mashina «tashuvchisi»da joylashtirish usulini belgilaydi. Ma'lumotlarni zamonaviy amaliy dasturlar vositalirida tashkil etishning bir bosqichi foydaloanuvchining aralashuvisiz avtomatik ravishda ta'minlaydi. Amaliy va universal dastur vositalarida foydalanuvchi, qoida tariqasida, ma'lumotlarni mantiqiy tashkil etish haqidagi tushunchalar bilan operasiyalar bajaradi.

2.2. Ma'lumotlarni mantiqiy tashkil etish

Mashina «tashuvchisi»da ma'lumotlarni mantiqiy tashkil etishda, foydalanayotgan dastur vositalari va mashinachi muhitida ma'lumotlarni yuritishga bog'liq. Ma'lumotlarni tashkil etishning mantiqiy usuli ma'lumotlar tuzilishining foydalanilayotgan turi hamda dastur vositalari orqali ta'minlanadigan model shaklida ifodalanadi.

Ma'lumotlar modeli – o'zaro bog'langan ma'lumotlar tuzilishi va ular ustida bajariladigan operasiyalar majmuidir. Model shakli va unda foydalanadigan ma'lumotlar tuzilishining turi MBBT (ma'lumotlar bazasini boshqarish tizimi) yordamchi model yoki ma'lumotlarni ishlashning amaliy dasturi yaratiladigan dasturlash tizimi tilida qo'llanuvchi ma'lumotlarni tashkil etish va ishlash tamoyillarni aks ettiradi.

Qayd etish lozimki, aynan bir axborotni mashina ichki muhitida joylashtirish uchun turli xil ma'lumotlar tuzilishi va modellari ishlatilishi mumkin. Ulardan birini tanlash axborotlar bazasini yaratayotgan foydalanuvchi

ixtiyorida bo'lib, avtomatlashtirilayotgan vazifaning murakkabligi, axborot hajmi bilan belgilanadi[16].

Ma'lumotlar modeli quyidagi tarkibiy qismdan iborat:

1. Foydalanuvchining ma'lumotlar bazasiga munosabatini namoyish etishga mo'ljallangan ma'lumotlar tuzilmasi.

2. Ma'lumotlar tuzilishida bajarilishi mumkin bo'lgan operasiyalar. Ular ko'rib chiqilayotgan ma'lumotlar modeli uchun ma'lumotlar tilining asosini tashkil etadi. Yaxshi ma'lumotlar tuzilmasining o'zigina yetarli emas. Ma'lumotlarni aniqlash tili (MAT) va ma'lumotlar bilan amallar bajarish tili (MBAT)ning turli operasiyalari yordamida bu tuzilma bilan ishlash imkoniga ega bo'lish zarur.

3. Yaxlitlikni nazorat qilish uchun cheklashlar. Ma'lumotlar modeli uning yaxlitligini saqlash va himoya qilishga imkon beruvchi vositalar bilan ta'minlangan bo'lishi lozim. Quyida shunday cheklanishlarning namunalari keltirilgan.

a) har bir «kichik daraxt» uzelliga ega bo'lishi kerak ma'lumotlarning iyerarxik bazalarda dastlabki uzelsiz «tugma» uzellarini saqlash mumkin emas.

b) ma'lumotlarning relyasion bazasiga nisbatan bir xil kortyojlar bo'lmaydi. Fayl uchun bu cheklash barcha yozuvlarning yagonaligini talab etadi.

2.3. Fayllar modeli

Fayllar tizimida yassi fayl turidagi model amalda bo'ladi. Bunday modelda mashina ichi axborotlar bazasi (AB) bir xil ko'rinishdagi yozuvlardan tuzilgan o'zaro bog'lanmagan (mustaqil) fayllar bilan chiziqli (bir bosqich) tuzilmalarining yig'indisida iborat.

Fayllar modeli ma'lumotlarning tuzilmalari. Fayllar modeli ma'lumotlarning asosiy tuzilmalari (strukturalari) –maydon, yozuv, fayl. Yozuv ma'lumotlarning ishlashning asosiy tuzilma birligi tezkor va tashqi xotira o'rtasidagi almashuv birligi hisoblanadi.

Maydon - ma'lumotlarni tashkil etishning oddiy birligi bo'lib, axborotning alhida, bo'linmas birligi bo'lishi rekvizitga mos keladi.

Yozuv – mantiqan bog'langan rekvizitlarga mos keluvchi maydonlar yig'indisidir. Yozuvning tuzilishi o'z tarkibiga kiruvchi har bir oddiy ma'lumotga ega maydonlar tarkibi va ketma – ketligi bilan belgilanadi.

Fayl - aloxida maydonlarda mazmunga ega bo'lgan bir xil tuzilishdagi ko'plab yozuv nus'halaridir. Yozuv nus'hasi maydonlarning ma'lum mazmunga ega bo'lgan yozuvlarini aks ettiradi. Fayl yozuvi tuzilishi-chiziqli, ya'ni maydon yagona mazmunga ega va guruh ma'lumotlar mavjud emas. Har bir yozuv nusxasi yagona yozuv kaliti bir xil bo'ladi. Umumiy hollarda yozuv kalitlari ikki xil ko'rinishda: dastlabki (birlamchi) va ikkilamchi kalitlar bo'ladi.

Dastlabki kalit (DK) – yozuvni ma'no jihatidan bir xillashtiruvchi bir yoki bir necha maydonlardir. Dastlabki kalit bir maydondan iborat bo'lsa u oddiy, agar bir nemcha maydonli bo'lsa – turli tarkibli kalit hisoblanadi.

Ikkilamchi kalit (IK) – dastlabkidan farqli o'laroq, shunday maydonki, uning mazmuni faylning bir necha yozuvlaridan takrorlanadi, ya'ni u yagona emas. Agar dastlabki kalitning mazmuniga ko'ra faqat bitta yozuv nus'hasi topilsa, ikkinchi kalit bo'yicha bir necha nusxa topilishi mumkin.

Ma'lumotlarning sanab o'tilgan tuzilishi bir qator MBBT da qo'llaniladi. Bu esa ushbu tushunchani ma'lum ma'noda umumlashtiradi.

Indeksatsiyalash. Kalit bilan fayl yozuvlariga kirishning samarali vositasi indeksisiyalashdir. Indeksisiyalashda indeksali qo'shimcha fayl yaratiladi. U

ma'lumotlar faylli kalitining barcha mazmuni tartiblashtirib o'zida saqlaydi. Indeksli faylda har bir kalit mazmuni uchun ma'lumotlar fayllining tegishli yozuviga mo'ljallangan ko'rsatgich bo'ladi. Hajmi asosiy fayldan kichik indeksli fayl mavjud bo'lganda berilgan kalit bo'yicha qidirilayotgan yozuv tez topiladi. Ma'lumotlar fayllida yozuv ko'rsatkichi yordamida ushbu yozuvga bevosita yo'l ochiladi. Indeksasiyalash faqat dastlabki emas, balki ikkilamchi kalit bo'yicha ham amalga oshirilishi mumkin[20].

2.4. Ma'lumotlarni tashkil qilishning tarmoqli va iyerarxik modellari

Mashina muhitidagi ma'lumotlarning murakkabroq modeli (faylga nisbatan) tarmoqli va iyerarxik modellar bo'lib, ular tegishli turdagi ma'lumotlar bazasini boshqarish tizimida ta'minlanadi. Ma'lumotlarning tarmoqli yoki iyerarxik modeli MBBTda ma'lumotlarni mantiqiy tashkil etishning tegishli usulini aks ettiradi. Bunday model o'zaro bog'liq obyektlarning majmuidir. Ikki ob'yektning aloqasi ularning bir-biriga bo'ysunishini aks ettiradi. Tarmoqli yoki iyerarxik modelning ob'yekti MBBTda taminlanadigan ma'lumotlar tizimining asosiy turidir. Turli MBBTlarda tuzilishning bu ko'rinishi turlicha belgilanishi va nomlanishi mumkin (yozuv turi, fayl, segment)[15].

2.5. C++ Builder dasturida ilovalar bilan ishlash

Standart komponentalar

TMainMenu - bosh menyu buyruqlari panelini va ularga mos keladigan tushib qoladigan menyularni yaratadi. Barcha menyu buyruqlarining identifikatorlari menyuning har qanday konkret buyrug'iga kirish huquqiga ega bo'lgan Items xususiyati bilan aniqlanadi, AutoMerge xususiyati Merge va Unmerge metodlari bilan birgalikda turli shakldagi menyularning birlashish jarayonini boshqaradi.

TPopUpMenu - shakl yoki bironta boshqa komponenta uchun maxsus menu yaratadi. E'tiborga oling, aynan shu maqsad uchun har qanday boshqa komponenta

PopupMenu xususiyatiga ega bo'lib, bu xususiyatda siz bilan bog'liq menyuga iqtibos qilishingiz mumkin. Agar siz sichqonchaning o'ng tugmasini shaklga yoki berilgan komponenta mansub bo'lgan biron boshqa elementga bosish bilan maxsus menyu ekranda paydo bo'lishini xohlasangiz, AutoPopup xususiyatining true qiymatini o'rnating. Voqea qayta ishlatgichi OnPopup yordamida bevosita maxsus menyuning paydo bo'lishi oldidan bajariladigan protsedurani aniqlash mumkin.

TLabel - shaklda tahrir qilib bo'lmaydigan statik matnning to'rtburchak sohasini aks ettiradi. Odatda matn boshqa komponenta nomidan iborat bo'ladi. Nom matni Caption xususiyatining qiymatidir. Alignment xususiyati matnni tekislash usulini aniqlaydi. Shrift o'lchami avtomatik tarzda sohaning maksimal to'ldirilishiga mos kelishi uchun AutoSize xususiyatining true qiymatini o'rnating. Kalta soha ichida matnning hammasini ko'rish imkoniga ega bo'lish uchun WordWrap xususiyatining true qiymatini bering. Transparent xususiyatining true qiymatini o'rnatsangiz, boshqa komponentaning bir qismini to'g'ri uning ustida joylashtirilgan ncm orasidan ko'rinib turadigan qilishingiz mumkin.

TEdit - axborot yakka satrining tahrir qilinayotgan kiritishidagi to'rtburchak sohani shaklda aks ettiradi. Tahrir sohasining ichidagi boshlang'ich narsalarni Text xususiyatining qiymati bo'lgan satr aniqlaydi. TEdit komponentasi TCustomEdit sinfining to'g'ridan-to'g'ri hosilasi bo'lib, uning barcha xususiyatlari, metodlari va voqealariga vorislik qiladi.

TMemo - axborot ko'plab satrining tahrir qilinayotgan kiritishidagi to'rtburchak sohani shaklda aks ettiradi. Tahrir sohasining ichidagi boshlang'ich narsalarni Lines xususiyatining qiymati bo'lgan satrlar massivi aniqlaydi. Ushbu xususiyat qiymati ustunida tugmachani bossangiz, ro'yxat elementlari muharririning darchasi ochiladi. TMemo komponentasi TCustomMemo sinfining to'g'ridan-to'g'ri hosilasi bo'lib, uning barcha xususiyatlari, metodlari va voqealariga vorislik qiladi.

TButton - yozuvli to'rtburchak tugmani yaratadi. Tugmacha bosilganda, dasturda biror bir xatti-harakat nomlanadi (initsiallashtiriladi). Tugmachalar ko'proq dialogli darchalarda qo'llanadi. Default xususiyatining true qiymati tomonidan tanlab olingan yashirin tugmacha, dialog darchasida har gal Enter klavishi bosilganda, OnClick voqea qayta ishlatgichini ishga tushiradi. Cancel xususiyatining true qiymati tanlab olgan uzish tugmachasi, dialog darchasida har gal Escape klavishi bosilganda, OnClick voqea qayta ishlatgichini ishga tushiradi. TButton komponentasi TButtonControl sinfining hosilasi hisoblanadi.

TCheckBox - ikkita holatga hamda tavsifiy matnga ega bo'lgan kvadrat chekboksni yaratadi (bunda tavsifiy matn chekboksning vazifasini spetsifikatsiya qiladi). Boks holatini bildiruvchi «check» biror bir variantning tanlanishiga mos keladi (boks ustidan tortilgan chiziq bilan belgilanadi) «unchecked» holati esa tanlov olib tashlanishiga mos keladi bunda Checked komponentasining xususiyati mos ravishda o'zgaradi hamda OnClick voqeasi yuzaga keladi. Tavsifiy matn Caption xususiyatida saqlanadi. AllowGrayed xususiyatining true qiymatini o'rnatib, boksni to'qroq rangli (masalan, kulrang) qilish mumkin. State xususiyati joriy holatni va boks rangini aks ettiradi. TCheckBox komponentasi TButtonControl sinfining hosilasidir.

TRadioButton - ikkita holatga hamda tavsifiy matnga ega bo'lgan yumaloq tugmachani yaratadi (bunda tavsifiy matn yumaloq tugmachaning vazifasini spetsifikatsiya qiladi). Radio-tugmalar bir-birini istisno qiladigan tanlov variantlarining to'plamidan iborat: ya'ni ushbu vaqt daqiqasida faqat bitta tugma tanlab olinishi mumkin (ichki qora doiracha bilan belgilanadi), avval tanlangan tugmadan esa tanlov avtomatik tarzda olinadi. Radiotugma bosilganda, Checked komponentasining xususiyati ham mos ravishda o'zgaradi va OnClick voqeasi yuzaga keladi. Odatda radio-tugmalar avvaldan shaklda o'rnatilgan konteyner

ichigajoylashtiriladi. Agar bitta tugma tanlangan bo'lsa, ushbu guruhga mansub barcha boshqa tugmalarning tanlovlari avtomatik tarzda olib tashlanadi.

TComboBox - tahrir sohasi hamda matn variantlarining tushib qoladigan ro'yxati kombinatsiyasini tanlash uchun yaratadi. Text xususiyatining qiymati bevosita tahrir sohasiga kiritib qo'yiladi. Foydalanuvchi tanlab olishi mumkin bo'lgan ro'yxat elementlari Items xususiyatining dasturning bajarilish paytida tanlab olinishi mumkin bo'lgan element raqami ItemIndex xususiyatining, tanlab olingan matnning o'zi esa SelText xususiyatining ichida bo'ladi. SelStart va SelLength xususiyatlari matnning qaysi qismini tanlab olishni belgilab berish yoki matnning qaysi qismi tanlab olinganini bilish imkonini beradi.

TScrollBar - darcha, shakl yoki boshqa komponenta ichidagilarini ko'rib chiqish uchun, masalan, biror bir parametr qiymatini berilgan interval ichida harakatlanishi uchun yugurgichli aylantirish lineykasini yaratadi. Aylantirilayotgan obyekt xulq-atvorini OnScroll voqealar qayta ishlatgichi aniqlaydi. Foydalanuvchi lineykaning o'zida sichqonchani bosganda (yugurgichning har ikkala tomonida), yugurgich qanchaga surilishi kerakligini LargeChange xususiyatining qiymati aniqlab beradi. Foydalanuvchi sichqonchani strelkali tugmachalar (lineyka oxiridagi) ustida bosganda yoki pozitsiyalash tugmachalarini bosganda, yugurgich qanchaga surilishi kerakligini SmallChange xususiyatining qiymati aniqlab beradi. Min va Max xususiyatlarining qiymatlari yugurgichning yo'l qo'yilishi mumkin bo'lgan joy almashinuvlari intervallarini belgilaydi. Sizning dasturingiz yugurgichni Position xususiyatining qiymati aniqlab beradigan kerakli pozitsiyaga joylashtirishi mumkin. SetPcirums metodi bir paytning o'zida Min, Max va Position ga tegishli barcha xususiyatlar qiymatlarini aniqlab beradi.

TPanel - boshqa komponentalarni o'z ichiga olishi mumkin bo'lgan bo'sh panelni yaratadi. Siz TPanel dan o'z shaklingizda instrumentlar paneli yoki holatlar satrlarini yaratish uchun foydalanishingiz mumkin. TPanel panel komponentasi

TCustomPanel sinfining hosilasi bo'lib, uning barcha xususiyatlari, metodlari va voqealari to'liq vorislik qiladi. Windows komponentalari Windows komponentalari sizning dasturingizga Windows ning 12 ta interfeys elementlarining ulanishini amalga oshiradi.

TTabControl - bir-birini qisman yopib turadigan qo'shimcha kartoteka elementlarining to'plamini aks ettiradi. Qo'shimcha ilovalarning nomlari Tabs xususiyatining ro'yxatiga ushbu xususiyat qiymati ustunidagi tugmacha bilan kiritiladi.

TPageControl - ko'p varaqli dialogni tashkil etish uchun mo'ljallangan qisman bir-birining ustini yopuvchi qo'shimcha kartoteka ilovalari ko'rinishidagi maydonlar to'plamini aks ettiradi. Tegishli qo'shimcha ilovaga ega bo'lgan yangi dialogli sahifani yaratish uchun berilgan komponentaning kontekst menusidan New Page opsiyasini tanlab oling. Siz konkret sahifani quyidagi usullarning biri yordamida faollashtirishingiz mumkin: ActivPage xususiyatining tushib qolayotgan ro'yxatidan tanlab olingan sichqoncha yordamida shuningdek, kontekst menuning NextPage va Previous Page opsiyalari yordamida qo'shimcha ilovalarni varaqlash vositasida. PageIndex xususiyati faol sahifa raqamiga ega. Tab Visible xususiyatining false qiymatini o'rnatib, shu sahifani ko'rinmas qilish mumkin.

TImageList - bir xil o'lchamdagi grafik tasvirlardan iborat kolleksiya uchun konteyner yaratadi. Bu tasvirlarning har bittasini ularning 0 dan n-1 gacha qiymatlar intervalidagi indeks bo'yicha tanlab olish mumkin. Grafik kolleksiyalar bitli obrazlar yoki piktogrammalarning katta to'plamlariga samarali xizmat ko'rsatish uchun qo'llanadi. Ushbu bitli obrazlar va piktogrammalar eni k*n bo'lgan yagona bitli obraz sifatida saqlanadi. Niqobli monoxrom tasvirlar tiniq trafaretlar sifatida aks ettiriladi. TImageList komponentasi saqlanayotgan tasvirlarni yozish, tanlash va chiqarish jarayonlarini osonlashtiruvchi metodlarga ega.

THeaderControl - dasturning bajarilish jarayonida enini o'zgartirish mumkin bo'lgan ustunlar sarlavhalarining to'plami uchun konteyner yaratadi. Sections xususiyatida sanab o'tilgan sarlavhalarni axborot maydonlari ustida, masalan, TListBox komponentalari ro'yxatlari ustida joylashtirish mumkin. Sarlavhali seksiyalar muharririning darchasi ushbu xususiyat qiymatlarining grafasida tugmacha bilan ochiladi.

TRichEdit RTF (Rich Text Format) - formatidagi ko'p satrli axborotning tahrir qilinadigan kiritish sohasini aks ettiradi. ATF formati shrift atributlari va paragraflarni formatlashning turli variantlarini o'z ichiga oladi. Bu formatni ko'plab professional matniy protsessorlar masalan: MicrosoftWord qabul qiladi.

TBitBtn - bit $\text{O}^{\wedge}\Pi^{\wedge}$ tasviri tushirilgan tugmachani yaratadi. Bunday tugmachalar ko'pmq maxsus dialogli darchalarda qo'llanadi. Grafik tugmachalar bit $\text{O}^{\wedge}\Pi^{\wedge}$, ularning ko'rinishi va tugmachada joylashishini spetsifikatsiyalash xususiyatlariga ega bo'ladi. Siz C++Builder qurilmasiga kirgan atohida tasvirlar katalogidagi grafik tugmalarning tayyor stillaridan foydalanishingiz hamyoki bo'lmasa tasvirlarni tahrir qilish tizimlarining biri tomonidan yaratilgan suratlardan foydalanishingiz ham mumkin.

TSpeedButton - odatda ma'lum menu buyruqlarini tez chaqirish yoki \wedge mlami o'rnatish paneli (TPanel) da joylashtiriladigan grafik tugmani yaratadi. Tezkor tugmaning turli holatlariga (masalan, « \wedge s \wedge an», «qo'yib yuborilgan», «taqiqlangan» va h.k.) turli bit $\text{O}^{\wedge}\Pi^{\wedge}$ mos kelishi mumkin. Bir- birining o'rnini bosadigan tasvirlar va yozuv matnini tanlash uchun xususiyatlar mavjud. Kengayishli tasvirlar fayllari muharririning darchasi Glyph xususiyati qiymatlarining grafasidagi tugma bilan ochiladi. Tez tugmalarning boshqa xususiyatlari ularning biror bir guruhdagi ishini tashkil etadi.

TMaskEdit - o'ziga xos formatdagi ma'lumotlarning tahrir qilinadigan nazoratdagi to'rtburchak sohasini yaratadi. Kiritilayotgan matnning to'g'riligi ruxsat etilgan formatlarni kodlovchi niqob vositasida tekshiriladi. Bu formatlarga matn kiritilgan va foydalanuvchiga taqdim etilgan bo'lishi mumkin (sana, vaqt, telefon raqami va h.k.). EditMask xususiyati joriy niqob kodini saqlaydi. Niqoblar muharriri darchasi ushbu xususiyat qiymatlari grafasida tugma bilan ochiladi. TMaskEdit komponentasi TCustomMaskEdit sinfining to'g'ridanto'g'ri hosilasidir. U satrlar yoki ustunlar bo'yicha belgili ketmaketliklarni aks ettirish uchun mo'ljallangan muntazam (regulyar) to'rni yaratadi.

TStringGrid - komponentaga tegishli barcha xususiyatlarning nomlari va vazifalari bo'lib, siz ulardan dasturni loyihalash bosqichida to'la foydalanishingiz mumkin. Ular keyingi paragrafda tavsifi berilgan TDrawGrid komponentasi xususiyatlariga to'liq to'g'ri keladi. SimvoNi ketma-ketliklar bilan bog'liq barcha obyektlar kerakli obyektga murojaat qilish imkonini beradigan Objects xususiyatida mujassam bo'lgan. Dastur bajarilish paytida simvoNi ketma-ketliklar va setka ustunining ular bilan bog'liq obyektlari Cols xususiyati bilan adreslanadi. Rows xususiyati setka satrlari bilan xuddi shunday ish tutish imkonini beradi. Setkaning barcha simvolli ketma-ketliklari setkaning kerakli uyasini adreslaydigan (manzillaydigan) Cells xususiyatida mujassamdir.

TImage - shaklda grafik tasvir konteynerini yaratadi (bu bit obrazi, piktogramma yoki metafayla bo'lishi mumkin). Tasvirlar fayllari muharririning darchasi Picture xususiyati qiymatlari grafasidagi tugma bilan ochiladi. Konteyner o'z o'lchamlarini tasvirni to'liq sig'diradigan qilib o'zgartirishi uchun AutoSize xususiyatining true qiymatini o'rnatish. Kichikroq o'lchamdagi dastlabki tasvir butun konteynerga cho'zilib ketishi uchun Stretch xususiyatining true qiymatini o'rnatish. Tasvirlar fayllarining dinamik yuklanishi va saqlanishi

uchun Picture obyekt xususiyatining LoadFromFile va SaveToFile metodlaridan quyidagi turlar yordamida foydalaning:

```
Image->Picture->LoadFromFile(«<fayl nomi>»);
```

```
Image->Picture->SaveToFile(«<fayl nomi>»);
```

TScrollBar - darchada o'lchamlari o'zgaruvchan boksni yaratadi, bu boks shu topdayoq avtomatik tarzda zaruratga ko'ra aylantirish lineykalari bilan ta'minlanadi. Aylantirib ko'rish boksi yordamida darchaning ayrim sohalarini aylantirib ko'rishdan himoyalash mumkin. Masalan, instrumentlar paneli va holat panelini himoyalash uchun avval darchani aylantirish lineykasini berkitib qo'ying, keyin esa aylantirish boksini mijoz sohasida instrumentlar paneli va holat paneli o'rtasida joylashtiring. Boksni aylantirib ko'rish lineykasi darchaga tegishli bo'lib ko'rinadi, biroq aylantirish faqat boks ichida amalga oshiriladi. Aylantirib ko'rish bokslaridan yana boshqachasiga ham foydalanish mumkin: ular biror bir darchada ko'plab aylantirib ko'rilayotgan sohalar (turlar) yaratish imkonini beradi. Turlar ko'pincha tijoriy matn protsessorlarida, buxgalteriya dasturlarida va loyihalarni rejalashtirish dasturlarida qo'llanadi. Aylantirib ko'rish boksi boshqa komponentlarga, masalan, Tbutton va TCheckBox ga ham ega bo'lishi mumkin.

C++Builder6 da qilingan ixtiyoriy dastur ilova deyiladi.

Ilova odatda Project1 deb nomlanadi, ammo dasturchi hohishiga ko'ra ixtiyoriy nom berish mumkini bo'ladi.

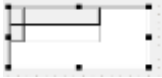
Ilova tarkibida har xil kengaytmali 9 ta va undan ortiq fayllar bo'lishi mumkin, masalan, Project1.res, Unit1.cpp, Unit1.dfm, Unit.h, shular jumlasidandir.

C++Builder6 da berilganlar bazasi bilan ishlash komponentalari



DataSource komponentasi - berilganlar bazasiga ba'zi

parametrlarni o'rnatish uchun ishlatiladi, ma'lumotlar o'zgarishini uzatish uchun ishlatiladi.



DBGrid komponentasi – ma'lumotlar bazasidagi jadvallarni vizual ifodalash va taxrirlash uchun ishlatiladi



DBNavigator komponentasi DBGrid komponentasi ma'lumotlar bazasidagi jadval yozuvlarini o'chirish, yozuv qo'shish, yozuvlarni sanash, yozuv boshiga va oxiriga o'tish kabi ishlar uchun ishlatiladi.

C++Builder6 da ma'lumotlar bazasi bilan ishlash komponentalari.



ADOConnection1 bu komponenta ADO komponentalarini ma'lumotlar bazasi bilan bog'lash va tranzaksiyani boshqarish imkonini beradi.



ADOTable Access da ma'lumotlar bazasidagi bitta jadvalni ifodalash uchun ishlatiladi, bu komponenta juda ko'p hodisa va xossalarga ega.



ADOStoredProc klient – server arxitekturasida ishlatiladi, bir yoki bir nechta jadvallardan yozuvlar to'plamini olish imkonini beradi.

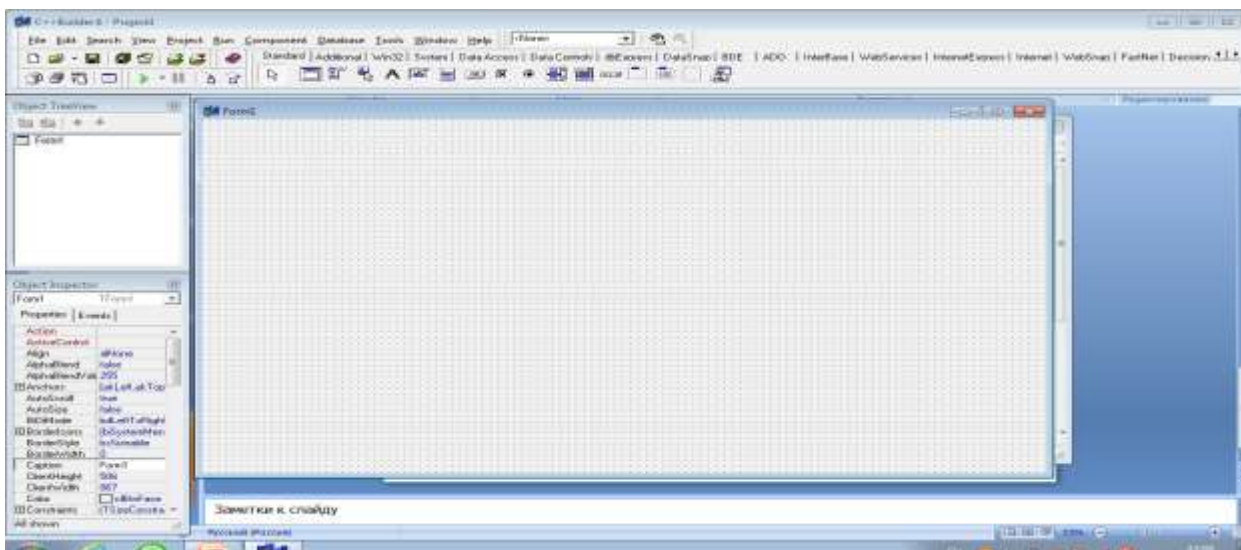


Bu komponent SQL so'rovlar yozishda foydalaniladi, Bu so'rovlar sirasiga SELECT, INSERT, DELETE, UPDATE, ALTER TABLE, CREATE TABLE kiradi.

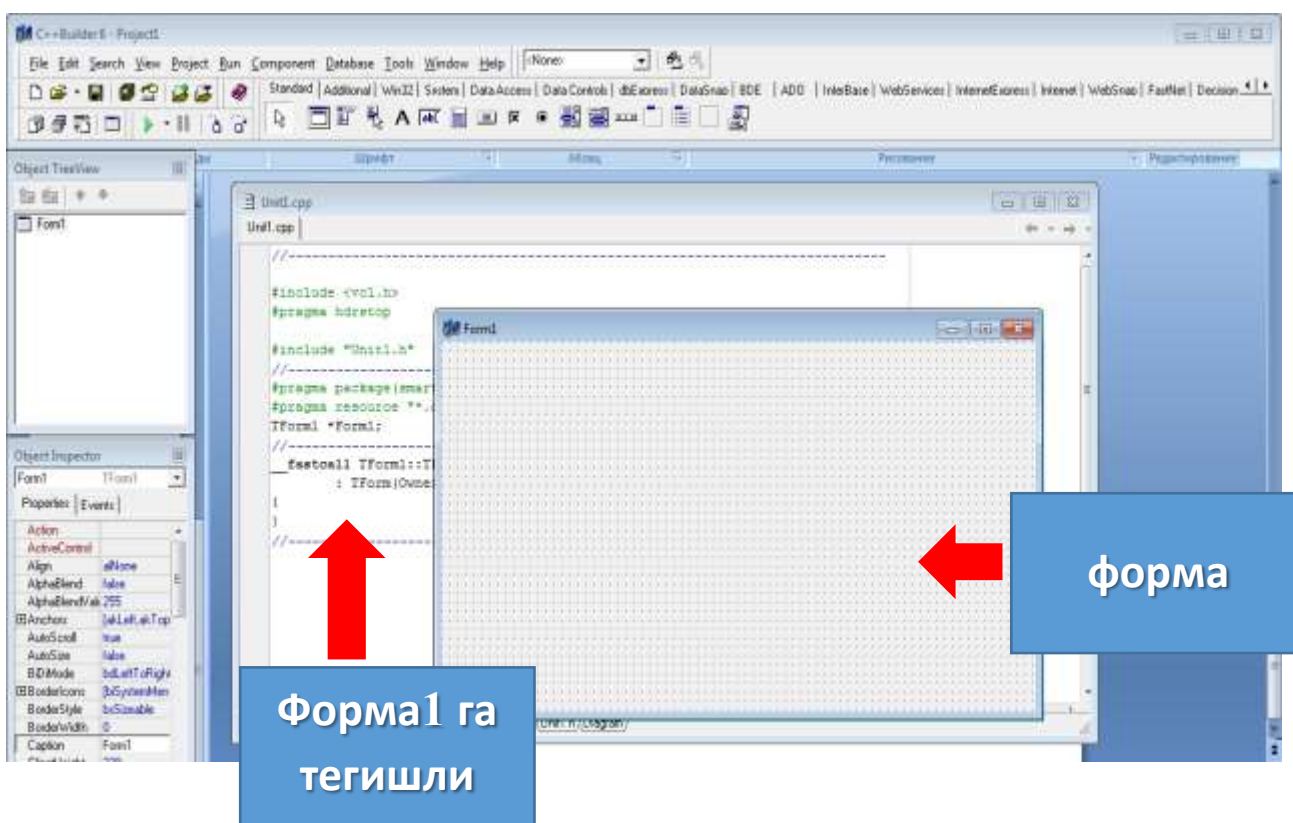


ADODataset bu komponenta bir yoki bir nechta ma'lumotlar bazasi bilan select so'rovi yordamida ishlash imkonini beradi.

C++Builder dasturi interfeysi

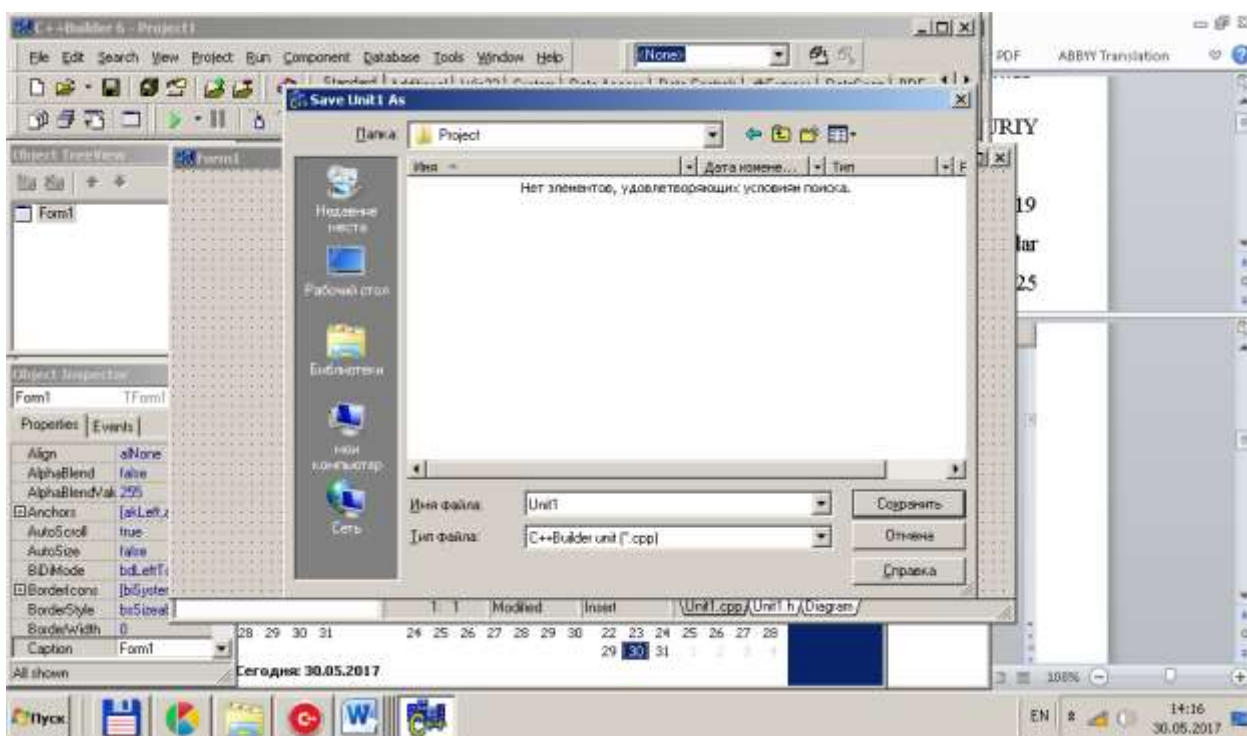


2.1. rasm. C++Builder dasturi interfeysi



2.2.Rasm. Project ning tuzilishi

C++Builder6 dasturida ilova yaratish. File menyusidan Save All buyrug'i tanlanadi natijada quyidagi saqlash oynasida mos fayl nomi kiritiladi birinchi navbatda Unit nomi kiritiladi keyin esa Project nomi kiritiladi va Сохранить tugmasi bosiladi.



2.3.Rasm. proyektни saqlash.

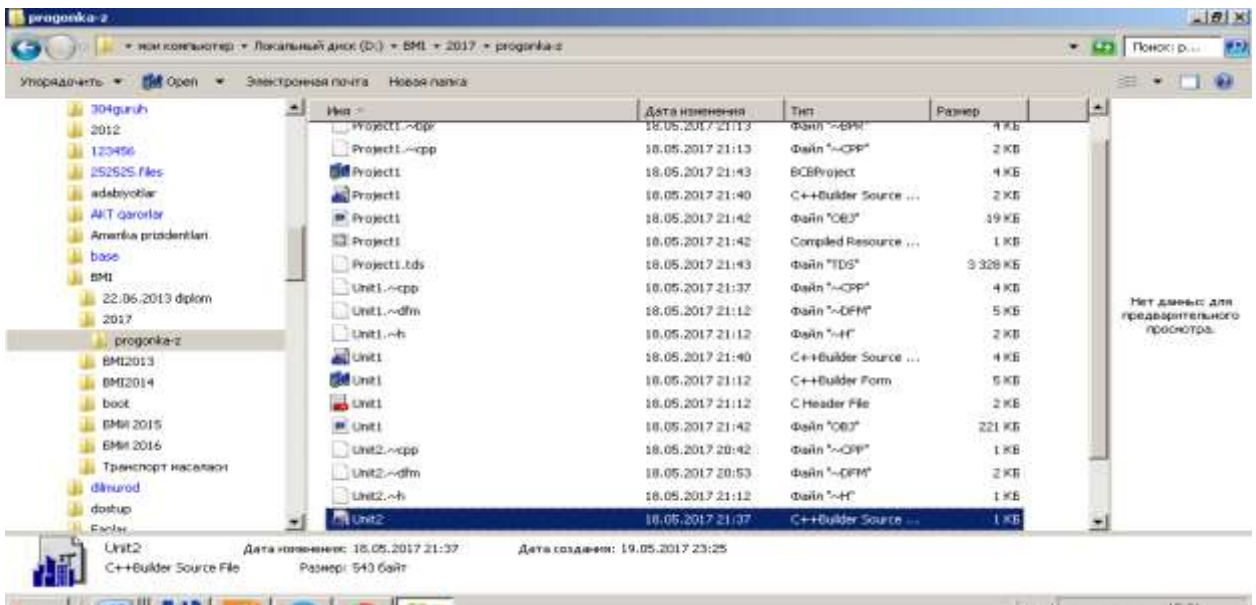
Biz Proectni project va Unitni project deb nomlaymiz

C++Builder6 dasturida Projectni kerakli papkaga saqlash



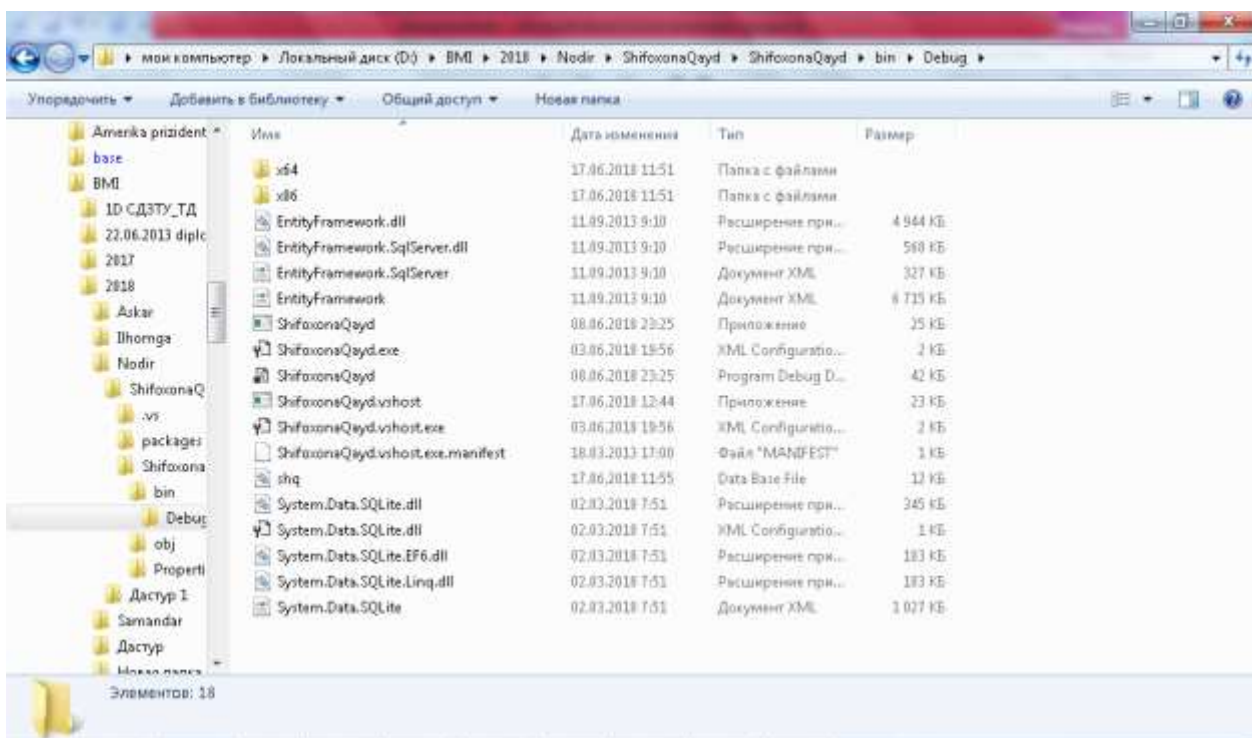
2.4.Rasm. proyektни saqlash.

Shundan so'ng shifoxona nomli papkada quyida ko'rsatilgan fayllar hosil bo'ladi



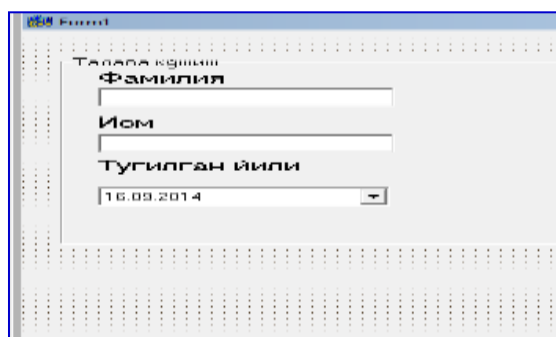
2.5.Rasm. Hosil bo'lgan fayllar.

CTRL+F9 tugmalarini birgalikda bosib Projectni kompilyatsiya qilgandan so'ng OTM nomli papkada *.exe va boshqa fayllar ham hosil bo'ladi.



2.6.Rasm. Hosil bo'lgan fayllar.

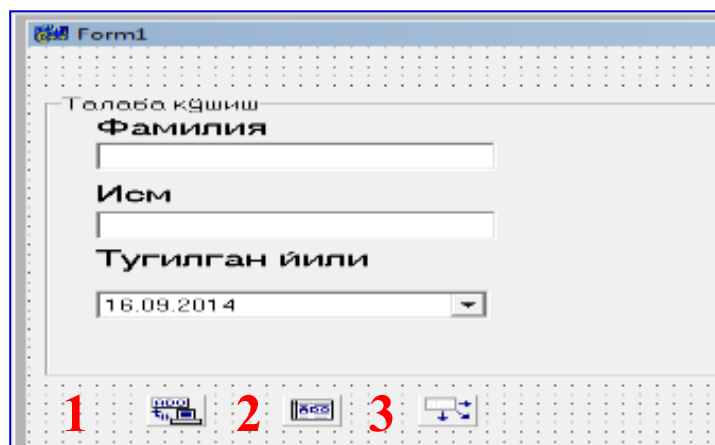
Oldin hosil qilgan Access faylimizni ham shu papka ichiga tashlaymiz. Project ga tegishli fayllar bitta papkada turishi kerak chunki exe faylni boshqa kompyuterda ham ishlashi uchun



2.7.Rasm. Formalardan foydalanish.

Dasturni ishga tushurib Familiya, Ism va boshqa maydonlarni to'ldirib OK tugmasini bosganimizda ushbu yo'lovchi haqidagi ma'lumotlar project bazasining yo'lovchi jadvalida hosil bo'lishi kerak.

Buning uchun biz Komponentalar ro'yxatidagi ADO, Data Access va Data Controls vkladkalariga tegishli komponentalardan foydalanamiz.



2.9.Rasm. Komponentalarni joylashtirish.

Yuqorida ko'rsatilgan komponentalarga izoh:

1-ADOConnection

2-ADOQuery

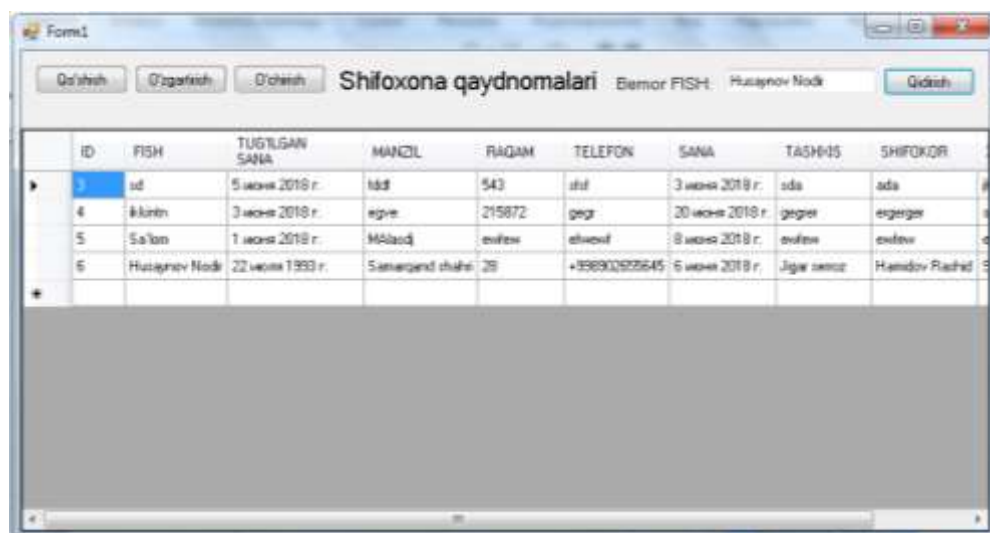
3-DataSource

1-ADOConnection – Baza va bizning Project ni bog'lash uchun xizmat qiladi.

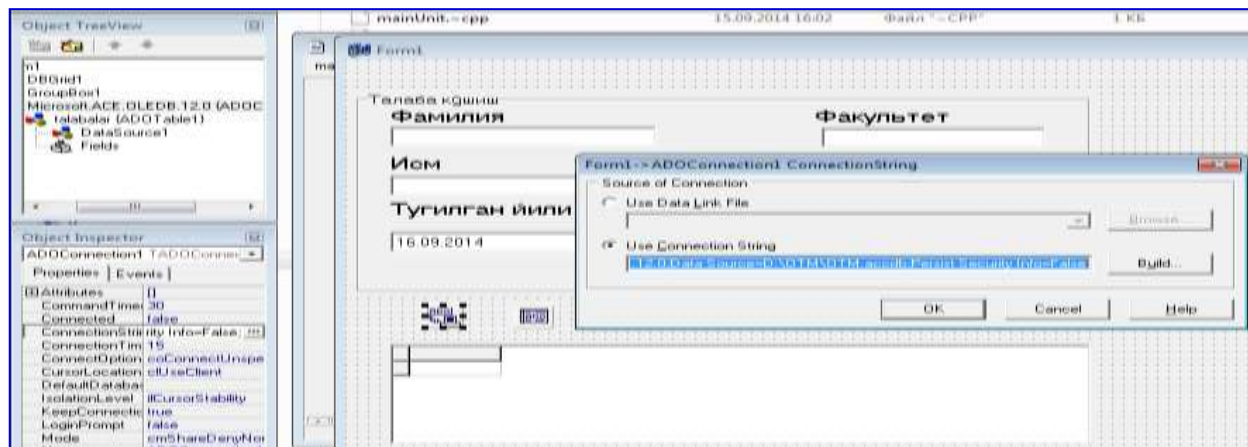
2-ADOQuery – Bazadagi talaba jadvaliga bog'lash uchun xizmat qiladi.

3 - DataSource–ADOQuery ma'lumotlarni DBGrid, DBText, DBEdit va boshqa ob'ektlarda aks ettirish uchun xizmat qiladi.

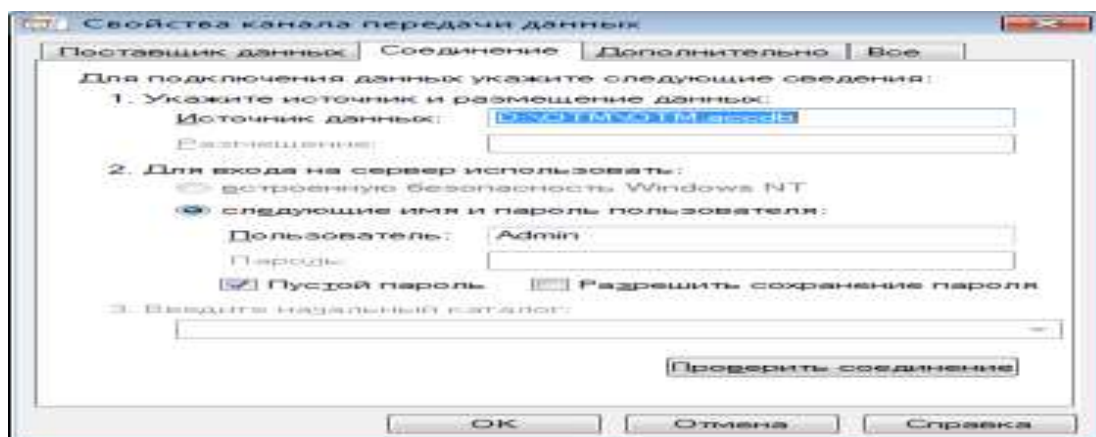
Endi Baza bilan bog'lanishga kirishamiz. Buning natijasini ko'rib turish uchun formamizga DBGrid joylashtiramiz.



2.10.Rasm. TDBGridni o'rnatish.

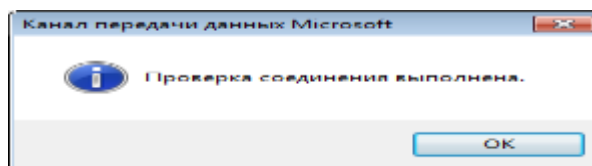


2.11.Rasm. ADOConnection.

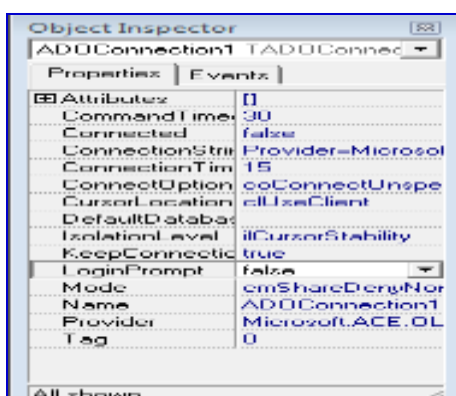


2.12.Rasm. Bog'lash oynasi

Источник данных bo'limiga bazaning yo'lini ko'rsatishimiz kerak shundan so'ng Проверит соединение tugmasi bosiladi natijada quyida rasmda ko'rsatilgan xabar chiqadi.



2.13.Rasm. Xabar oynasi

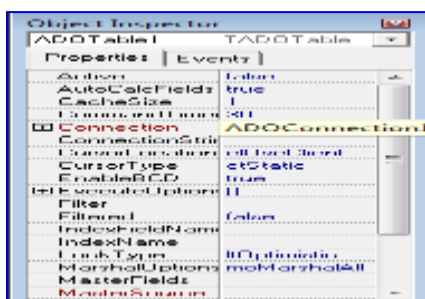


Connected –true Shundan so'ng(true qiymat qabul qilsa) baza bilan bog'lanish o'rnatiladi.

LoginPrompt –false Agar uning qiymati true belgilanadi qachonki Bazada parol bo'lganda.

ADOTable ni ADOConnection bilan bog'lash

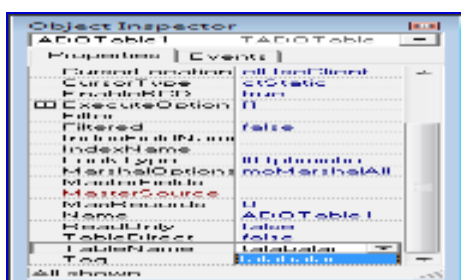
2.13.Rasm. Holat oynasi



Active qatoriga true qiymat beriladi.

ADOTablening Connectionga ADOConnection1 tanlanadi.

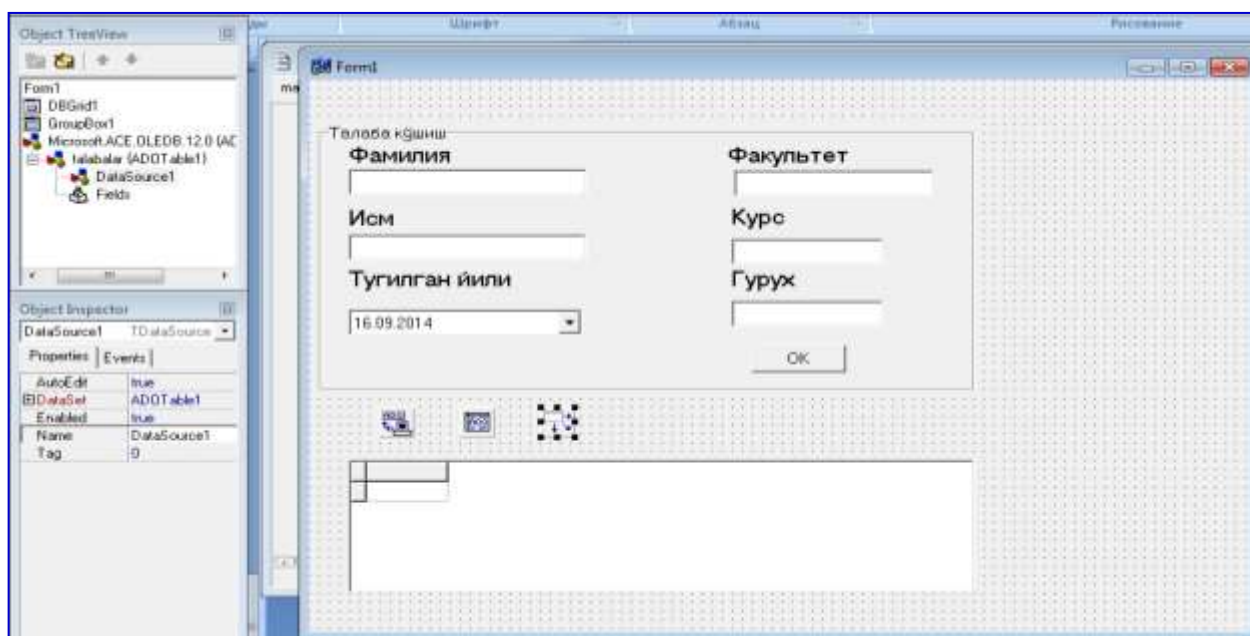
2.14.Rasm. Holat oynasi



TableName qatoriga talabalar jadvali ko'rsatiladi.

2.15.Rasm. Holat oynasi

DBGridda bazadagi ma'lumotlarni aks ettirish



2.16.Rasm. DBGrid da bazadagi ma'lumotlarni aks ettirish

DataSourcening DataSet qatoriga ADOTable1 ko'rsatiladi va DBGrid ning DataSource ga DataSource1 ko'rsatiladi.

Natijada DBGrid da bazadagi ma'lumotlarni aks etadi. Keyingi ishimiz ma'lumotlar bazasiga formadan turib talaba haqidagi ma'lumotlarni kiritish bo'ladi[10].

MBga yangi ma'lumotni kiritish. Buning uchun OK tugmasining OnClickga quyidagi kodlar yoziladi:

```
ADOTable1->Append(); // Jadvalga yangi yozuv qo'shish  
ADOTable1->FieldByName("Fam")->AsString=Edit1->Text;  
//Edit1 ga yozilgan Familiyani mos Fam maydoniga o'zlashtirish  
ADOTable1->FieldByName("Ism")->AsString=Edit2->Text;  
//Edit2 ga yozilgan Ismni mos Ism maydoniga o'zlashtirish
```

```
ADOTable1->Post();
```

```
//Jadvalga kiritilgan ma'lumotlarni saqlash
```

Izlash

Buning uchun biz ADOQuery komponentasidan foydalanamiz bu esa shifoxona SQL (zapro) so'rovni bilishi kerak.

Demak bu bo'yicha biz asosiy namunaviy ba'zi so'rov ko'rinishlarini keltirib o'tamiz:

SELECT * FROM bemorlar jadvalining barcha maydonlaridagi ma'lumotlarni chiqarish

```
SELECT * t.Fam, t.Ism, t.Kurs
```

```
FROM talabalar t
```

```
where ((t.Kurs=1) or (t.Kurs=2))and(t.Fam like "Ab*")
```

ADOQuery ga SQL zaprosni yozish

```
select *from talabalar t
```

```
Where t.Fam=' ' )or(t.Fam=")
```

```
ADOQuery1->SQL->Text="Select * from bemorlar t where t.Fam=
```

```
 '"+EditFamIzlash->Text+'";
```

```
ADOQuery1->Active=true;
```

2.6. Masalaning algoritmi

Bu masalani xal qilish uchun quyidagi algoritmni tashkil etamiz.

1. Kirish qismini tashkil etamiz.
2. Dastur bilan ishlashni va dastur haqida qo'shimcha ma'lumotlar beruvchi qisimni tashlil etamiz.

3. Ma'lumotlar bazasiga kiritilgan ma'lumotlarni maxsus oynada ko'rish qismini tashkil etamiz. Bu qismda ma'lumotlarni tezkor izlash, ma'lumotlar qo'shish, ma'lumotlarni o'chirish, ma'lumotlarni yangilash va chiqish kabi qo'shimcha qismlar mavjud bo'ladi.

4. Ma'lumotlarni qo'shish qismini tashkil etish.

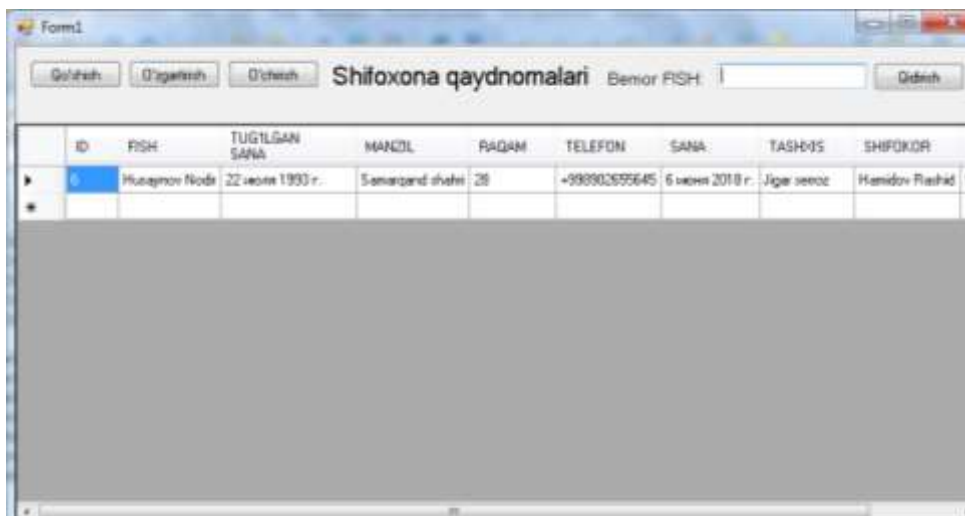
5. Ma'lumotlar bazasini tashkil etish.

6. Dasturdan chiqish qismini tashkil etish.

III Bob. DASTURIY TA'MINOT VA UN DAN FOYDALANISH

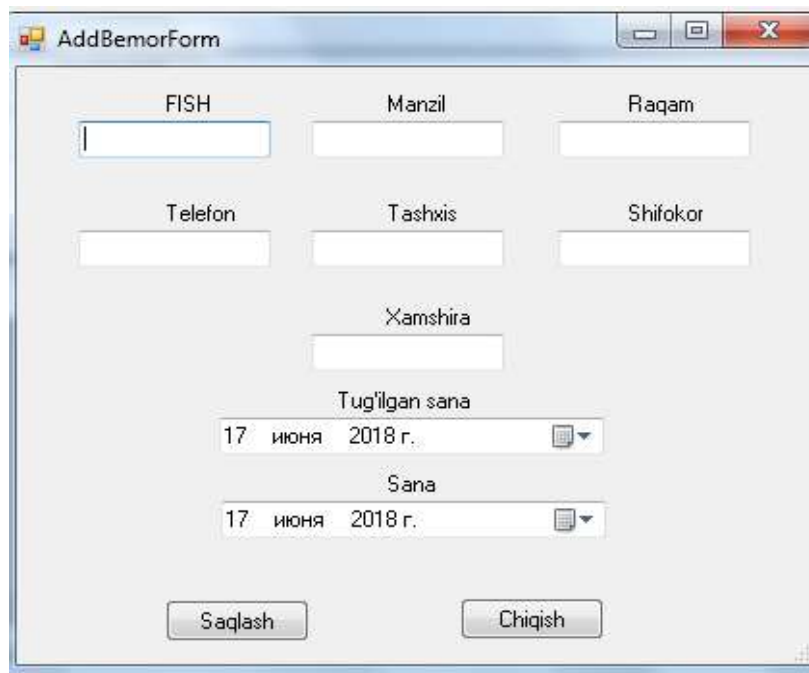
3.1. Dasturiy ta'minotni ishlatish va natija olish

ShifoxonaQayd.exe fayli ochilsa dasturiy ta'minotning bosh sahifasi quyidagi ko'rinishda ochiladi:



3.1-rasm. Dasturning bosh sahifasi

Bu oynadan qo'shish tugmasini bosish orqali bemorlar haqidagi ma'lumotlarni bazaga kiritishimiz mumkin bo'ladi:



3.2-rasm. Ma'lumotlar kiritish bo'limi

Bu bo'limga barcha ma'lumotlarni kiritib saqlash tugmasi bosilsa kiritilgan ma'lumotlar bazada saqlanadi.

Biror satrni belgilab O'zgartirish tugmasini bossak quyidagi o'zgartirish bo'limini ochishga muvafaq bo'lamiz:

The screenshot shows a window titled 'AddBemorForm' with a light blue border. Inside, there are several input fields for patient data. The fields are arranged in a grid-like fashion. At the bottom, there are two buttons: 'Saqlash' (Save) and 'Chiqish' (Exit).

FISH	Manzil	Raqam
Husaynov Nodir	Samarqand shahri	28
Telefon	Tashxis	Shifokor
+998902655645	Jigar serroz	Hamidov Rashid
Xamshira	Syfiyeva Sanobar	
Tug'ilgan sana	22 июля 1993 г.	
Sana	6 июня 2018 г.	

3.3-rasm. Ma'lumotlarni ozgartirish bo'limi

O'chirish tugmasi orqali biror keraksiz satrni ochirish imkoniyati mavjud.

The screenshot shows a window titled 'Form1' with a light blue border. At the top, there are three buttons: 'Qo'shish' (Add), 'O'zgartirish' (Edit), and 'O'chirish' (Delete). To the right of these buttons is a search bar labeled 'Shifoxona qaydnomalari' with the text 'Bemor FISH: Husaynov Nodir' and a 'Qidirish' (Search) button. Below the search bar is a table with columns for patient information. The first row is highlighted in blue, indicating it is selected. Below the table, there is a large grey rectangular area, likely a placeholder for a detailed view or a list of records.

ID	FISH	TUG'ILGAN SANA	MANZIL	RAQAM	TELEFON	SANA	TASHXIS	SHIFOKOR
6	Husaynov Nodir	22 июля 1993 г.	Samarqand shahri	28	+998902655645	6 июня 2018 г.	Jigar serroz	Hamidov Rashid

3.4-rasm. Qidiruv bo'limi.

Bu bo'limda biror bemorning familiyasi kiritiladi va qidirilish tugmasi bosiladi va bemor haqidagi ma'lumotlar hosil bo'ladi.

3.2. Shifoxona xodimlarini favqulodda vaziyat sharoitida himoyalash

Bemor va xodimlarni favqulodda vaziyat sharoitida himoyalash, shu hodisaning noxush oqibatiga yo'l quymaslik yoki uni maksimal darajada pasaytirish bo'yicha qo'llaniladigan choralar majmuaviy tadbirlardir. Bemor va xodimlarni himoyalashning samaradorligiga mavjud barcha vosita va usullar to'la ishlatilganda va xavfsizlikni ta'minlash tamoyillari to'laligicha anglab yetilgan taqdirdagina erishish mumkin.

Xavfsizlikni ta'minlash tamoillari qo'llanilishiga ko'ra uch guruhga bo'linadi:

1. Oldindan tayyorgarlik ko'rmoq (shaxsiy va jamoaviy), himoya vositalarini to'plash, ularni tayyor holda saqlash hamda aholini xavfli hududdan evakuatsiya qilish bo'yicha tadbirlarni tayyorlash va amalgam oshirish.

2. Differensiyallashgan yo'nalish tadbirlari hajmi va xarakteri xavfli va zaharli omillar manbaining turi va mahalliy sharoitlarga bog'liq.

3. Tadbirlar majmuasi FV oqibatlaridan himoyalaniish usullari va vositalaridan samarali foydalanish, zamonaviy texno-ijtimoiy muhitda hayotiy faoliyat xavfsizligini ta'minlash bo'yicha tadbirlarbilani birga olib borishni taqazo etadi.

FV larda aholini himoya qilishning asosiy usullari, aholini evakuatsiya qilish, himoya inshootlariga joylashtirish, shaxsiy himoya hamda tibbiy profilaktik vositalarini qo'llashdan iboratdir.

Siyosiy-harbiy nizolar ayniqsa, zamonaviy qurol yarog'lar bilan hal qilingan taqdirda qirg'indan saqlab qolish uchun eng samarali usullardan biri aholini himoya inshootlariga joylashtirishdir. Ayniqsa radiktiv va kimyoviy moddalardan

saqlanishda bu choraning samarasi yuqoridir. Himoya inshootlari – fizik, ximik va biologik xavfli va zararli omillardan himoyalash maqsadida maxsus qurilgan injenerlik inshootlaridir.

Bu inshootlar himoyalash xossalari bo'yicha pana joy va radiatsiyaga qarshi yashirinish uchun yopiq pana joylarga bo'linadi. Bular qurilish me'yorlariva qoidalari SNiP 2.01.51-90 bo'yicha loyihalanadi.

Shaxsiy himoya vositalari - odam organizmining ichki a'zolariga, terisiga va keyimiga radioktiv, zaharlovchi va bacterial vositalarning ta'siridan saqlaydi.

Shaxsiy himoya tibbiy vositalari FV larda jabrlangan aholiga tibbiy yordam ko'rsatish va profilaktik tadbirlar uchun mo'ljallanadi. Ular yordamida odam hayotini saqlab qolish, ko'riladigan zararni kamaytirish yoki oldini olish, ayrim xavfli yoki zararli omillarning kishi organizmiga ta'siriga chidamliligini oshiradi.

FV larda aholini hayotiy faoliyat xavfsizligini ta'minlashda qator tadbirlar avvaldan ko'rib qo'yish muhim rol o'ynaydi, xususan: aholini FV larga o'qitish, FV haqida vaqtida xabar berish, kimyoviy va bakterial razvedka ishlarini yo'lga qo'yish va amalga oshirish, hamda dozimetrik va laboratoriya nazoratlarini tashkil qilish, gaz xavfsizligiga qarshi, yong'inga qarshi, sanitariya-gigiyenik tadbirlarni o'tkazish, qutqaruv va boshqa zudlik bilan bajariladigan ishlar uchun moddiy boyliklar zaxirasini to'plash.

Favqulodda vaziyatlarda hayotiy faoliyat xavfsizligini ta'minlash-odam faoliyatining barcha jabhalarida sog'lig'ini va hayotini saqlashga qaratilgan tashkiliy, injener-texnik tadbirlar va vositalar majmuasidir.

Hayotiy faoliyat xavfsizligini ta'minlashga qaratilgan asosiy yunalishlar quyidagilardan iborat:

-FV larni va ularning oqibatlarini oldindan aytish;

-FV paydo bo'lishi ehtimolini yo'qotish yoki kamaytirish tadbirlarini rejalashtirish, hamda ular oqibatlarini ko'lamini kamaytirish;

-FVlarda xalq xo'jaligi ob'yektlari ishining muntazamligini, uzluksizligini ta'minlash;

-FV holatlariga aholini o'qitish;

-FV oqibatlarini to'gatish.

Ushbu yo'nalishlarning qisqacha mazmunlari quyidagichadir:

FV ni va ularning oqibatlarini oldindan aytish va baholash.

Bu-tabiiy ofatlar, avariya va falokatlar tufayli yuzaga kelgan vaziyatni taxminiy aniqlash va baholash usulidir.

Hozirgi vaqtda seysmik rayonlar, tuproq siljishi mumkin bo'lgan sel oqimlari yo'nalishlari, to'g'onlar buzilganda, suv toshqini bo'lganda, gazli zonalarda odamlarga va hududga zarar keltirishi mumkin bo'lgan hududlar chegaralari aniqlangan. Qulay ish sohasini tanlash hozirgi paytda yangi texnologik jarayon, kichik korxonalarining yangi mashina va uskunalari albatta maxsulot sifatligiga binoan bo'lishi, ergonomik ko'rsatgichlar bo'yicha belgilangan talablarga javob berishi kerak.

Ish joyida jihoz va uskunalarning noto'g'ri joylashtirilishi, ularning shakli, rangini noto'g'ri tanlanishi ishchini charchatadi, to'g'ri joylashganda esa aksincha, ish unumdorligini oshirish imkonini beradi. Uskunalar shunday joylashtirilishi lozimki, ishchi ulardan ortiqcha xarakatsiz, o'zini zo'riqtirmasdan osonlik bilan foydalansin. Ish joyining balandligi ham muhim ahamiyatga ega odatda uning qiymati 1000-1600 mm, orasida bo'lishi maqsadga muvofiqdir: Bundan tashqari, ko'zning imkoniyat doirasini ham hisobga olish kerak. Gorizont bo'yicha qurish burchaklari quyidagicha tanlanadi.

Mehnat jarayonida qattiq, suyuq va par holatidagi qismlardan va elektr ta'siridan odam tanasida sodir bo'ladigan jarohatlanish **-shikastlanish** deyiladi. Shikastlanish natijasida yarador bo'lish, suyak chiqishi va sinishi, kislota yoki issiqlik kuyish, elektr toki ta'sirida odam a'zolari faoliyatining buzilishi va boshqalar misol bo'la oladi. Shikastlanish – baxtsiz xodisa xisoblanadi. Baxtsiz xodisalar ishlab chiqarish bilan bo'g'langan yoki bo'g'lanmagan bo'lishi mumkin. Ishlab chiqarish bilan bo'g'langan baxtsiz xodisalarga xodimning ish joyida yoki shu korxonada bog'liq ishlarini bajarayotganda (masalan ish joyida tashqarida, lekin, asosiy ishga aloqador boshqa ishni bajarish paytida ishga boorish va qaytishdan, mehnat safariga boranda va boshqalar) olgan agar shikastlanishlar kiradi.

Agar nogironlik keltirib chiqaruvchi jarohatlanish zaharlanish, issiq urish, kuyish, sovuq urishi, cho'kish yoki avariya tufayli salomatlikning boshqacha shikastlanishi quyidagi hollarda sodir bo'lsa (huquqqa qarshi elektr toki yoki yashin urishi, shuningdek tabiiy ofatlar qattiq harakatlardan tashqari), nogironlik mehnatda jarohatlanish tufayli sodir bo'lgan deb hisoblanadi:

- A) Mehnat vazifalarini bajarishda (shu jumladan xizmat safari vaqtida), shuningdek korxonada yoki tashkilot manfaatlari yo'lida, hatto maxsus topshiriqsiz biron ish bajarganda;
- B) Korxonada, tashkilot xududida yoki boshqa ish joyida, ish vaqti davomida (belgilangan tanaffus vaqti ham shunga kiradi), shuningdek ish boshlanishi yoki tugashi oldidan, ishlab chiqarish qurollari, kiyim bosh va boshqalarni tartibga keltirish uchun zarur bo'lgan vaqt davomida;
- V) Ish vaqtida (belgilangan tanaffuslar ham shunga kiradi) korxonada tashkilot yoki boshqa ish joyi yaqinida, agar u yerlarda bo'lish ichki mehnat tarkibi qoidalariga zid bo'lmasa;
- G) Ishga borayotganda yoki ishdan qaytayotganda;

D) Shaxtali shaharcha xududida, shaxta-ekspeditsiya usuli bilan va tabiiy ofat vaqtida, smenali dam olishda bo'lgan xodim bilan yuz berganda;

Davlat yoki jamoat vazifalarini bajarishda, shuningdek O'zbekiston Respublikasi qonun hujjatlariga muvofiq belgilangan tartibda ro'yxatga olingan jamoat tashkilotlarining topshiriqlarini (garchi bu topshiriqlar asosiy ish bilan bog'liq bo'lmasa ham) bajarishda; Mehnat vazifalariga bajarish deganda yollanib ishlashga doir o'z xizmati vazifasini bajarish tushuniladi.

Baxtsiz xodisa-extiyotsizlikdanmi, ya'ni xodimning o'z aybi bilanmi yoki ishlab chiqarish jarayonini tashkil qilishda ma'muriyat tomonidan texnika xavfsizligiga yo' qo'yilganidanmi buni ahamiyati yo'q.

Safardagi baxtsiz xodisa natijasida yuzaga kelgan nogironlik agar baxtsiz hodisa mehnat vazifasini bajarish davrida yoki safarga jo'natilgan erga borish vaqtida, xizmat chaqiruvlarida yuz bergan bo'lsa, mehnatda jaraxotlanish hisoblanadi. Xodim ishga ketayotganda yoki undan qaytayotganda uning qaysi marshrut va transport turini tanlanishidan qat'iy nazar, yo'lda sodir bo'lgan baxtsiz hodisa ham mehnatda jarahotlanish hisoblandi.

Davlat va jamoat vazifalarini, mahalliy hokimiyat organlarining kasbiy yoki jamoat tashkilotlarining maxsus topshiriqlarini bajarishda bu topshiriqlar asosiy ish bilan bogliq-bog'liq emasligidan qat'iy nazar mehnatda jarahotlanish hisoblanadi.

Ishlab chiqarish mikroiklimining gigiyenik normalari. Ishlab chiqarish mikroiklimi normalari mehnat xafsizligi standartlari tizimi "Ish zonasi mikroiklimi" (GOST 12.1· 005-76) ga asosan belgilanadi. Ular gigiyenik, texnik va iqtisodiy negizlarga asoslangan.

Sanoat korxonalaridagi xonalar, yil fasllari va ish toifasiga qarab, ulardagi harorat, nisbiy namlik va havo harakatining ish joylari uchun ruxsat etilgan normalari belgilangan.

Ishlab chiqarish xonalari, kompyuter xonalari, ish joylardagi havoning harorati, nisbiy namligi va xarakat tezligining risoladagi normalari

Yil fasli	Ish toifalari	Havoning harorati, S	Nisbiy namlik %	Xarakat tezligi m/s
sovuq	Yengil – I	20-23	60-30	0,2
	O'rtacha og'irlikdagi – I a	18-20	60-40	0,2
	O'rtacha og'irlikdagi – I b	17-19	60-40	0,3
iliq	Og'ir – III	16-18	60-40	0,3
	Yengil – I	20-25	60-40	0,2
	O'rtacha og'irlikdagi – I a	21-23	60-40	0,3
	O'rtacha og'irlikdagi – I b	20-22	60-40	0,4
	Og'ir – II	18-21	60-40	0,5
	issiq	Yengil – I	20-30	60-30
o'rtacha og'irlikdagi – I a		20-30	60-30	0,4-0,5
o'rtacha og'irlikdagi – I b		20-30	60-30	0,5-0,7
Og'ir – III		20-30	60-30	0,5-1,0

Sanoat korxonalarini va kompyuter xonalarida yoritish. Yorug'lik inson xayoti faoliyati davomida juda muxim rol o'ynaydi. Shuning uchun ham sanoat korxonalarini, kompyuter bor honalarni rasional yoritish sifatli maxsulot ishlab chiqarishni ta'minlash bilan birga ishlab chiqarish sharoitini yaxshilaydi,

ishchilarni charchashdan saqlaydi va mehnat unumdorligini oshiradi. Oqilona yoritilgan zonalarda ishlayotgan ishchilarning kayfiyati yaxshi bo'ladi, buning natijasi baxtsiz xodisalar keskin kamayadi.

Bundan ko'rinib turibdidiki, kompyuter xonalarini yoritish faqatgina gigiyenik talab qo'yilmasdan, balki texnik-iqtisodiy talablar ham qo'yiladi.

Kompyuter xonalarida yoritishning mukammalligi sifat va son ko'rsatkichlari bilan tavsiflanadi. Son ko'rsatkichlari nur oqimi, yorug'lik kuchi, yorqinlik, nur qaytarish koeffitsiyentlari, yorug'lik kiradi.

Kompyuter xonalarini yoritishga qo'yilgan asosiy talablar.

1. Ish joylarini yoritish sanitariya – gigiyena normalari asosida ish toifalariga moslashgan bo'lishi kerak. Ish joylarini maksimal yoritish albatta ish sharoitini yaxshilashga olib keladi. Bundan ish olib borilayotgan ob'yektning kurinishi yaxshilanadi, natijada ish unumdorligi ortadi. Ba'zi bir aniq ishlarni bajarganda yoritilishni 50 lk dan 1000 lk gacha oshirish bilan ish unumi 25 foizga oshganligi ma'lum. Ko'z bilan ko'rib ishlash unchalik shart bo'lmagan ishlarni bajarganda ham yoritilishni 50 lk dan 300 lk ga oshirish ish unumini 5-7 foizga oshirgan.

2. Ish olib borilayotgan yuzaga va ko'zga ko'rinadigan atrof muxitga yorug'lik bir tekis tushadigan bo'lishi kerak. Chunki agar ish olib borilayotgan yuzada va atrof muhitda yaltiroq uchastkalar mavjud bo'lsa, unda ko'zning jimirlashi va ma'lum vaqt ko'nikishi kerak bo'ladi. Bu esa ko'zning tez charchashiga olib keladi.

3. Yoritish miqdori vaqt bo'yicha o'zgaras bo'lishi kerak. Yoritilishning ko'payib – kamayishi, agar u o'qtin-o'qtin ro'y beradigan bo'lsa, ko'zga zarar keltiradi, chunki ko'z yorug'lik o'zgarishlariga ko'nikishga to'g'ri keladi. Bu esa ko'zning tez charchashiga olib keladi. Yoritilishning o'zgarasligi muxim o'zgaras kuchlanishli manbalaridan foydalanish yo'li bilan erishish mumkin.

4. Yorug'lik qurilmalari qo'shimcha xavf va zararlar manbai bo'lmasligi kerak. Shuning uchun yoritish manbalari ajratadigan issiqlikni, tovush chiqarishni maksimal kamaytirish zarur.

5. Yoritish qurilmasi ishlatish uchun qulay, o'rnatish oson va iqtisodiy samarador bo'lishi kerak.

Yoritgichlar va ularni joylashtirish. Yorug'lik manbalarida yoritish armaturasi joylashadi va ular birgalikda yoritgichlar yoki chiroqlar deb ataladi. Yoritgichlar konstruksiyalariga qo'ydagi talablar qo'yiladi:

- 1) nur oqimining yo'nalishini ishchi yuzalar tomonga qayta taqsimlanishini ta'minlash;
- 2) lampaning nur tarqatayotgan yuzalarining yaraqlab ko'zga ta'sir ko'rsatishidan muhofaza qilish;
- 3) lampani har xil sanoat iflosliklari va changdan himoya qilish;
- 4) lampani portlash, o't olish xavflaridan muhofaza qilish.

Shovqindan saqlanish. Hozirgi zamon texnika taraqqiyoti davrida sanoat korxonalarida shovqinga qarshi kurash masalalari muhim muammolar qatoriga kiradi. Bu – asosan mashinasozlik sanoati, transport vositalarini ishlatishda, energetika sanoatida va sanoat korxonalarida juda jiddiy muammo bo'lib turibdi. Shovqinning oqibatlarini ma'lum. Katta shovqin ta'sirida insonning asab tizimlari charchab, eshitish faoliyati susayib ketadi. Shuning uchun ham sanoat korxonalarida shovqinni kamaytirish chora-tadbirlarini belgilash inson salomatligini saqlashdek juda muhim ijtimoiy ahamiyatga egadir.

Shovqinga qarshi kurash chora-tadbirlari qo'yidagi usullarda olib boriladi:

- shovqinni ajralib chiqayotgan manbaida kamaytirish;
- shovqinning tarqalish yo'nalishini o'zgartirish;

- sanoat korxonalarini va kompyuter xonalarini oqilona rejalashtirish;
- sanoat korxonalarini va kompyuter xonalariga akustik ishlov berish;
- shovqinni tarqalish yo'lida kamaytirish.

Elektr xavfsizligini ta'minlash. Sanoat korxonalarida o'rnatilgan kompyuter, yordamchi mashina-mexanizmlar, shkaflar, boshqarish pultrlari hamda elektr asboblari va boshqa jihozlarning hammasi, agar ularni ishlatish uchun foydalanadigan elektr quvvati 42 V dan yuqori kuchlanishga ega bo'lsa, albatta yerga ulangan bo'lishi shart. Buning uchun har qanday kompyuter xonalarini yerga ulash vositalari qulay, oson va yengil bajariladigan bo'lishi kerak. Bu vositalar yerga yetarli darajada puxta ulangan yoki nolga ulangan bo'lishi shart. Bu vositalarda yerga ulash simlarini osongina mahkamlash mumkin bo'lgan vintsimon qurilmalar yoki qiskich bo'lishi kerak.

Yong'inga qarshi ishlatiladigan texnik qurilmalar. Sanoat korxonalarini binolarini yong'indan muhofaza qilish uchun ishlatiladigan asosiy texnik qurilmalar GOST 12,4009-75 asosida aniqlanadi. Har qanday yong'inni o'chirishda yong'inning kuchayishiga olib kelayotgan omillarni va sharoitni aniqlash muximdir. Bunda yonishning davom etishini to'xtatuvchi sharoit yaratish katta rol o'ynaydi. Yong'inni o'chirish paytida qattiq jismlar yonganda yong'inning tezligi 4m/min, suyuqliklar yuzasi bo'yicha esa 30 m/min bo'lishini hisobga olish kerak.

Yong'in (o't) o'chirish vositalari va usullari. O't o'chirish usullari qo'ydagicha bo'lishi mumkin:

- yonayotgan zonani ko'p miqdorda issiqlik yutuvchi materiallar yordamida sovitish;
- yonayotgan materiallarni atmosfera havosidan ajratib qo'yish;
- yonayotgan zonaga kirayotgan kislorod miqdorini kamaytirish;

- maxsus kimyoviy vositalarni qo'llash.

O't o'chirish vositalari sifatida, suv bug'lari, kimyoviy va mexanik ko'piklar, inert va yonmaydigan gazlar, qattiq kukunsimon materiallar va aralashmalardan foydalaniladi[17].

Suv bilan o'chirish. Suv eng ko'p tarqalgan arzon va shuning bilan birga deyarli hamma yerda mavjud bo'lgan o't o'chirish vositasi bo'lib, suv bilan har qanday masshtabdagi yong'inlarni o'chirish mumkin.

Bug' bilan o'chirish. Bug' bilan o'chirishning asosiy mohiyati shuki, xonalarga yuborilgan bug' kislorodga boy havoni siqib chiqarib, uning o'rnini egallaydi. Bug'ning o't o'chirish samaradorligi uning ma'lum bir xonaga yuborilgan miqdoriga bog'lik bo'ladi.

Yong'inga qarshi suv ta'minoti. Odatda o't o'chirish uchun ishlatiladigan suv katta bosim ostida kuchli oqim sifatida alanganayotgan joyga yuboriladi. Buning uchun yetarli bo'lgan bosimni shahar sharoitiga umumiy vodoprovod tarmoqlari orqali xosil qilinadi. Yoki bo'lmasa ba'zi bir yerlarda maxsus tayorlangan xovuz va idishlardan foydalanilinish mumkin.

Isitish va shamollatish tizimlari. Sanoat korxonalarida odatda markaziy isitish tizimi orqali isitiladi. Shuning uchun bunday tizimlarning yong'in xafi bo'lgan uchastkalarida isitish radiatorlarining murakkab turlaridan foydalanish tavsiya etilmaydi. Chunki changlar truba va radiator ustki qismlarida yig'ilib qolishi, isitish natijasida qizib, yong'in chiqarish xafini kuchaytiradi.

Xonalari markaziy usulda isitilgan sanoat korxonalarida, havo asosan koloriferlarda isitilib, undan keyin korxonalariga yuboriladi. Bunday xollarda isitilgan havoning harorati 66⁰S dan oshmasligi kerak. Ammo bunday isitish tizimida havo kanallar orqali xonalarga tarqatilgani sababli yong'in bo'lgan

taqdirda bu kanallar orqali alanga tarqatilishi mumkin. Chunki alanga va tutun butun bino bo'ylab tarqalib ketishi tufayli yong'in xavfi kuchayadi[6].

XULOSA

Ushbu bitiruv malakaviy ishida quyidagilar amalga oshirildi:

- Shifoxonalarda bemorlarni ro'yxatga olishni avtomatlashtiruvchi dasturiy vosita yaratish masalasi qo'yildi;
- Ro'yxatga olishni avtomatlashtirishning ba'zi usullari haqida umumiy ma'lumotlar berildi;
- Ruyxatga olish uchun kerakli ma'lumotla keltirildi;
- Avtomatlashtirilgan axborot tizimlari haqida ma'lumotlar keltirildi.
- Shifoxonaga oid ma'lumotlar bazasini yaratish bosqichlari ishlab chiqildi;
- Ma'lumotlarni tashkil qilish, ma'lumotlarni mantiqiy tashkil etish, fayllar modeli haqida tushunchalar keltirildi
- Ma'lumotlarni tashkil qilishning tarmoqli va iyerarxik modellari bo'yicha tushunchalar berildi.
- C++ Builder dasturida ilovalar bilan ishlash bo'yich ma'lumotlar berildi.
- Shifoxona xodimlarini favqulodda vaziyat sharoitida himoyalash haqida ma'lumotlar keltirib o'tildi.
- Dasturda ma'lumotlar bazasini hosil qilish jadvallarini yaratishda SQL bilan bog'landi.
- Dasturda qidirish protseduralaridan foydalanish imkoniyatlari yaratildi.

Xulosa qilib aytish mumkinki, mavzu bo'yicha bajarilgan ishlar Respublikamizda olib borilayotgan islohotlarni amalga oshirishdagi oldinga qo'yilgan bir qadam hisoblanib, uning ahamiyati hayotda o'z o'rniga egadir.

Foydalanilgan adabiyotlar ro'yxati

1. Каримов И.А. Жаҳон молиявий-иқтисодий инқирози, Ўзбекистон шароитида уни бартараф этишнинг йўллари ва чоралари.-Т.: Ўзбекистон, 2009. – 56 б.
2. Каримов И.А.«Мамлакатимизни модернизасия қилиш ва кучли фуқаролик жамияти барпо этиш – устувор мақсадимиздир» ҳамда «Асосий вазифамиз – Ватанимиз тараққиёти ва халқимиз фаровонлигини янада юксалтиришдир» номли маърузаларини ўрганиш бўйича Ўқув-услубий мажмуа. – Тошкент: Иқтисодиёт. - 2010. – 340 бет.
3. Каримов И.А. Юксак маънавият энгилмас куч. –Тошкент: «Маънавият», 2008. -176 б.
4. Ўзбекистон Республикаси Президентининг қарори. ПҚ-3987.«Давлат ҳокимияти ва бошқарув органлари фаолиятининг очиқлиги тўғрисида». 2012 йил 20 март.
5. Ўзбекистон Республикаси Президентининг қарори. ПҚ-1730.« Замонавий ахборот-коммуникация технологияларини жорий этиш ва ривожлантириш чора-тадбирлари тўғрисидаги ». 2012 йил 12 март.
6. Узбекистон Республикаси Вазирлар Махкамасининг қарори “Ўзбекистон Республикаси фавкулотда вазиятларда уларни олдини олиш ва ҳаракат қилиш давлат тизими тугрисида”,23.12.1997, №558.
7. Peng C.K., Buldyrev S.V., Havlin S., Simons M., Stanley H.E., Goldberger A.L., Mosaic organization of DNA nucleotides. // Phys Rev E. 1994.— 49 (2).— P. 1685–1689.
8. Қосимов С.С. Ахборот технологиялари. Ўқув қўлланма. – Т.: Алоқачи нашриёти, 2006. -370 б.
9. Кузнецов М.В. MySQL 5 (+CD). СПб.: BHV-Санкт-Петербург, 2006.-1024 с.

10. Храмцов П.Б., Брик С.А., Русак А.М., Сурин А.И. Основы web-технологий Интернет-университет информационных технологий - ИНТУИТ.ру, 2003.
11. Кузнецов М.В. MySQL 5 (+CD). СПб.: BHV-Санкт-Петербург, 2006.-1024 с.
12. Фролов А. В., Фролов Г. В. «Базы данных в Интернете: практическое руководство по созданию Web-приложений с базами данных».- М.: Издательско-торговый дом «Русская Редакция», 2000. 448 с.
13. Чуи К. Введение в вэйвлеты. — М.: Мир, 2001. — 416 с.
14. Культин Н. Delphi3 Программирование на Object Pascal, BHV-Санкт-Петербург, 1998г. – 304с.,ил.
15. Дарахвалидзе А. Марков «Delphi 4 МАСТЕР», Москва, 2000г.
16. Хоманенко А. «Самоучитель Delphi 5» Киев, 1999г.
17. A. Qudratov, T. G'aniyev va boshqalar. Nayot faoliyati xavfsizligi. Toshkent: Aloqachi, 2005. 186 bet.
18. Хаймен М. Delphi 6 для "чайников". - К.: "Диалектика", 1995. - 416 с.
19. Хаймен М. Delphi 7 для "чайников". - К.: "Диалектика", 1998. - 414 с.
20. <http://www.megabook.uz>
21. <http://www.sdelaysayt.ru>
22. <http://www.yiiframework.com/>
23. <http://fractals.narod.ru/intro.htm>
24. WWW.ziyonet.uz
25. WWW.google.uz

ILOVA

```
using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SQLite;

using System.IO;

namespace ShifoxonaQayd

{

    public partial class Form1 : Form

    {

        private string dbFileName;

        private SQLiteConnection sqlConn;

        private SQLiteCommand sqlCmd;

        bool sqlNull = false;

        HisobKitob hk = new HisobKitob();

        public Form1()
```

```

{
    InitializeComponent();
}

private void Form1_Load(object sender, EventArgs e)
{
    sqlConn = new SQLiteConnection();
    sqlCmd = new SQLiteCommand();
    dbFileName = "shq.db";
    sqlConnect();
}

private void sqlConnect()
{
    /// Bazaga ulanish qismi
    /// Agar baza mavjud bo`lmasa u bazani yaratadi
    if (!File.Exists(dbFileName))
    {
        SQLiteConnection.CreateFile(dbFileName);
        sqlNull = true;
    }
    try
    {

```

```

        sqlConn = new SQLiteConnection("Data Source=" + dbFileName +
";Version=3;");

        sqlConn.Open();

        sqlCmd.Connection = sqlConn;

        if (sqlNull)

        {

            /// Baza bush bulgan bulsa client jadvalini yaratadi;

            sqlCmd.CommandText = "CREATE TABLE `bemorlar` (" +

                "id`    INTEGER    NOT    NULL    PRIMARY    KEY

AUTOINCREMENT," +

                "`fish`    TEXT NOT NULL," +

                "`tsana`    TEXT NOT NULL," +

                "`manzil`    TEXT NOT NULL," +

                "`raqam`    TEXT NOT NULL," +

                "`telefon`    TEXT NOT NULL," +

                "`sana`    TEXT NOT NULL," +

                "`tashxis`    TEXT NOT NULL," +

                "`shifokor`    TEXT NOT NULL," +

                "`xamshira`    TEXT NOT NULL); ";

            sqlCmd.ExecuteNonQuery();

        }

```

```

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void readAll()
{
    DataTable dt = new DataTable();

    string sqlQuery;

    try
    {
        sqlQuery = "SELECT * FROM bemorlar";

        SQLiteDataAdapter adapter = new SQLiteDataAdapter(sqlQuery,
sqlConn);

        adapter.Fill(dt);

        if (dt.Rows.Count > 0)
        {
            dataGridView1.Rows.Clear();

            for (int i = 0; i < dt.Rows.Count; i++)
            {

```

```

        dataGridView1.Rows.Add(dt.Rows[i].ItemArray);
//MessageBox.Show(dataGridView1.CurrentRow.Cells[14].Value.ToString());
    }
}
else
{
    //MessageBox.Show("Bazada malumotlar mavjudmas");
    dataGridView1.Rows.Clear();
}
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
for (int i = 0; i < dataGridView1.Rows.Count - 1; i++)
{
    /// 1 11 12
    //string tmp1 =
    //MessageBox.Show(dataGridView1.Rows[i].Cells[1].Value.ToString());
    dataGridView1.Rows[i].Cells[1].Value =
hk.converterDK(dataGridView1.Rows[i].Cells[1].Value.ToString());

```

```

        dataGridView1.Rows[i].Cells[3].Value =
hk.converterDK(dataGridView1.Rows[i].Cells[3].Value.ToString());

        dataGridView1.Rows[i].Cells[5].Value =
hk.converterDK(dataGridView1.Rows[i].Cells[5].Value.ToString());

        dataGridView1.Rows[i].Cells[7].Value =
hk.converterDK(dataGridView1.Rows[i].Cells[7].Value.ToString());

        dataGridView1.Rows[i].Cells[8].Value =
hk.converterDK(dataGridView1.Rows[i].Cells[8].Value.ToString());

        dataGridView1.Rows[i].Cells[9].Value =
hk.converterDK(dataGridView1.Rows[i].Cells[9].Value.ToString());
    }
}

private void addBemor()
{
    AddBemorForm addCl = new AddBemorForm();
    if (addCl.ShowDialog() == DialogResult.OK)
    {
        try
        {
            addCl.Fish = hk.converterK(addCl.Fish);

            addCl.Manzil = hk.converterK(addCl.Manzil);

            addCl.Telefon = hk.converterK(addCl.Telefon);
        }
    }
}

```

```

addCl.Tashxis = hk.converterK(addCl.Tashxis);

addCl.Shifokor = hk.converterK(addCl.Shifokor);

addCl.Xamshira = hk.converterK(addCl.Xamshira);

//MessageBox.Show(addCl.Fish);

sqlCmd.CommandText = "INSERT INTO `bemorlar`(" +

    "`fish`," +

    "`tsana`," +

    "`manzil`," +

    "`raqam`," +

    "`telefon`," +

    "`sana`," +

    "`tashxis`," +

    "`shifokor`," +

    "`xamshira`)" +

    "VALUES (" +

    addCl.Fish + "," +

    addCl.TSana + "," +

    addCl.Manzil + "," +

    addCl.Raqam + "," +

    addCl.Telefon + "," +

    addCl.Sana + "," +

```

```

        addCl.Tashxis + "," +
        addCl.Shifokor + "," +
        addCl.Xamshira + ");";

//MessageBox.Show(sqlCmd.CommandText.ToString());

//MessageBox.Show(hk.converterK(addCl.Fish));

sqlCmd.ExecuteNonQuery();

}

catch (Exception ex)

{

    MessageBox.Show(ex.Message);

}

}

readAll();

}

private void changeBemor()

{

    AddBemorForm addCl = new AddBemorForm();

    string curId = dataGridView1.CurrentRow.Cells[0].Value.ToString();

    addCl.Fish = dataGridView1.CurrentRow.Cells[1].Value.ToString();

    addCl.TSana = dataGridView1.CurrentRow.Cells[2].Value.ToString();

    addCl.Manzil = dataGridView1.CurrentRow.Cells[3].Value.ToString();

```

```

addCl.Raqam = dataGridView1.CurrentRow.Cells[4].Value.ToString();
addCl.Telefon = dataGridView1.CurrentRow.Cells[5].Value.ToString();
addCl.Sana = dataGridView1.CurrentRow.Cells[6].Value.ToString();
addCl.Tashxis = dataGridView1.CurrentRow.Cells[7].Value.ToString();
addCl.Shifokor = dataGridView1.CurrentRow.Cells[8].Value.ToString();
addCl.Xamshira = dataGridView1.CurrentRow.Cells[9].Value.ToString();
//addCl.Komm = dataGridView1.CurrentRow.Cells[17].Value.ToString();
addCl.syncData();

//if (vf.ShowDialog() == DialogResult.OK)

//{

if (addCl.ShowDialog() == DialogResult.OK)
{
    addCl.Fish = hk.converterK(addCl.Fish);
    addCl.Manzil = hk.converterK(addCl.Manzil);
    addCl.Telefon = hk.converterK(addCl.Telefon);
    addCl.Tashxis = hk.converterK(addCl.Tashxis);
    addCl.Shifokor = hk.converterK(addCl.Shifokor);
    addCl.Xamshira = hk.converterK(addCl.Xamshira);

    try
    {
        sqlCmd.CommandText = "UPDATE `bemorlar` SET " +

```

```

        "`id`=" + curId +
        ",`fish`=" + addCl.Fish + "" +
        ",`tsana`=" + addCl.TSana + "" +
        ",`manzil`=" + addCl.Manzil + "" +
        ",`raqam`=" + addCl.Raqam + "" +
        ",`telefon`=" + addCl.Telefon + "" +
        ",`sana`=" + addCl.Sana + "" +
        ",`tashxis`=" + addCl.Tashxis + "" +
        ",`shifokor`=" + addCl.Shifokor + "" +
        ",`xamshira`=" + addCl.Xamshira + "" +
        " WHERE `id`=" + curId + ";";

//MessageBox.Show(sqlCmd.CommandText.ToString());

sqlCmd.ExecuteNonQuery();
}

catch (Exception ex)

{

    MessageBox.Show(ex.Message);

}

}

readAll();

//}

```

```

//else

// MessageBox.Show("Buyruq tasdiqlanmadi");
}

private void delBemor()
{
    string curId = dataGridView1.CurrentRow.Cells[0].Value.ToString();

    ///Uchirish

    try
    {
        sqlCmd.CommandText = "DELETE FROM `bemorlar` WHERE " +
            "`id`=" + curId + ";";

        sqlCmd.ExecuteNonQuery();

    }

    catch (Exception)
    {
        MessageBox.Show("Klient tanlanmagan");

    }

    readAll();

    //}

//else

// MessageBox.Show("Buyruq tasdiqlanmadi");

```

```

    }

private void fishSearch()
{
    if (textBox1.Text.Length < 1)
    {
        readAll();
    }
    else
    {
        DataTable dt = new DataTable();
        string sqlQuery;
        try
        {
            sqlQuery = "SELECT * FROM bemorlar WHERE fish=" +
hk.converterK(textBox1.Text) + """;

            SQLiteDataAdapter adapter = new SQLiteDataAdapter(sqlQuery,
sqlConn);

            adapter.Fill(dt);

            if (dt.Rows.Count > 0)
            {
                dataGridView1.Rows.Clear();
            }
        }
    }
}

```

```

        for (int i = 0; i < dt.Rows.Count; i++)
        {
            dataGridView1.Rows.Add(dt.Rows[i].ItemArray);
//MessageBox.Show(dataGridView1.CurrentRow.Cells[14].Value.ToString());
        }
    }
else
{
    //MessageBox.Show("Bazada malumotlar mavjudmas");
    dataGridView1.Rows.Clear();
}
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
for (int i = 0; i < dataGridView1.Rows.Count - 1; i++)
{
    /// 1 11 12
    //string tmp1 =
//MessageBox.Show(dataGridView1.Rows[i].Cells[1].Value.ToString());

```

```

        dataGridView1.Rows[i].Cells[1].Value =
hk.converterDK(dataGridView1.Rows[i].Cells[1].Value.ToString());

        dataGridView1.Rows[i].Cells[3].Value =
hk.converterDK(dataGridView1.Rows[i].Cells[3].Value.ToString());

        dataGridView1.Rows[i].Cells[5].Value =
hk.converterDK(dataGridView1.Rows[i].Cells[5].Value.ToString());

        dataGridView1.Rows[i].Cells[7].Value =
hk.converterDK(dataGridView1.Rows[i].Cells[7].Value.ToString());

        dataGridView1.Rows[i].Cells[8].Value =
hk.converterDK(dataGridView1.Rows[i].Cells[8].Value.ToString());

        dataGridView1.Rows[i].Cells[9].Value =
hk.converterDK(dataGridView1.Rows[i].Cells[9].Value.ToString());
    }
}

private void Form1_Activated(object sender, EventArgs e)
{
    readAll();
}

private void btnAdd_Click(object sender, EventArgs e)
{
    addBemor();
}

```

```

    }

    private void btnChange_Click(object sender, EventArgs e)
    {
        changeBemor();
    }

    private void btnDel_Click(object sender, EventArgs e)
    {
        delBemor();
    }

    private void btnSearch_Click(object sender, EventArgs e)
    {
        fishSearch();
    }
}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

```

```

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

using System.Data.SQLite;

namespace ShifoxonaQayd
{
    class HisobKitob
    {
        private string dbFileName;

        private SQLiteConnection sqlConn;

        private SQLiteCommand sqlCmd;

        /*public int dateDifToDay(DateTimePicker startDtp, DateTimePicker
endDtp)
        {
            int startDays, endDays, difDays;

            startDays = (365 * (startDtp.Value.Year - 1) +
startDtp.Value.DayOfYear);

            endDays = (365 * (endDtp.Value.Year - 1) + endDtp.Value.DayOfYear);

            difDays = endDays - startDays;

            return difDays;

```

```

}*/

private void sqlConnect()
{
    /// Bazaga ulanish qismi

    sqlConn = new SQLiteConnection();

    sqlCmd = new SQLiteCommand();

    dbFileName = "lesaDB.db";

    if (!File.Exists(dbFileName))
    {
        SQLiteConnection.CreateFile(dbFileName);
    }

    try
    {
        sqlConn = new SQLiteConnection("Data Source=" + dbFileName +
";Version=3;");

        sqlConn.Open();

        sqlCmd.Connection = sqlConn;

        //MessageBox.Show("Xush kelibsiz");
    }

    catch (Exception ex)
    {

```

```

        MessageBox.Show(ex.Message);
    }
}

public string converterK(string text)
{
    string temp = "";

    int n = text.Length;

    for (int i = 0; i < n; i++)
    {
        if (text[i] == "\\")
        {
            temp = temp + "\\\"";

            //n++;
        }
        else
        {
            if (text[i] == "'")
            {
                temp = temp + "\\';

                //n++;
            }
        }
    }
}

```

```

        temp = temp + text[i];
    }
}

return temp;
}

public string converterDK(string text)
{
    string temp = "";
    int n = text.Length;
    for (int i = 0; i < n; i++)
    {
        if (text[i] == '\\' && text[i + 1] == '^')
        {
            continue;
        }
        else
        {
            temp = temp + text[i];
        }
    }
    return temp;
}

```

```

}

//////////Probelll

/*

public string converterP(string text)

{

    string temp = "", tmp1 = "";

    int n = text.Length, t = 0;

    for (int i = n - 1; i >= 0; i--)

    {

        if (t % 3 == 0 && t != 0)

        {

            temp = temp + " ";

            //n++;

        }

        temp = temp + text[i];

        t++;

    }

    n = temp.Length;

    ///Reverse

    for (int i = n - 1; i >= 0; i--)

    {

```

```

        tmp1 = tmp1 + temp[i];
    }

    //MessageBox.Show(tmp1);

    return tmp1;
}

public string converterDP(string text)
{
    string temp = "", tmp1 = "";

    int n = text.Length;

    for (int i = n - 1; i >= 0; i--)
    {
        if (text[i] == ' ')
        {
            continue;
        }

        else
        {
            temp = temp + text[i];
        }
    }

    n = temp.Length;

```

```

    for (int i = n - 1; i >= 0; i--)
    {
        tmp1 = tmp1 + temp[i];
    }

    return tmp1;

}*/

}

}

namespace ShifoxonaQayd
{
    partial class AddBemorForm
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>

        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>

        /// <param name="disposing">true if managed resources should be disposed;
        otherwise, false.</param>

```

```

protected override void Dispose(bool disposing)
{
    if (disposing && (components != null))
    {
        components.Dispose();
    }
    base.Dispose(disposing);
}

#region Windows Form Designer generated code

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>

private void InitializeComponent()
{
    this.tbFish = new System.Windows.Forms.TextBox();
    this.tbManzil = new System.Windows.Forms.TextBox();
    this.tbRaqam = new System.Windows.Forms.TextBox();
    this.tbTelefon = new System.Windows.Forms.TextBox();
    this.tbTashxis = new System.Windows.Forms.TextBox();
    this.tbShifokor = new System.Windows.Forms.TextBox();
}

```

```
this.tbXamshira = new System.Windows.Forms.TextBox();

this.label1 = new System.Windows.Forms.Label();

this.label2 = new System.Windows.Forms.Label();

this.label3 = new System.Windows.Forms.Label();

this.label4 = new System.Windows.Forms.Label();

this.label5 = new System.Windows.Forms.Label();

this.label6 = new System.Windows.Forms.Label();

this.label7 = new System.Windows.Forms.Label();

this.label8 = new System.Windows.Forms.Label();

this.lSana = new System.Windows.Forms.Label();

this.dtpTSana = new System.Windows.Forms.DateTimePicker();

this.dtpSana = new System.Windows.Forms.DateTimePicker();

this.btnSave = new System.Windows.Forms.Button();

this.btnExit = new System.Windows.Forms.Button();

this.SuspendLayout();

//

// tbFish

//

this.tbFish.Location = new System.Drawing.Point(32, 29);

this.tbFish.Name = "tbFish";

this.tbFish.Size = new System.Drawing.Size(100, 20);
```

```
this.tbFish.TabIndex = 0;

//

// tbManzil

//

this.tbManzil.Location = new System.Drawing.Point(153, 29);

this.tbManzil.Name = "tbManzil";

this.tbManzil.Size = new System.Drawing.Size(100, 20);

this.tbManzil.TabIndex = 1;

//

// tbRaqqam

//

this.tbRaqqam.Location = new System.Drawing.Point(281, 29);

this.tbRaqqam.Name = "tbRaqqam";

this.tbRaqqam.Size = new System.Drawing.Size(100, 20);

this.tbRaqqam.TabIndex = 2;

//

// tbTelefon

//

this.tbTelefon.Location = new System.Drawing.Point(32, 88);

this.tbTelefon.Name = "tbTelefon";

this.tbTelefon.Size = new System.Drawing.Size(100, 20);
```

```
this.tbTelefon.TabIndex = 3;

//

// tbTashxis

//

this.tbTashxis.Location = new System.Drawing.Point(153, 88);

this.tbTashxis.Name = "tbTashxis";

this.tbTashxis.Size = new System.Drawing.Size(100, 20);

this.tbTashxis.TabIndex = 4;

//

// tbShifokor

//

this.tbShifokor.Location = new System.Drawing.Point(281, 88);

this.tbShifokor.Name = "tbShifokor";

this.tbShifokor.Size = new System.Drawing.Size(100, 20);

this.tbShifokor.TabIndex = 5;

//

// tbXamshira

//

this.tbXamshira.Location = new System.Drawing.Point(153, 144);

this.tbXamshira.Name = "tbXamshira";

this.tbXamshira.Size = new System.Drawing.Size(100, 20);
```

```
this.tbXamshira.TabIndex = 6;

//

// label1

//

this.label1.AutoSize = true;

this.label1.Location = new System.Drawing.Point(74, 13);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(31, 13);

this.label1.TabIndex = 7;

this.label1.Text = "FISH";

//

// label2

//

this.label2.AutoSize = true;

this.label2.Location = new System.Drawing.Point(189, 13);

this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(37, 13);

this.label2.TabIndex = 8;

this.label2.Text = "Manzil";

//

// label3
```

```
//  
  
this.label3.AutoSize = true;  
  
this.label3.Location = new System.Drawing.Point(314, 13);  
  
this.label3.Name = "label3";  
  
this.label3.Size = new System.Drawing.Size(41, 13);  
  
this.label3.TabIndex = 9;  
  
this.label3.Text = "Raqqam";  
  
//  
  
// label4  
  
//  
  
this.label4.AutoSize = true;  
  
this.label4.Location = new System.Drawing.Point(74, 72);  
  
this.label4.Name = "label4";  
  
this.label4.Size = new System.Drawing.Size(43, 13);  
  
this.label4.TabIndex = 10;  
  
this.label4.Text = "Telefon";  
  
//  
  
// label5  
  
//  
  
this.label5.AutoSize = true;  
  
this.label5.Location = new System.Drawing.Point(189, 72);
```

```
this.label5.Name = "label5";

this.label5.Size = new System.Drawing.Size(43, 13);

this.label5.TabIndex = 11;

this.label5.Text = "Tashxis";

//

// label6

//

this.label6.AutoSize = true;

this.label6.Location = new System.Drawing.Point(314, 72);

this.label6.Name = "label6";

this.label6.Size = new System.Drawing.Size(46, 13);

this.label6.TabIndex = 12;

this.label6.Text = "Shifokor";

//

// label7

//

this.label7.AutoSize = true;

this.label7.Location = new System.Drawing.Point(189, 128);

this.label7.Name = "label7";

this.label7.Size = new System.Drawing.Size(50, 13);

this.label7.TabIndex = 13;
```

```
this.label7.Text = "Xamshira";  
  
//  
  
// label8  
  
//  
  
this.label8.AutoSize = true;  
  
this.label8.Location = new System.Drawing.Point(163, 173);  
  
this.label8.Name = "label8";  
  
this.label8.Size = new System.Drawing.Size(76, 13);  
  
this.label8.TabIndex = 14;  
  
this.label8.Text = "Tug\`ilgan sana";  
  
//  
  
// lSana  
  
//  
  
this.lSana.AutoSize = true;  
  
this.lSana.Location = new System.Drawing.Point(189, 218);  
  
this.lSana.Name = "lSana";  
  
this.lSana.Size = new System.Drawing.Size(32, 13);  
  
this.lSana.TabIndex = 15;  
  
this.lSana.Text = "Sana";  
  
//  
  
// dtpTSana
```

```
//  
  
this.dtpTSana.Location = new System.Drawing.Point(105, 189);  
  
this.dtpTSana.Name = "dtpTSana";  
  
this.dtpTSana.Size = new System.Drawing.Size(200, 20);  
  
this.dtpTSana.TabIndex = 16;  
  
//  
  
// dtpSana  
  
//  
  
this.dtpSana.Location = new System.Drawing.Point(106, 234);  
  
this.dtpSana.Name = "dtpSana";  
  
this.dtpSana.Size = new System.Drawing.Size(200, 20);  
  
this.dtpSana.TabIndex = 17;  
  
//  
  
// btnSave  
  
//  
  
this.btnSave.Location = new System.Drawing.Point(77, 287);  
  
this.btnSave.Name = "btnSave";  
  
this.btnSave.Size = new System.Drawing.Size(75, 23);  
  
this.btnSave.TabIndex = 18;  
  
this.btnSave.Text = "Saqlash";  
  
this.btnSave.UseVisualStyleBackColor = true;
```

```
this.btnSave.Click += new System.EventHandler(this.btnSave_Click);  
  
//  
  
// btnExit  
  
//  
  
this.btnExit.Location = new System.Drawing.Point(230, 286);  
  
this.btnExit.Name = "btnExit";  
  
this.btnExit.Size = new System.Drawing.Size(75, 23);  
  
this.btnExit.TabIndex = 19;  
  
this.btnExit.Text = "Chiqish";  
  
this.btnExit.UseVisualStyleBackColor = true;  
  
this.btnExit.Click += new System.EventHandler(this.btnExit_Click);  
  
//  
  
// AddBemorForm  
  
//  
  
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);  
  
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;  
  
this.ClientSize = new System.Drawing.Size(413, 322);  
  
this.Controls.Add(this.btnExit);  
  
this.Controls.Add(this.btnSave);  
  
this.Controls.Add(this.dtpSana);  
  
this.Controls.Add(this.dtpTSana);
```

```
this.Controls.Add(this.lSana);  
this.Controls.Add(this.label8);  
this.Controls.Add(this.label7);  
this.Controls.Add(this.label6);  
this.Controls.Add(this.label5);  
this.Controls.Add(this.label4);  
this.Controls.Add(this.label3);  
this.Controls.Add(this.label2);  
this.Controls.Add(this.label1);  
this.Controls.Add(this.tbXamshira);  
this.Controls.Add(this.tbShifokor);  
this.Controls.Add(this.tbTashxis);  
this.Controls.Add(this.tbTelefon);  
this.Controls.Add(this.tbRaqam);  
this.Controls.Add(this.tbManzil);  
this.Controls.Add(this.tbFish);  
this.Name = "AddBemorForm";  
this.Text = "AddBemorForm";  
this.ResumeLayout(false);  
this.PerformLayout();  
}
```

```
#endregion

private System.Windows.Forms.TextBox tbFish;

private System.Windows.Forms.TextBox tbManzil;

private System.Windows.Forms.TextBox tbRaqam;

private System.Windows.Forms.TextBox tbTelefon;

private System.Windows.Forms.TextBox tbTashxis;

private System.Windows.Forms.TextBox tbShifokor;

private System.Windows.Forms.TextBox tbXamshira;

private System.Windows.Forms.Label label1;

private System.Windows.Forms.Label label2;

private System.Windows.Forms.Label label3;

private System.Windows.Forms.Label label4;

private System.Windows.Forms.Label label5;

private System.Windows.Forms.Label label6;

private System.Windows.Forms.Label label7;

private System.Windows.Forms.Label label8;

private System.Windows.Forms.Label lSana;

private System.Windows.Forms.DateTimePicker dtpTSana;

private System.Windows.Forms.DateTimePicker dtpSana;

private System.Windows.Forms.Button btnSave;

private System.Windows.Forms.Button btnExit;
```

}
}