

N.G. Mardanova

Microsoft ACCESS da

Ma'lumotlar bazasini yaratish bo'yicha

Chuqurlashtirilgan kurs

«Dasturlash III» predmeti bo'yicha AKT tizimlar bo'yicha dasturlashga yo'naltirilgan mutaxassislar uchun

TOSHKENT 2006

Tasdiqlagan: Buxoro viloyati Kogon sanoat-transport kasbiy kolledji Pedagogik kengashi

O'QUV QO'LLANMASI «AXBOROT

VA TELEKOMMUNIKATSIYA TEXNOLOGIYALAR

SOHASIDAGI KASBIY TA'LIMGA KO'MAKLASHISH - KOOPERATSIYA LOYIHASI» O'ZBEK-NEMIS SINOV
LOYIHASI DOIRASIDA ISHLAB CHIQILGAN

MATERIAL QUYIDAGI MUTAXASSISLIK BO'UYICHA O'QITISH UCHUN MO'LJALLANGAN

«INFORMATSION VA KOMMYNIKATSION TIZIMLAR » (kod 3521916)

Retsenzentlar:

1. O.M.Raupov, o'qituvchi, Buxoro viloyati Kogon sanoat-transport kasbiy kolledji
2. prof. Sh.A.Nazirov, Toshkent Axborot Texnologiyalari Universiteti «Dasturlash Texnologiyasi» kafedrası mudiri

Mundarija

Mundarija.....	3
Makroslar.....	6
Makroslarni yaratish texnikasi.....	6
Makroslarni muharrirlash	11
Makroslarni makroguruhga birlashtirish.....	12
Makroslarni menyu komandalari bilan bog`lash.....	14
Makroslarni hodisalar bilan bog`lash.....	15
Makrosni tugma bilan bog`lash	15
Shartlar bilan bog`langan makrosni bajarish.....	16
ACCESS hodisalari.....	19
Ma'lumotlar hodisasi.....	19
Joriy yozuv.....	19
O`chirish.....	19
Del tasdiqlanganicha.....	19
Del tasdiqlanganidan so`ng.....	19
Qo`shimchagacha.....	20
Qo`shimchadan so`ng.....	20
Yangilashgacha.....	20
Yangilashdan so`ng.....	20
O`zgarish.....	20
Ro`yhatda bo`lmaslik.....	21
O`zgarishlar kiritilgan.....	21
Yangilashda.....	21
Yozuvdan ketish.....	22
Bekor qilish.....	22
Fokus hodisalari.....	22
Kirish.....	22
Chiqish.....	22
Fokus olish.....	22

Fokusni yo`qotish.....	22
Ulanish.....	23
Uzilish.....	23
Klaviatura hodisalari.....	23
Klavisha pastga va Klavisha tepaga.....	23
Klavisha bosilishi.....	23
Sichqoncha hodisasi.....	24
Tugmani bosilishi.....	24
Tugmani ikki bosilishi.....	24
Ko`rsatkichni ko`chirish.....	25
Sichqoncha g`ildirakchasi.....	25
Tugma tepaga va Tugma pastga.....	25
Chop etish hodisalari.....	25
Formatlash.....	25
Qaytish.....	26
Chop etish.....	26
Sahifa.....	26
Ma'lumotlar yo`qligi.....	26
Filtr hodisalari.....	26
Filtr qo`llash.....	26
Filtrlash.....	27
Oyna hodisalari.....	27
Ochilish.....	27
Yopilish.....	27
Yuklanish.....	27
Yuklanishni qaytarish.....	27
O`lchamni o`zgartirish.....	28
Hatolar hodisalari.....	28
Taymer hodisasi.....	28
ACCESS hodisalari ketma-ketligi.....	28
Forma boshqarish elementlari uchun fokus hodisalari ketma-ketligi.....	28

Boshqarish elementidagi ma'lumotlar o'zgarish hodisalar ketma-ketligi.....	28
Yozuvlar bilan bog'liq forma hodisalari ketma-ketligi.....	29
Formadagi hodisalar ketma-ketligi.....	29
Klaviatura va sichqoncha hodisalari ketma-ketligi.....	30
Hisobotdagi hodisalar ketma-ketligi.....	31
Hodisalarni qayta ishlash protsedurasini yaratish.....	31
ACCESS dasturlash muhiti.....	33
Kodni tahrirlash oynasi.....	33
Parametrlar oynasi.....	36
Sozlash Instrumental vositalari.....	40
Menyu Debug.....	40
Locals oynasi.....	41
Call Stack oynasi.....	42
Call Stack dialog oynasi.....	42
Hodisani qayta ishlash protsedurasini yaratish.....	42
Visual Basic Sintaksisi.....	43
Protseduralar va funksiyalar, ularning VBA chaqirilishi va parametrlarning uzatilishi.....	43
O'zgaruvchilar, konstantalar va ma'lumotlar turlari.....	45
Massivlar.....	48
O'zgaruvchining va konstantaning harakat sohasi.....	49
Protsedura va funksiyaning harakat sohasi.....	50
Tarmoqlanish.....	51
Sikllar.....	53
So'rovlar yordamida ma'lumotlarni tahlil qilish.....	56
Kesishgan so'rovlar.....	56
Kesishgan so'rov yaratish.....	56
Ustunlar fiksirlangan sarlavhalaridan kesishgan so'rovlarda foydalanish.....	58
So'rovda detalizatsiya darajasini o'zgartirish.....	59
Hisob-kitob uchun statistik funksiyalar.....	60
Hamma oralikdagi yozuvlarni hisoblash.....	61
Jadvalning ajratilgan yozuvlarda hisoblash.....	62

Yig`ma jadvallar.....	63
Yig`ma jadvallarni yaratish.....	63
Yig`ma jadvallarning berilganlarini filtratsiya qilish.....	68
Yig`ma jadvalini o`zgartirish.....	71
Yig`ma jadvalda tartiblash.....	71
Detallashtirish darajasini o`zgarishi.....	72
Umumiy va oraliq yakunlar.....	73
Jamlangan jadvalni formatlash.....	76
Tarkibli formalarni yaratish.....	78
Elementlar paneli.....	78
Boshqaruv elementlarining tarkib tiplari bo`yicha ko`rinishlari.....	79
Yozuv boshqaruv elementi.....	79
Maydon boshqaruv elementi.....	80
Tanlov boshqaruv elementlari:O`chiruvchi, Qayta ulagich, Bayroqcha va Guruh.....	80
Guruh boshqaruv elementi.....	81
Ro`yhat boshqaruv elementi.....	82
Ro`yhatli maydon boshqaruv elementi.....	82
Tugma boshqaruv elementi.....	83
Qo`shimcha sahifalar to`plami boshqaruv elementi.....	84
Ko`pjadvalli formalarni yaratish.....	85
Asosiy va qaram formalarni avtoforma masterlari yordamida yaratish.....	85
Formalar masteri yordamida asosiy va qaram formalarni yaratish.....	87
Formalarda hisoblash.....	89
Hisoblash maydonlarini yaratish.....	89
Yakuniy qiymatlarni aks etish uchun hisoblanadigan maydonlarni yaratish.....	90
Formalarda statistik funksiyalaridan foydalanish.....	90
Funksiya DLookup.....	91
Dcount va Dsum funksiyalari.....	91
Funksiya Dmax.....	92
Tarkibiy hisobotlarni tayyorlash.....	93
Tarkibiy hisobotlarni yaratish.....	93

Tobe so`rovlar qo`llanishi.....	93
Tobe hisobotlar yaratish.....	97
Ma'lumotlarni saralash va guruhlash.....	101
Mijozlar buyurtmalari hisoboti tarkibi aniqlanishi.....	103
Hisoblashni olib borish.....	105
To`plangan summani o`z ichiga oluvchi maydonlarni yaratish.....	105
Foizlarni hisoblash.....	105
Yozuvlarni nomerlash va hisobotda yozuvlar sonini hisoblash.....	105
Matnli ma'lumotlarni birlashtirish va hisobotdagi yozuvlarni tanlash shartlarini chop etish.....	106
Hisobotda ma'lumotlarni filtratsiya qilish.....	106
Ma'lumotlar bazasini administirlash.....	107
Foydalanuvchini boshqarish va foydalanuvchi huquqlari.....	107
Ishchi guruhi faylidan foydalanish.....	107
Hisob yozuvlarini boshqarish.....	108
Ma'lumotlar bazasi ishonchliligi.....	111
Ma'lumotlar bazasi rezerv nushasini yaratish.....	112
Ma'lumotlar bazasini zichlash.....	112
Buzilgan ma'lumotlar bazasini tiklash.....	113
Shifrlash yordamida ahborotni himoyalash.....	113
Import va Export.....	114
SQL - serverdan ob'ektlarni ACCESSga import qilish.....	114
ACCESS ob'ektlarini SQL-server ma'lumotlar bazasiga eksport qilish.....	114
Replikatsiya.....	114
ACCESS ma'lumotlar bazasi replikatsiyasi	115
Tarmoqqa kirish.....	116
Ma'lumotlar bazasi ob'ektlarida giperko`rsatkichdan foydalanish.....	117
HTML formatga o`tkazish.....	120
SQL Server da ma'lumotlar bazalari himoyasi.....	121
SQL Server da ma'lumotlar bazalari ob'ektlari himoyasi.....	122
SQL Server hisob yozuvlarini boshqarish.....	122

Makroslar

Access foydalanuvchilariga ma'lumotlar bazasi bilan ishlash uchun ikkita avtomatizatsiya vositasini taqdim etadi: makroslar tili va Visual Basic for application(VBA) tili. Bu vositalar mashaqqatli operatsiyalar ketma-ketligini qaytarilishni, tugmani yoki tugmalar kombinatsiyasini bosish yoki menyu komandasini aktivizatsiya qilishdan iborat oddiy protseduraga aylantirishga imkon beradi.

VBA ning imkoniyatlari makroslar tiliga nisbatan beqiyosdir. Bunga qaramasdan makroslar tilidan ma'lumotlar bazasini boshqarish bo'yicha ko'pgina operatsiyalarni avtomatizatsiyalash uchun foydalanish mumkin.

Makroslarni yaratish texnikasi

Makroslar oynasi ma'lumotlar bazasi oynasidagi **Макросы** qo'shimcha varaqasidagi **Создать**(yaratish) yoki **Конструктор** tugmasini bosishda ochiladi. Oyna to'rtta ustunni o'z ichiga oladi

Имя макроса (Makros nomi)

Условие (Shart)

Макрокоманда (Makrokomanda)

Примечание (Izoh)

O'rnatilgan bo'yicha yangi makros yaratilayotganda faqatgina Makrokomanda va Izoh

ustunlari aks etadi. Qolgan ustunlarni namoyishi makroslar nomi opsiyasi vositasi bo'yicha va **Вид** menyusidan shart bo'yicha o'rnatiladi. Agar oyna bir necha makroslarni o'z ichiga olsa, berish lozim bo'lgan makros nomini **Имя макроса** ustunida ko'rsatiladi. Makroslar nomlari ko'rsatilayotganda qaytarishlari bo'lmashligi kerak. **Условие** ustunida makrosni faqat bir qismi bajarilishi uchun shart kiritish (mantiqiy ifoda) amalga oshiriladi. **Макрокоманда** ustunida bajarish lozim bo'lgan harakatlar (makrokomandalar) kerakli ketma-ketlikda sanalib chiqiladi. Dasturga sharh saqlovchi **Примечание** ustuni makros bajarilganda dastur tomonidan e'tiborga olinmaydi, biroq uni to'ldirish tavsiya etiladi, chunki bunday xolda makros matni tushunarliroq.

Makrosni qiyin bo'lmagan protsesslarni avtomatizatsiyalash uchun ishlab chiqarish qulaydir, xususan bir necha forma yoki hisobotlarni ochilishi va yopilishi, bir necha xujjatlarni ekranga yoki bosmaga bosib chiqarish va boshqalar.

Makroslarni yaratilishi va qo'llanilishi o'rgangan xolda, makros yordamida ma'lumotlari bazasida bir necha ob'yektlarni ochilish misolini ko'rib chiqamiz. Aniq operatsiyalarni avtomatizatsiyalash uchun mo'ljallangan ma'lumotlar bazasi ko'pgina jadvallar, formalar, so'rovlar va hisobotlardan iborat bo'ladi. Odatda shunday ma'lumotlar bazasida operator ko'p bo'lmagan bir xil miqdordagi ob'ektlar bilan ishlaydi. Xar bir baza bilan ishlash seans boshida qo'shimcha kerakli ob'yektlarni ochish uchun qo'shimcha vaqt ketadi. Bu jarayonni tezlatishga harakat qilamiz: kerakli xujjatlarni ochuvchi va aniq tartibda ekranga joylashtiruvchi makros yaratamiz:

Yangi ma'lumotlar bazasi oynasini oching.

Макросы qo'shimcha sahifasiga o'ting va **Создать** tugmasini bosing, buning natijasida makrosni **Конструктор** oynasi ochiladi.

Ekkranda ham ma'lumotlar bazasi oynasi, ham makros oynasi aks etishi uchun **Окно** menyusidagi **Слева направо**(chapdan o`ngga) buyrug`ini amalga oshiring.

Ma'lumotlar bazasi oynasida **Формы** qo`shimcha sahifasiga o`ting.

O`qituvchilar bo`yicha soatlar miqdori formasiga belgi qo`ying, sichqoncha yordamida makros oynasiga tashib o`ting va **Макрокоманда** ustunini birinchi yacheykasiga joylashtiring. Maydonchada **Открыть форму**(formani ochisi) makrokomandasi paydo bo`ladi. Shunday qilib, makrosda Кол-часов по преподавателям (O`qituvchilar bo`yicha soatlar miqdori) formasini ochish operatsiyasi qo`shiladi.

Примечание ustunida xuddi o`sha qatorga shunday matn kiriting: “ Кол-часов по преподавателям ” formasini oching.

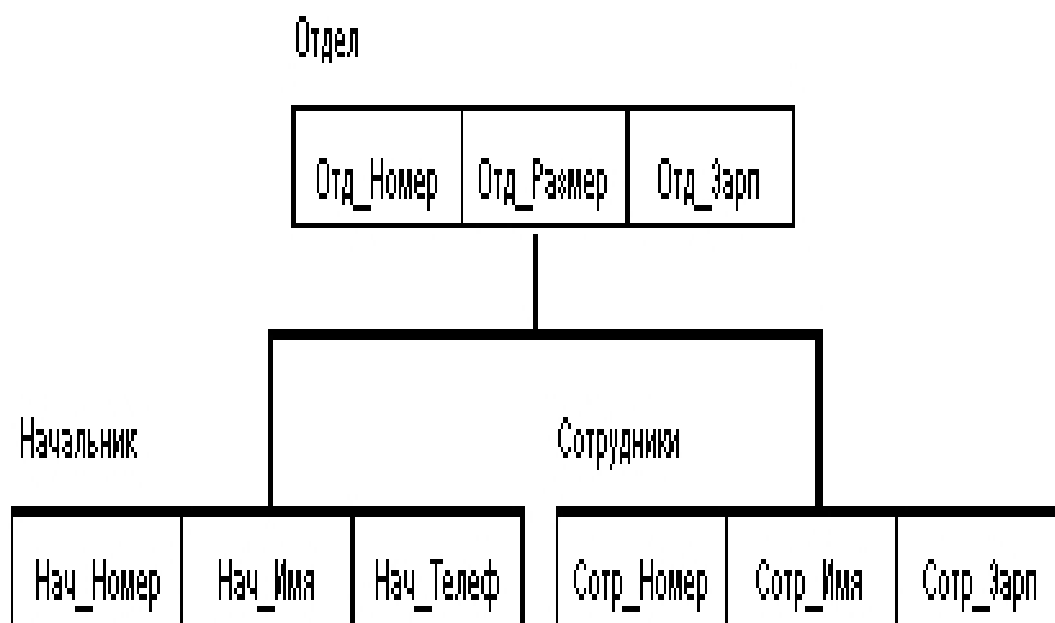
[Enter] klavishi yordamida makrokomanda ustunining ikkinchi qatoriga o`ting

Shu harakatni ochilishi kerak bo`lgan hamma ob'yektlar (misol uchun, Распределение предметов(Darslarni taqsimlash), Список преподавателей (O`qituvchilar ro`yxati) jadvallari) uchun qaytaring.

Makrokomanda ustunining keyingi bo`sh yacheykasini bosing va kirish mumkin bo`lgan makrokomandalar ro`yxatini oching. **Выполнить команду**(buyruqni bajarish) buyrug`iga belgi qo`ying.

Аргументы макрокоманды sohasida Komanda maydonchasini ishga solamiz va ro`yxatdan Tile Vertically elementini tanlaymiz. Shunday qilib, agar bir necha oyna ochilsa, makrokomanda ekranni bo`lish operatsiyasini o`zi ichiga oladi.

Файл menyusidan **Сохранить**(saqlash) buyrug`ini chaqiring va makrosni Размещение на экране nomi ostida saqlab qo`ying.



Размещение на экране(ekkranda joylashtirish) makrosini ishga tushirilgandan so`ng hamma kerakli ma'lumotlar bazasining ob'yektlari yuklanadi va kerakli xolda joylanadi.

Отдел

310	25	12500000
-----	----	----------

Начальник

1846	Иванов	33
------	--------	----

Сотрудники

84	Иванов	1000000
----	--------	---------

83	Сидоров	1000010
----	---------	---------

Xar bir makrokomandani amalga oshirish uning argumentlarining qiymatiga bog`liq( ba'zi makrokomandalar, misol uchun *Развернуть*(yoyish) argumentlarga ega emas). Foydalanuvchi makrokomandalarning sintaksisini yodlab olishi kerak emas- barcha argumentlar buning uchun maxsus makros oynasini pastki qismida joylashgan maydonchasiga kiritiladi. Agar ruxsat etilgan qiymatlar to`plami ba'zi argumentlar uchun chegaralangan bo`lsa, Access ularni ro`yxat sifatida rasmiylashtiriladi. Argumentlarni klaviatura yordamida kiritish mumkin, birok, mavjud bo`lmagan qiymatlarni bermaslik uchun ro`yxatdan tanlagan yaxshirokdir. Argumentlarni makros oynasining pastki qismida qanday joylashgan bo`lsa, shunday berish tavsiya etiladi.

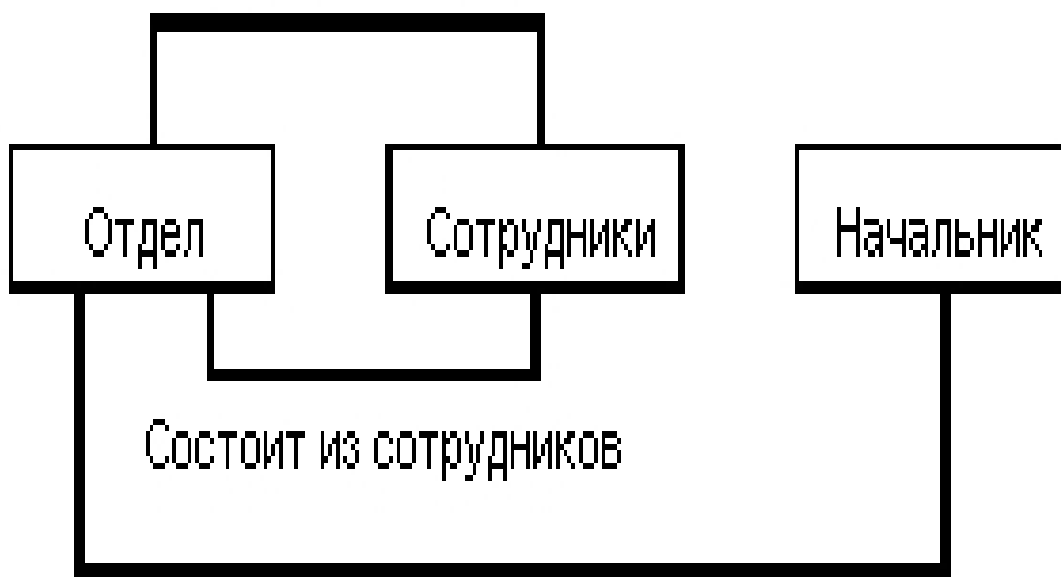
Отдел

Отд_Номер	Отд_Размер	Отд_Зарп
-----------	------------	----------

Начальник

Нач_Номер	Нач_Имя	Нач_Телеф
-----------	---------	-----------

Работают в отделе



Имеет начальника

Макросни ishlab chiqishni biz ob'yektlarni ochish, ya'ni ma'lumotlar bazasining oynasidan mos ravishdagi ob'yektlarni makrosning Конструктор oynasi yacheekalarga tashlab o'tish yo'li bilan makrokomandalarni kiritishni boshladik. Shu bilan birga Access qaysi ob'yekt haqida gap borayotganini avtomatik o'zi tanib, mos ravishda makrokomandalarni tanlaydi: forma uchun *Открыть*(ochish) *форму* ochish yoki jadval uchun *Открыть таблицу*(jadvalni ochish). Tashib o'tilgan ob'yektning nomi makrokomandaning argumentlar sohasida *Имя*(nom) parametri qiymati sifatida paydo bo'ladi.

Макрокомандada ob'yekt nomi ochilishi argumentlar sohasiga klaviatura orkali kiritish mumkin. Asosiysi makros bajarayotganda ochilayotgan ob'yekt mavjud bo'lishi kerak, aks xolda dastur xatolik haqida xabar beradi.

Режим maydonchasiga kiritilgan Макрокоманда ochilish ob'yektining keyingi argumenti ekranda aks etish rejimini aniqlaydi.

Ushbu argumentning ruxsat etilgan qiymati *Режим* menyusidagi opsiyalarga mos keladi.

Режим аргументи qiymati	Qo`llanish	Ta'rif
Печать	Hisobotni	Hisobotni bosmaga chiqarishni beradi
Форма	Formani	Forma rejimini aktivlashtiradi
Таблица	Jadval, so`rov, formani	To`ldirish va o`zgartirish rejimini aktivlashtiradi
Конструктор	Jadvalning, so`rovning, hisobotning, formaning	Konstruktor rejimini aktivlashtiradi
Просмотр	Jadvalni, so`rovni, hisobotni, formani	Sahifani ko`rib chiqish rejimini o`matadi

Выполнить команду makrokomanda yordamida Access menyusidagi ko`pgina buyruqni bajarishni topshirish mumkin. Bajarilayotgan buyruqning nomi *Команда* maydonchasida argument sifatida ko`rsatiladi. Menyu qatorlari tarkibi aktiv ob'yekt axvoli va tipiga bog`liq. *Выполнить команду* makrokomandasidan foydalanilganda, qaysi ob'yekt oxirgi bo`lib aktivizatsiya qilingan va qaysi rejimda joylashganligiga e'tibor berish kerak. Bundan menyu komandasining ochiqligi va bajarilish to`g`riligiga bog`liq.

Microsoft Access Makrokomandalari

Kategoriya	Vazifasi	Makrokomanda
Forma va hisobotlarda ma'lumotlar bilan ishlash	Ma'lumotlarni tanlash	Применить Фильтр (ApplyFilter)
	Ma'lumotlar bo'yicha ko'chish	СледующаяЗапись(FindNext) НайтиЗапись(FindRecord) КЭлементуУправления(GoToControl) НаСтраницу(GoToPage) НаЗапись (GoToRecord)
	Ma'lumotlarni yoki ekranni yangilash	Обновление(Requery) ПоказатьВсеЗаписи (ShowAllRecords)
Bajarish	Komandani bajarish	Выполнить Команду (RunComraand)
	Makros, protsedura yoki so'rovni bajarish	ЗапускМакроса(RunMacro) ЗапускПрограммы (RunCode) ОткрытьЗапрос (OpenQuery) Запуск Запроса SQL (RunSQL)
	Boshqa ilovani bajarish	Запуск Приложения (RunApp)
	Bajarishni uzish	ОтменитьСобытие(CancelEvent) ОстановитьВсеМакросы(StopAllMacros) ОстановитьМакрос (StopMacro)

	Microsoft Accessdan chiqish	Выход (Quit)
Import / eksport	Microsoft Access	ВывестиВФормате(OutputTo)
	Ob'yektlarini boshqa ilovalarga uzatish	ОтправитьОбъект (SendObject)
	Ma'lumotlar formatini o'zgartirish	ПреобразоватьБазуДанных (Transfer Database) ПереносБазыДанныхSQL(TransferSQLDatabase) Преобразовать ЭлектроннуюТаблицу (Transfer Spreadsheet) ПреобразоватьТекст (TransferText)
Ob'yektlar bilan ishlash	Ob'yektdan nusxa olish, Ob'yekt nomini o'zgartirish va ob'yektni saqlash	КопироватьОбъект(CopyObject)
		Копировать Файл БазыДанных (CopyDatabase File) Переименовать (Rename) Сохранить (Save)
O'zgalar	Maxsus yoki xos instrumentlar panelini ekranga chiqarish yoki berkitish Tovush signalini berish	ПанельИнструментов (ShowToolbar) Сигнал (Beep)

Makroslarni muharrirlash

Makroslar oynasidagi jadval tarkibi, ma'lumotlar bazasining oddiy jadval tarkibiga o'xshab ketadi.

Yacheyka tarkibini o'chirish, ko'chirish va ko'paytirish kabi sizga ma'lum matnni muharrirlash buyruqlari makros jadvallarida ham qo'llanishi mumkin. Makroslarni muharrirlash Конструктор rejimida amalga oshiriladi, uni o'rnatish uchun ma'lumotlar bazasi oynasida **Конструктор** tugmasini bosish kerak.

Sharhlarni *Примечание* ustuniga kiritish tavsiya etiladi. Ular makrosga muharrir yoki boshqa foydalanuvchi tomonidan o'zgartirishlar kiritilganda foydali bo'lishi mumkin, xamda makrobuyruqlar haqida, ularni dastlabki markirovkasiz, to'liq ma'lumot olishga imkon beradi. Makrobuyruqning argumentlari va qisqacha tavsifi makros oynasida faqatgina uni markirovka ilganda aks etadi.

Misol sifatida yaratilgan makrosga o'zgartirishlar kiriting-ma'lumotlar bazasi oynasining o'lchamlarini tugmacha o'lchamigacha kichiklashtiring:

Ma'lumotlar bazasi oynasida makroslar qo'shimcha sahifaga o'ting.

Размещение на экране makrosini markerlang va **Конструктор** tugmasi yordamida makros Конструктор oynasini oching.

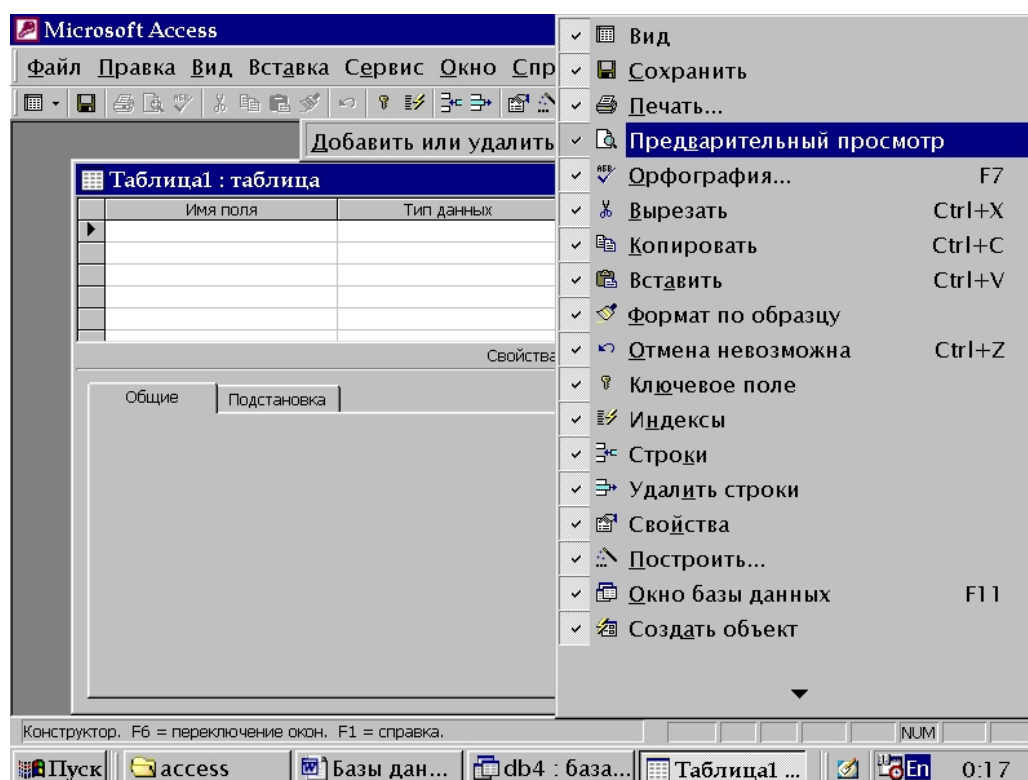
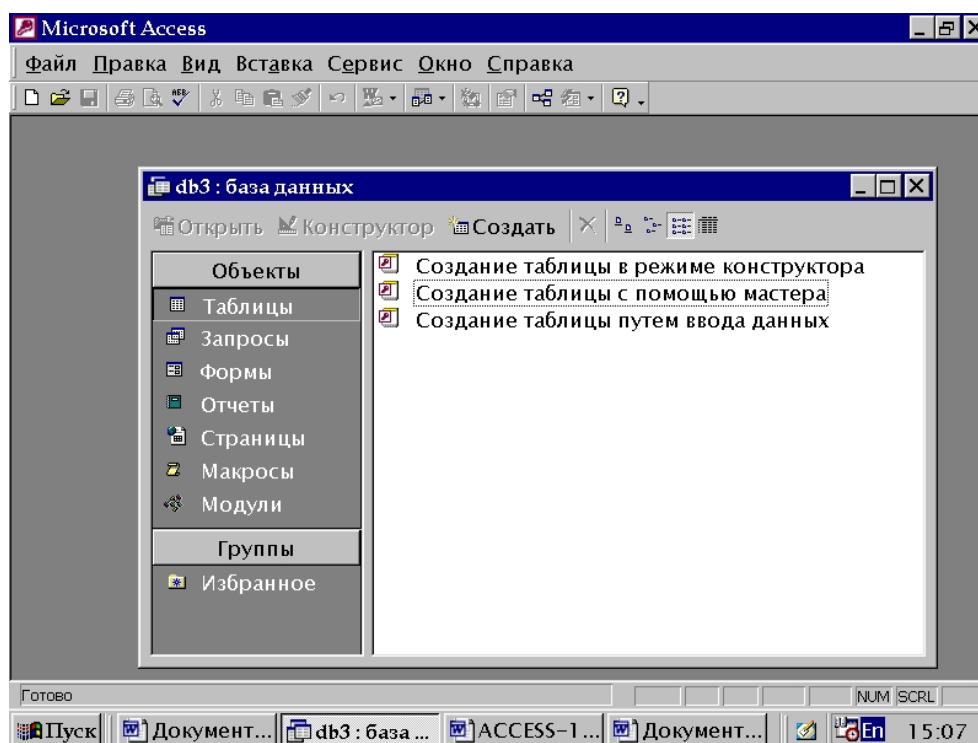
Sichqoncha bilan birinchi qator maydonlaridan biriga sichqoncha tugmasini bosing(yoki uni butunlay markerlang) va **Вставка**(joylash) menyusidagi **Строки** buyrug'ini yoki instrumentlar panelidagi **Добавить строку** ni ishga tushiring. Natijada markerlangan satr oldida bo'sh satr qo'yilgan bo'ladi.

Shu qatorning *Макрокоманда* maydoniga *Свернуть*(jig'ish) buyrug'ini joylashtiring.

Shu ketma ketlikni ochilgan ob'yektlarning biri uchun qaytaring, ob'yekt ochilgan satrdan so'ngbo'sh satr qo'yib va unga *Свернуть* buyrug'ini joylashtirib.

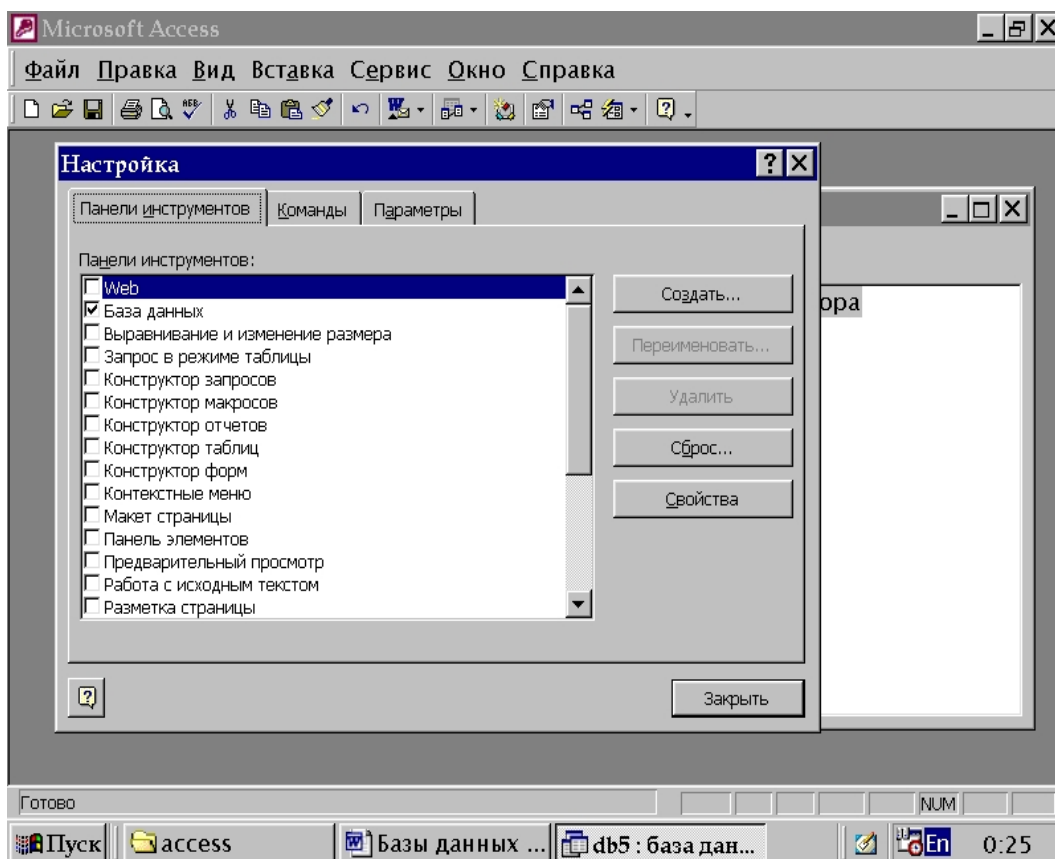
Макросни saqlab qo`ying va ishga tushiring.

Макрос buyrug`ining birinchi ishi, uning yordamida ob'yekt oynasi tugma o`lchamlarigacha kichiklashadi, makrosni chaqirish paytida aktiv ob'yekt(ma'lumotlar bazasi oynasi)ga tegishlidir.



Ma'lumotlar bazasi bittagina ob'yekti bilan ishlash uchun, o'nlab makroslar kerak bo'lishi mumkin, ularning xar biri biror jarayonni avtomatlashtiradi.

Bunday makroslarni alohida ishlab chiqish va navbatma-navbat ishga tushirish mumkin. Lekin bitta ob'yektni ishlatish uchun mo'ljallangan hamma makroslarni bir guruhga birlashtirib, ularga guruh chegarasida unikal nom berish maqsadga muvofiqdir. Guruhga birlashtirilgan makroslarga bitta makros oynasida murojaat qilish mumkin (*Имя макроса* ustuni albatta aks ettirilishi kerak). Keltirilgan rasmda makroslar guruhi(makroguruh)ning tarkibi keltirilgan. Xar bir makros boshi uning *Имя макроса* ustunidagi ismi bilan belgilanadi. Tahrirlash va ko'rish oson bo'lishi uchun ikkita qo'shni makros guruhini bo'sh satr bilan ajratish maqsadga muvofiq.



Makroslar guruhini yaratish uchun quyidagi amallar bajarilishi kerak.

Guruhlashtirilayotgan makroslar joylashtirilishi kerak bo'lgan makros Конструктор oynasini oching.

Ekranida *Имя макроса* ustunini aks ettirish uchun, makroslar ismlari tugmalarini bosing(unda *xyz* literlari aks ettirilgan)yoki **Вид**(ko'rinish) menyusidagi **Имена макросов** buyrug'ini ishga tushiring.

Имя макроса ustunining birinchi yacheykasiga birinchi makros nomini kiriting. Misol uchun *Test1*.

Макрокоманда ustunining yacheykalariga loyihalashtirilayotgan makrosning tanasini tashkil etuvchi makrokomandalarni kiriting.

Keyingi bo'sh satrni ishga tushiring va *Имя макроса* maydoniga ikkinchi makros nomini kiriting. Makroslar orasida bo'sh satrlar qoldirish mumkin.

Qolgan makroslarni yozib, **Сохранить** buyrug'i yoramida makroguruhni saqlab qo'ying.

Makroguruhga kiruvchi makrosni va alohida saqlangan makrosni chaqirish o'zaro farq qiladi. **Открыть** tugmasini bosish

natijasida makroguruhning faqatgina birinchi makrosi bajariladi. Ma'lumotlar bazasidagi makroguruh nomini ikki marta tez bosish natijasida ham xuddi shu narsa sodir bo'ladi.

Makrosni makroguruhdan chaqirish makroguruh va makrosni guruhdagi nomini aniq ko'rsatish yo'li bilan amalga oshiriladi. Bu nomlar nuqta bilan ajratiladi. Agar makroguruh ma'lumotlar bazasi vse *Макросы* nomida kerakli makros esa *-Test1* nomga ega bo'lsa makrosni chaqirish uchun:

Сервис menyusidan **Макрос/выполнить** makros buyrug'ini chaqirish.

Ochilayotgan dialog oynasida *Все Макросы. Test1* tanlansin.

OK tugmasi bosib makros ishga tushirilsin.

Yaratilgan makroguruhga ixtiyoriy paytda yangi makros qo'shish mumkin.

Makroslarni menyu komandalari bilan bog'lash

Accessning ko'pgina buyruqlarini ishga tushirish uchun, ularni bajarish usullarini aniqlovchi qo'shimcha ma'lumotlarni ko'rsatish kerak.

Kerakli ma'lumotlarni olish uchun Access dialog oynasini ochadi, unda foydalanuvchi ko'rsatmalarni bajaradi. Bu amallarni mos keluvchi makrobuyruqlarni yaratish yo'li bilan avtomatlashtirish mumkin. Ularning ba'zilar menyu buyruqlari bilan ekvivalent. *Выполнить команду* makrobuyrug'i bajarilishi kerak bo'lgan menyu buyrug'ini berishga imkon beradi. Misol uchun jadvalda, so'rovda yoki formada yozuvni topish kerak bo'lsa, Find argumentli makrobuyruqdan foydalanish mumkin. Bu makrokomndani bajarishda qidirish natijalarini ko'rsatish uchun dialog oynasi ochiladi.

Qidirish buyrug'i ko'p foydalanilgani uchun mumkin bo'lgan makrobuyruqlar tarkibiga *Найти запись* makrobuyrug'i kiritilgan. Bu makrobuyruq argumentlar ro'yxatida - qidirish dialog oynasining hamma elementlari. U **Правка**(Taxirlash) menyusining **Найти**(Topish) buyrug'i yordamida ochiladi. Bu argumentlar qiymatlarini yaratilish boskichida berib, izingizni makrobuyruqni bajarishda qidirish dialog oynasini to'ldirish majburiyatidan xalos etasiz. Shunday qilib faqat o'zgarmas qidirish shartlarini berish mumkin. Yozuvni o'zgarib turuvchi tarkibi bo'yicha qidirish uchun Find argumentli *Выполнить команду* makrobuyrug'idan keyingi qidirish dialog oynasi maydonlarini to'ldirish kerak. Unda qidirilayotgan original qiymat ko'rsatiladi.

1. Yangi makros oynasini oching va birinchi maydonga СПИСОК ПРЕПОДАВАТЕЛЕЙ (O'qituvchilar ruyxati) jadvalini ko'chiring.

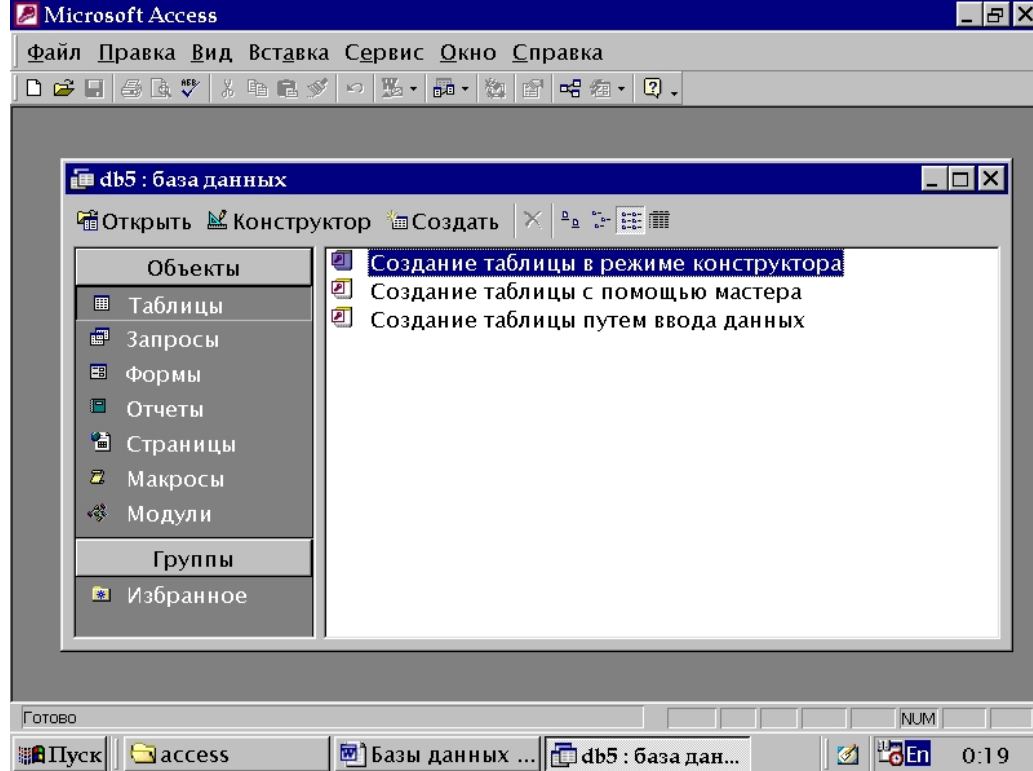
2. Ikkinchi maydonga *К Элементу Управления* buyrug'ini tanlang.

3. *Имя элемента* argumentiga FIOPrep. qiymatini o'rning. Shunday qilib, orasida qidirish amali oshiriladigan maydon berasiz.

4. Keyingi maydonni ishga tushiring va ro'yxatdan *Найти запись* makrobuyrug'ini tanlang.

5. *Образец поиска* argumentiga Samarin V. S qiymatini bering, qolgan argumentlarga bo'lsa rasmda ko'rsatilgan qiymatlarni bering.

6. Makrosni *Поиск записи* nomi ostida saqlab qo'ying va ishga tushiring.



Natijada dastur jadvalni ochadi, qidirish bajaradi va topilgan qiymatni markerlaydi. Agar berilgan qiymat(familiya) topilmasa, jadvalning birinchi elementi(birinchi yozuvi) markerlanadi.

Qidirish shartini bajarilish maydoni **Правка** menyusining **Найти** buyrug`ini chakirib tekshirib ko`rish mumkin. Ochilgan dialog oynasida *Найти Запись* buyrug`ini bajarishda ishlatilgan hamma argumentlar qatnashadi. Biror maydon tarkibini almashish buferiga nusxa ko`chirish uchun, avval bu maydonni *К Элементу Управления* makrobuyrug`i yordamida tanlab olish kerak, keyin:

Поиск Записи makrosini oching va Стаж argumentli *К Элементу Управления* makrobuyrug`ini qo`shing.

Keyingi maydonda Soru argumentli *Выполнить команду* makrobuyrug`ini tanlang.

Saqlab qoling va ishlating.

Natijada topilgan Staj maydoni tarkibi markerlanadi va almashish buferiga nusxa ko`chirilgan bo`ladi.

Макроslarni hodisalar bilan bog`lash

Ma'lumotlar bazasida uning ob'yektlari xolatlari xaqida axborot saklanadi. Forma yoki hisobot xolatining xar qanday o`zgarishi *hodisa* deyiladi. Bu ob'yektlarning xar biri o`z hodisalar to`plamiga ega.

Hodisalar bilan makroslarni bog`lash oson.

Biz tomondan yaratilgan *Размещение на экране* makrosini ma'lumotlar bazasi ochilishi bilan bog`lashga urinib ko`ring.

Размещение на экране makrosini ma'lumotlar bazasi oynasida markerlang.

Правка menyusidan **Переименовать**(Qayta nomlash) buyrug`ini tanlang va *Размещение на экране* makrosiga

yangi ism o`zlashtiring- *AutoExes*(bu nomli makros ma'lumotlar bazasini yuklashda avtomatik bajariladi). Biz ixtiyoriy ravishda qatorli va kayd qilish literlaridan foydalanishimiz mumkin.

Endi makros ma'lumotlar ochilishi bilan bog`liq. Shunday usul bilan xar gal ma'lumotlar bazasi yuklashda bajarilishi kerak bo`lgan operatsiyalar avtomatlashtiradi. Xar gal ma'lumotlar bazasi ochilishida Assess unda *AutoExes* nomli makros katnashayoganligini tekshiradi va agar topsa uni bajaradi. Aks xolda ma'lumotlar bazasi ochilishi hech qanday qo`shimcha harakatlarsiz yechiladi. Ma'lumotlar bazasi ichida faqat bitta *AutoExes* makrosini ishlatish mumkin.

Agar *AutoExes* nomi makroguruhga o`zlashtirish bo`lsa, u xolda ma'lumotlar bazasi ochilishida shu guruhning faqat birinchi makrosi bajariladi.

Yuklashda ketma ket bir nechta makros bajarilishi uchun, *AutoExes* makrosiga Запуск макроса макробуыруг`ini kiriting. Bu makrosni ixtiyoriy nom bilan bajarishga imkon beradi, xamda *AutoExes* makrosining keyingi makrobuyrug`iga qaytishga imkon beradi.

Makrosni tugma bilan bog`lash

Makroslarni chaqirish uchun boshqarish elementlarini yaratish maqsadga muvofiqdir. Bu ularga murojatni ancha tezlashtiradi va qulaylashtiradi. Bunday elementlarni joylashtirishga forma izohi va sarlavha hisoblanadi. Mashq sifatida makros bilan tugmani bog`lashga urinib ko`ramiz. U formaning aktiv yozuvini markerlaydi va uni nuxasini almashtirish buferiga ko`chiradi.

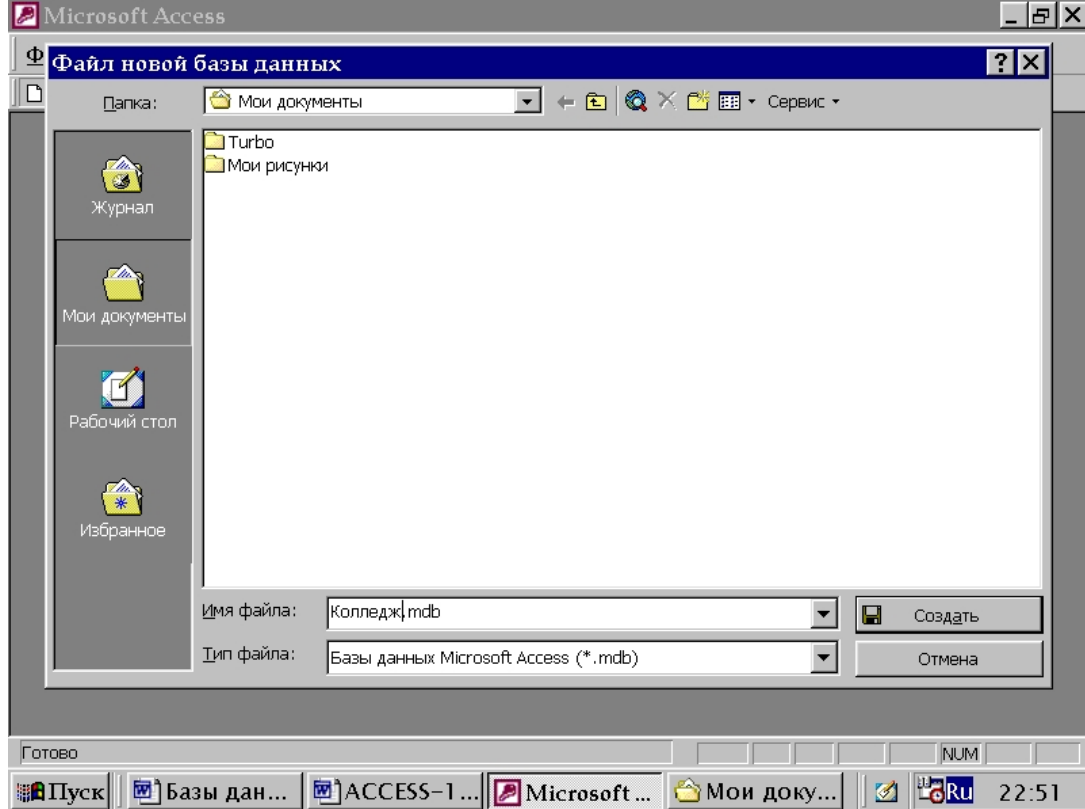
Makrokomanda	Argument
Komandani bajarish	Select Record
Komandani bajarish	Copy

Makrosni saqlang va uni oynasini yoping.

Адреса клиентов formasini oching va Конструктор rejimini ishga tushiring.

Окно(Ойна) menyusining Слева направо komandasini chakiring.

Ma'lumotlar bazasi oynasidagi Копирование Записи makrosini markerlang va uni forma loyihasi sarlavhasi sohasiga olib o`ting.



Сичқонча tugmasini qo`yib yuborishingiz bilan, makros uchun tugma pozitsiyasi saqlab qo`yiladi. Kerak bo`lsa uni o`lchamlarini maxsus manipulyatorlar yordamida o`zgartirish mumkin. Tugmani ko`chirish manipulyator yordamida bajariladi, u markerlaydigan ramkaning chap tepa burchagida joylashgan bo`ladi. Makros nomi avtomatik ravishda tugma ichida paydo bo`ladi, agar so`z juda uzun bo`lsa tugma o`lchamini kattalashtirish kerak. Tugmada butun yozuv ko`rinishi uchun, shrift o`lchamini kichiklashtirish yoki makrobuyruq nomidan farqli yozuv yaratish mumkin. Yozuvni tahrirlash tugma bosilganidan so`ng bajariladi. Tahrirlash rejimidan chiqish uchun tugma sohasidan tashqarida bosish yetarli.

Tugma boshqaruvchi element parametrlarni o`zgartirish uchun uning ustida ikki marta bosish yetarli, buning natijasida agar bundan oldin u mavjud bo`lmagan bo`lsa xossalar oynasi ochiladi.

Xar bir boshqarish elementida o`z mumkin bo`lgan hodisalar ro`yxati mavjud. Lekin bog`lashdagi amallar bir xil bo`ladi. Misol uchun, biror makrosni sichqonchanning forma bo`ylab ko`chishi bilan bog`lash uchun, Конструктор rejimida forma izohidan tashqarida ozod sohada ikki marta bosish va *Перемещение указателя* hodisasi uchun boglanayotgan makros nomini berish yetarli.

Shartlar bilan bog`langan makrosni bajarish

Makrosdagi makrobuyruqlar bajarilish tartibi, shart ustunida berilayotgan shartlar asosida beriladi. Shart bu - *rost(true)* yoki *yolg`on(false)* qiymat qabul qiluvchi mantiqiy ifoda. Keyingi misolni ko`rib chiqamiz. Сумма часов formasiga soatlar taqsimoti xaqidagi ma`lumotlar kiritilgandan so`ng ekranga xabar chiqaradigan makros yaratish kerak bo`lsin.

Yangi makros yaratib unga *Сообщение* nomini o`zlashtirish kerak.

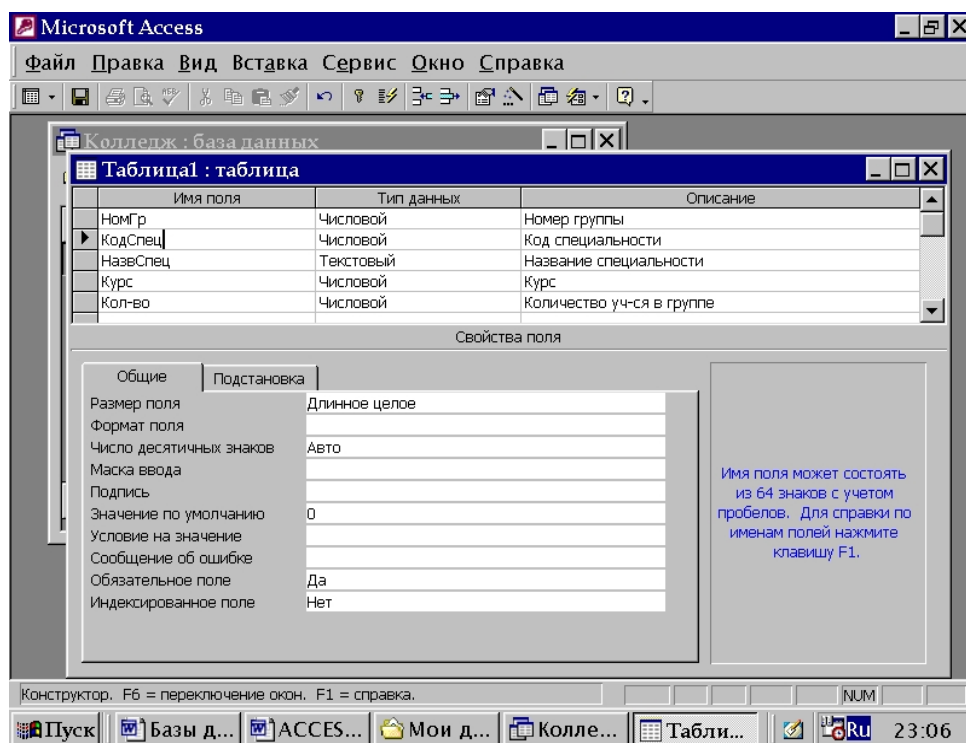
Agar *Условие* ustuni eranda aks etmasa instrumentlar panelidan **Условие** tugmasini bosing yoki **Вид** menyusining **Условие** buyrug`ini ishga tushiring.

Shartlar ustunining birinchi yacheykasiga ushbu mantiqiy ifodani kiring

[Forms]![Список часов][kol chas]>1000

Shu qatorda, lekin *Макрокоманда* ustunidan *Сообщение* makrobuyrug`ini tanlang va *Сообщение* argumentiga soatlar miqdorini 1000 dan katta qilib bering. Xuddi shu ibora berilgan shart bajarilishida ma'lumot oynasida paydo bo`ladi. *Тип* argumentiga *Информационное* qiymatini bering. Shu bilan birga ma'lumotlar oynasida matndan tashqari literli tasvir paydo bo`ladi (Windows muhitida ma'lumot chiqarishda qabul qilingan).

Yaratilgan makrosni saqlang.



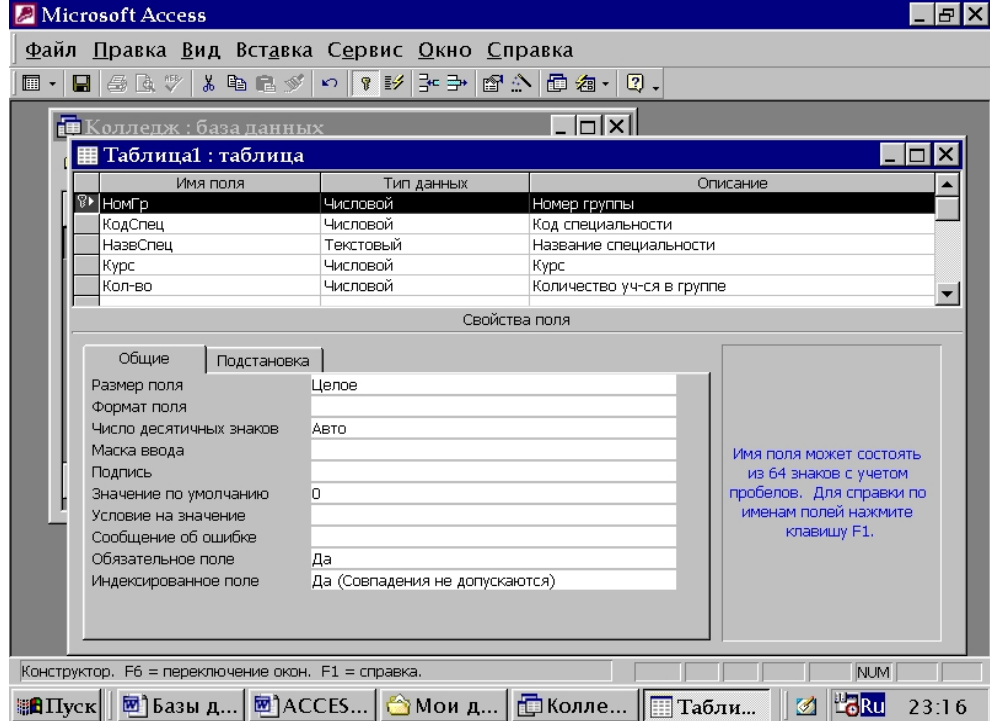
Shunday qilib, o`z ichiga makrobuyruqlar va bajarilish shartlar ketma-ketligini olgan makros tayyor. Shart *Список часов* formasidagi *КолЧас пункт* maydoniga taaluqli. Xabar *КолЧас* maydoniga 1000 dan ortiq qiymat berilganda paydo bo`ladi. Endi bu makrosni *Список часов* formasining *КолЧас* maydoni to`ldirilishi bilan bog`lash kerak:

Список часов formasini oching va *Конструктор* rejimini ishga tushiring.

Izoh sohasidan tashqarida ikki marta sichqonchani tugmasini tez bosib, formaning xossalar oynasini oching.

После обновления (yangilashdan so`ng) maydonini toping va ro`yxatdan *Сообщение* makrosini tanlang. Shu bilan siz makros bajarilishini, *КолЧас* maydoniga 1000 dan katta qiymat berish bilan bog`langan hodisa bilan beramiz.

Formaning xossalar oynasini yoping va to`ldirish rejimiga o`tkazing.



ACCESS hodisalari

Odatda hodisalarni foydalanuvchi harakati keltirib chiqaradi. Foydalanuvchi tomonidan ishlab chikilgan harakatlarga bog'liq ravishda hodisani bir necha turga bo'lsa bo'ladi:

1. Ma'lumotlar hodisasi.
2. Fokus hodisasi.
3. Klaviatura hodisasi.
4. Sichqoncha hodisasi.
5. Bosmaga chiqarish hodisasi.
6. Filtr hodisasi.
7. Oyna hodisasi.
8. Xatolar hodisasi.
9. Taymer hodisasi.

Ma'lumotlar hodisasi

Bu tur hodisasiga, foydalanuvchi kiritganda, o'chirganda yoki o'zgartirganda, xamda bir yozuvdan boshkasiga o'tganda sodir bo'luvchi hodisalar kiradi.

Joriy yozuv(Текущая запись)

Joriy yozuv(**Current**) hodisasi, yana bir yozuv yoki forma ma'lumotlar manbasiga - jadval yoki yozuvga qayta murojat qilish bajarilganda sodir bo'ladi. Shu yo'l bilan, u forma ochilishida paydo bo'lish bilan bir qatorda, bir yozuvdan boshkasiga o'tganda ham sodir bo'ladi. Odatda bu hodisa, bog'langan formalardagi yozuvni sinxronizatsiya qilish uchun ishlatiladi.

O'chirish(Удаление)

O'chirish hodisasi (**Delete**) foydalanuvchi formadan hodisani o'chirib tashlamoqchi bo'lganda sodir bo'ladi. U yozuv ma'lumotlar bazasidan o'chirib tashlanishidan oldin sodir bo'ladi. Bu hodisani qayta ishlash hodisasi **Cancel** parametriga ega. Bu parametrga protsedurada true qiymati berilishi yozuv o'chirilishini oldini oladi.

Del tasdiqlanganicha(До подтверждения Del)

Yozuvni ma'lumotlar bazasidan o'chirilishi shunday katta ahamiyatga egaki, u bilan bir nechta hodisalar bog'liq. O'chirish hodisasi(**Delete**)dan so'ng Access maxsus oyna chiqaradi. Bu oyna yozuv o'chirilishini tasdiqlashni so'raydi. Del tasdiqlanganicha (**BeforeDelConfirm**) hodisasi shu oyna paydo bo'lishidan oldin paydo bo'ladi. Ushbu hodisani qayta ishlash protsedurasi ikkita parametrga ega: **Cancel** va **Response**. Cancel parametriga True qiymatini berib o'chirishni bekor qilish

mumkin. Shunda tasdiqlash oynasi ekranga chiqmaydi. Bu o`chirishni bekor qilish ning yana bir programma yo`li. Agar Cancel parametriga False qiymati berilsa, Responce parametrini tasdiqlash kerak yoki kerakmasligini aniqlash uchun ishlatish mumkin. Agar Responce=1 bo`lsa u xolda yozuv tasdiqsiz o`chirib tashlanadi va agar Responce=0 bo`lsa Access yozuvni tasdiqlashni so`rovchi tasdiqlash oynasini chiqaradi. Agar o`chirishni tasdiqlash oynasi ochilmasa yoki Del tasdiqlanganicha (BeforeDelConfirm) hodisasi chikmasa, Параметры(Options)(Правка и Поиск qo`shimcha sahifasi Потверждение guruxi) oynasining Изменение записей(Record Changes) bayroqcha ko`rsatmasini tekshirib ko`ring.

Del tasdiqlanganidan so`ng(После подтверждения Del)

Del tasdiqlanganidan so`ng (**AfterDelConfirm**) hodisasi yozuvni o`chirish tasdig`idan so`ngsodir bo`lish bilan bir qatorda o`chirishni bekor qilish da ham ishlatiladi. Bu hodisani qayta ishlash hodisasi bitta - **status** parametriga ega. Bu parametr 0, 1 yoki 2 qiymatlarini qabul qiladi va yozuv haqiqatdan ham o`chirilganligini aniqlaydi. 0 qiymati o`chirish muvaffaqiyatli yakunlanganligini bildiradi, 1 qiymati o`chirish hodisalarini qayta ishlash programmasi tomonidan bekor qilinganligini bildiradi, 2 qiymat o`chirish foydalanuvchi tomonidan tasdiqlash oynasida bekor qilinganligini bildiradi. Bu hodisa yozuv o`chirilganligini tekshirish uchun ishlatilishi mumkin.

Qo`shimchagacha(До вставки)

Yangi yozuvni qo`yish bilan ikki hodisa bog`liq: Qo`shimchagacha(**Beforeinsert**) va Qo`shimchadan so`ng(**Afterinsert**).

Qo`shimchagacha hodisasi foydalanuvchi yangi yozuvga(maydonlardan biri, birinchisi bo`lishi shart emas) birinchi simvolni kiritishi bilan sodir bo`ladi, lekin yozuv yaratilguncha sodir bo`ladi. Bu hodisa qayta ishlash protsedurasi qo`shimchaga ruxsat berilganligini tekshirish uchun ishlatilishi mumkin. Protседura bitta Cancel parametriga ega. Agar uni qiymatini True ga tenglashtirib qo`yilsa, qo`shimcha yozuvi taqiqlanadi. Bu hodisadan so`ngqo`shimchani bekor qilib bo`lmaydi, faqatgina qo`yilgan yozuvni o`chirib tashlash mumkin.

Qo`shimchadan so`ng(После вставки)

Hodisa jadvalga yangi yozuv qo`shilgandan so`ngsodir bo`ladi. Odatda bu keyingi formaga o`tishda sodir bo`ladi. Bu hodisa protsedurasi odatda yangi yozuvni chiqarish maqsadida ma'lumotlar manbasiga qayta murojat qilish uchun ishlatiladi.

Yangilashgacha(До обновления)

Yangilashgacha(**BeforeUpdate**) hodisasi Yangilashdan so`ng(**AfterUpdate**) hodisasi kabi yozuvdagi yoki boshqarish elementidagi ma'lumotlarni ixtiyoriy o`zgarishida sodir bo`ladi. Bu hodisa boshqarish elementiga qanday taaluqli bo`lsa yozuvga ham xuddi shunday bog`liq. Bu hodisani qayta ishlash protsedurasi faqat bitta - Cancel parametriga ega. U kiritilgan o`zgarishlarni o`zgartirish uchun ishlatiladi. Buning uchun unga True qiymatini o`zlashtiramiz. Bu hodisa odatda jadval maydonidagi yoki yozuvdagi shartlarni, agar bu shartlar murakkab bo`lsa(oddiiy shartlar odatda Условие xossasining boshqarish elementi qiymatida beriladi) tekshirish maqsadida ishlatiladi. Shartlar birdaniga bir nechta qiymatlarga tekshiriladi. Ularda boshqa formadagi boshqarish elementlariga ilovalar ishlatiladi. Kiritilgan ma'lumotlarning xar xil qiymatida xar xil xatolar xaqida xabarlar beriladi. Shartlar bajarilmaganda kiritilgan o`zgarishlarni keyingi yozuvga o`tishda bekor qilish mumkin.

Yangilashdan so`ng(После обновления)

Yangilashdan so`ng(**AfterUpdate**) hodisasi yozuvdagi yoki boshqarish elementini yangilagandan so`ngsodir bo`ladi.

Yangilash sodir bo`lgan bo`lsa xam, boshqarish elementini OldValue xossasidan foydalanib eski qiymatlarni tiklash mumkin. U boshqarish elementini eski qiymatini saqlab qoladi. Bu qiymat Yangilashdan so`nghodisasidan so`ngalmashadi.

Yangilashgacha(BtforeUpdate) hodisasi va Yangilashdan so`ng(AfterUpdate) hodisasi, xamda Qo`shimchagacha (BeforeInsert) va Qo`shimchadan so`ng(AfterInsert) hodisalari formaning boshqarish elementi qiymati VBA programmasi yoki

Ustanovit znachenie(SetValue) makrokomandasi yordamida o`zgarganda paydo bo`lmaydi. Bundan tashqari Yangilashgacha(BtforeUpdate) hodisasi va Yangilashdan so`ng(AfterUpdate) hodisasi hisoblanayotgan boshqarish elementlari uchun chiqmaydi.

O`zgarish(Изменение)

O`zgarish(**Change**) hodisasi quyidagi xollarda paydo bo`ladi:

Matn maydoni yoki yozuvli maydon tarkibi o`zgarishida, shu bilan birga o`zgarish deb bevosita kiritilgan yoki o`chirilgan simvol xisoblanadi.

Boshqarish elementi Текст(Text) xossasi qiymati VBA protsedurasi yoki makros yordamida o`zgarganda.

Набор вкладок(Tab Control) boshqarish qurilmasida bir qo`shimcha sahifadan boshqasiga o`tganda.

O`zgarish(Change) hodisasi hisoblanayotgan boshqarish elementi o`zgarishida, xamda VBA protsedurasi yoki makros yordamida matn maydoni yoki yozuvli maydon qiymati o`rnatilganda yoki yozuvli maydon qiymati ro`yxatdan tanlangan bo`lsa paydo bo`lmaydi.

Ro`yxatda bo`lmaslik(Отсутствие в списке)

Ro`yxatda bo`lmaslik(**Notinlist**) hodisasi, foydalanuvchi ro`yxatda bo`lmagan tekst maydoniga qiymatni qo`lda kiritib, boshqa maydonga o`tmokchi bo`lganda yoki yozuvni saqlab qo`ymokchi bo`lganda paydo bo`ladi. Ushbu hodisa sodir bo`lishi uchun Ограничиться списком(LimitToList) xossasiga Yes qiymatini berish kerak. Agar bu xossa No qiymatiga ega bo`lsa, u xolda ma'lumotlar maydoniga ro`yxatdagi birorta qiymat bilan mos kelmaydigan qiymat kiritish ruxsat beriladi. Bu hodisa qayta ishlash protsedurasi ikkita parametrga ega :NewData va Response. NewData parametri kiritilgan ma'lumotlarni, Response parametri esa hodisani qayta ishlashni boshqaradi va 0, 1 yoki 2 qiymatlarini qabul qilishi mumkin. 0 qiymati ekranga kiritilgan ma'lumotlar ro`yxatda yo`qligini ko`rsatuvchi standart xabar chikradi va kiritishni taqiqlab qo`yadi. 1 qiymati standart xabar o`rniga maxsus xabar chiqaradi. Ushbu xabar misol uchun ro`yxatga kiritilgan yangi qiymatlarni saqlab qo`yish kerakligini so`rovchi xabar bo`lishi mumkin. Bunda yangi ma'lumotlar ro`yxatga qo`shilmaydi. 2 qiymati ro`yxatga yangi qiymat qo`shishga ruxsat beradi. Bunda bu hodisani qayta ishlovchi protsedurasi yozuvli maydonining qatorlar manbasiga yangi qiymat kiritib qo`yish kerak. Bundan so`ng Access qatorlar manbasini qayta so`ragani uchun maydon yangilanadi.

Lekin yozuvli maydon uchun qatorlar manbayi, ma'lumotnoma-jadval bo`lsa, u xolda oddiy qo`shishning o`zi yetarli bo`lmasligi mumkin. Maxsus forma chiqarish kerak bo`ladi. Unda foydalanuvchi hamma kerak maydonlarni to`ldirishi kerak bo`ladi. Bu formada yozuvlar saqlanganidan so`ng yangi ma'lumotlar ro`yxatga qo`shiladi. Shunday harakatlar bajarish kerak bo`lgan xuddi shunday xolat yangi klientni unga standart xujjat yozib berilayotganda:hisob, nakladnoy va boshqalar.

O`zgarishlar kiritilgan(Внесены изменения)

O`zgarishlar kiritilgan(**Dirty**) hodisasi xuddi o`zgarish (**Change**) kabi quyidagi xollarda sodir bo`ladi:

Matnli maydon yoki yozuvli maydon tarkibi o`zgarishida sodir bo`ladi. Bunda o`zgarish ixtiyoriy kiritilgan yoki o`chirilgan simvol bo`lishi mumkin.

boshqarish elementining Текст (Text) xossasining qiymati VBA protsedurasi yoki makros yordamida o`zgartirilganda.

Набор вкладок(Tab control) boshqarish elementining bir qo`shimcha sahifadan boshkasiga utganda.

Lekin O`zgarish (Change) hodisasidan farqli, u formaga taaluqli. Protsedura bitta -Cancel parametrga ega. Agar unga True qiymati berilsa, u xolda hodisa bekor kilinadi.

Hodisa bekor kilinishi yozuvdagi hamma o`zgarishlarni otkat bo`lishiga olib keladi. Bu <Esc>klavishasini bosishga ekvivalent. Bu hodisani yozuvda o`zgarishlar bo`lganligini tekshirish uchun qulay.

Yangilashda(При обновлении)

Yangilashda(**Updated**) hodisasi OLE ob'yektini o`zgarishida sodir bo`ladi va faqat ozod va ob'yektga birlashtirilgan ramkaga ishlatiladi.

Bu hodisa qayta ishlash protsedurasi OLE ob'yekti ma'lumotlari oxirgi saqlashdan so`ng o`zgarganligini tekshiradi. Protседura bitta Code parametrğa ega. U ob'yekt qanday o`zgarganligini ko`rsatadi va 0, 1, 2, 3 qiymatlarini o`zgartira oladi. 0 qiymati ob'yekt o`zgarganligini ko`rsatadi. 1 qiymati ob'yekt yaratgan ob'yekt ma'lumotlari ko`rsatma bilan saklanib kolganligini bildiradi. 2 qiymati OLE ob'yekti fayli uni yaratgan ko`rsatma bilan yopiladi. 3 qiymati OLE ob'yekti fayli uni yaratgan ko`rsatma bilan ismi o`zgarganligini bildiradi.

Yozuvdan ketish(Уход с Записи)

Yozuvdan ketish hodisasi(**RecordExit**) xar gal foydalanuvchi hozirgi yozuvdan chiqmokchi bo`lganda sodir bo`ladi:boshqa yozuvga o`tish, formani yopish, formadagi ma'lumotlarni yangilash va boshqa. Bu hodisa qayta ishlash protsedurasi shu faylidagi yozuvlar aniqligini tekshirish uchun ishlatiladi. Protседura bitta Cancel parametrğa ega. Agar uni qiymatini True ga tenglashtirilsa, u xolda foydalanuvchiga ushbu yozuvni tark etishni ta'kiklab qo`yish mumkin.

Bekor qilish(Отмена)

Bekor qilish hodisasi foydalanuvchi ayni paytda maydondagi yoki ayni paytda yozuvda qilingan o`zgarishlarni bekor qilish uchun ishlatiladi. Misol uchun <ESC>klavishasini bosib yoki instrumentlar panelidagi \$\$ (**Undo Field/Record**) tugmasini bosib. Bu hodisani qayta ishlash protsedurasi faqat bitta Cancel parametrğa ega. Unga True qiymatini berib bekor qilish operatsiyasini uzib qo`yish mumkin, shunda ayni paytdagi maydondagi ma'lumotlar yoki yozuv o`zgarganicha bo`lib qoladi.

Fokus hodisalari

Fokus hodisasi forma, hisobot yoki formadagi boshqarish elementi qabul kilganda yoki yo`qotganda, xamda forma yoki hisobot aktiv yoki uning aksi bo`lganda sodir bo`ladi.

Kirish(Вход)

Kirish(**Enter**) hodisasi formadagi boshqarish elementi shu formaning boshqa boshqarish elementidan fokus qabul kilganda yoki formani ochganda fokusni birinchi boshqarish elementi qabul qilganda bajariladi. Uni shu element haqida ekranga qandaydir ma'lumotlar chiqarganda qulay. U Fokusni olish (GetFocus) hodisasigacha sodir bo`ladi, lekin Текущая запись(Current) hodisasidan keyin.

Chiqish(Выход)

Chiqish(**Exit**) hodisasi formadagi boshqarish elementi shu formaning boshqa boshqarish elementidan fokus qabul qilish gacha, lekin Fokusni yo`qotish (LostFocus) hodisasidan oldin sodir bo`ladi.

Kirish(Enter) va Chiqish(Exit) hodisalari guruhdagi bayroqcha xamda qayta ulagichlar uchun to`liqligicha aniqlanmagan. Kirish(Enter) va Chiqish(Exit) hodisalari sodir bo`lmaydi, agar fokus boshqa forma elementiga yoki hisobotga o`tsa. Bu ularni Fokusni olish (GetFocus) va Fokusni yo`qotish (LostFocus) hodisalaridan farqlaydi.

Fokus olish(Получение фокуса)

Fokus olish (**GetFocus**) hodisasi forma yoki formaning boshqarish elementi fokus olganda ishga tushadi. Boshqarish elementi forma qabul qila oladi, faqat agar u ko`rinsa va mumkin bo`lsa(uning Ekranga chiqarish xossasi(Visible) va Murojaat imkoni (Enabled) xossalari Ha qiymatiga ega). Shu bilan birga Fokusni olish(GetFocus) hodisasi Kiritish(Enter) hodisasidan so`ngsodir bo`ladi. Forma fokus qabul kila oladi agar undagi hamma maydonlar blokirovka qilingan bo`lsa, aks xolda Fokusni olish (GetFocus) hodisasi forma uchun chiqmaydi.

Fokusni yo`qotish(Потеря фокуса)

Fokusni yo`qotish (**LostFocus**) hodisasi xar gal forma yoki forma boshqarish elementi fokus yo`qotganda sodir bo`ladi. Hodisa Chiqarish(Exit) hodisasidan so`ngsodir bo`ladi.

Fokusni olish (GetFocus) va Fokusni yo`qotish (LostFocus) hodisalari na faqat foydalanuvchi harakatlari bilan(klaviatura yoki sichqoncha yordamida) balki SetFocus ob'yekti usuli chaqirishda yoki makrokomandalar bajarilishida ham sodir bo`ladi.

Ob'yektni ajratish(SelectObject), Yozuvga (GoTo Record), Boshqarish Elementiga (GoTo Control) va Sahifaga (Go To Page).

Bundan tashqari bu hodisalar guruh ichidagi elementlar (bayroqcha va qayta ulagich) uchun cham aniqlangan.

Ulanish (Включение)

Ulanish(**Active**) hodisasi forma yoki hisobot fokus olib aktiv forma yoki hisobot bo`lganda bajariladi. Bu forma va hisobot foydalanuvchi sichqoncha bilan boshqarish elementlaridan birini bosib fokusni ko`chirganda va VBA programmasida SetFocus ob'yekt usuli bajarilganda ochiladi. Ulanish (Active) hodisasi Fokusni olish(GetFocus) hodisasigacha sodir bo`ladi. Bu hodisani forma bilan bog`langan instrumentlar panelida ekranga chiqarish qulay.

Uzilish(Отключение)

Uzilish (**Deactive**) hodisasi fokus forma yoki hisobotdan boshqa oynaga ko`chirilganda sodir bo`ladi(jadvallar, so`rovlar, hisobotlar, makroslar, modullar yoki ma'lumotlar bazasi oynasi). Lekin bu hodisa fokus dialog oynasiga yoki izohga utganda sodir bo`lmaydi. Uzilish (Deactive) hodisasi **Fokusni yo`qotish** (LostFocus) hodisasidan so`ngchiqadi .

Klaviatura hodisalari

Klaviatura hodisasi formada yoki forma elementida, foydalanuvchi klaviatura tugmalarini bosganda yoki sendKeys makrokomandasini bajarganda sodir bo`ladi.

Hamma klaviatura hodisalari formadagi ayni damda fokusga ega bo'lgan ob'yekt bilan boglanadi. Odatda bu boshqarish elementlaridan biri. Forma fokus qabul qila oladi(demak klaviatura hodisalari ham formaga taaluqli bo'la oladi) faqat agar uning hamma boshqarish elementlari blokirka qilingan yoki ko'rinmas bo'lsa.

Agar bu hodisalarni aynan formaga bog'lash kerak bo'lsa, ya'ni forma elementiga emas, u xolda Tugma bosilishini ushlab olish (KeyPreviev) xossasiga forma uchun Ha(Yes) qiymati beriladi. Shunda klaviaturaning hamma hodisalari oldin forma uchun so'ngfokusga ega bo'lgan boshqarish elementi uchun bajariladi. Bu forma reaksiyasini fokus, formaning qaysi boshqarish elementida yotishidan kat'iy nazar ma'lum tugmalarni programmashtirish imkonini beradi.

Klaviatura hodisalari hisobot boshqarish elementlari uchun aniqlanmagan, xamda xamda guruhdagi bayroqcha va qayta ulagichlar uchun xam. Ular faqat guruhlarga to'laligicha aniqlangan.

Klavisha pastga(Клавиша вниз) va Klavisha tepaga (Клавиша вверх)

Klavisha pastga (KeyDown) va **Klavisha tepaga (KeyUp)** hodisalari xar gal foydalanuvchi klaviaturaning tugmalarini bosganda yoki qo'yib yuborganda ishga tushadi, bunda fokus boshqarish elementida yoki formada bo'ladi. Bu hodisani qayta ishlash protsedurasi foydalanuvchi qaysi tugmani bosganligini aniqlash uchun ishlatiladi-funksional, kursorni boshqarish tugmasi, raqamli panel tugmalari yoki <Shift>, <Ctrl>yoki<Alt>tugmalari kombinatsiyasi Ular ikkita:KeyCode va Shift parametrga ega. KeyCode bu - xar bir tugmani kodini ko'rsatuvchi butun son. Shift parametri qaysi tugmalar kombinatsiyasi bosilganligini ko'rsatadi- 1-<Shift>ga to'g'ri keladi, 2-<Ctrl>, 4-<Alt>.

0- tugmalar bosilmaganligini bildiradi. Agar <Shift>, <Ctrl>, <Alt> tugmalari ixtiyoriy kombinatsiyada bo'lsa, Shift parametri shu tugmalar qiymati yig'indisiga teng buladi.

Klavisha bosilishi(Нажатие клавиши)

Klavisha bosilishi(**KeyPres**) hodisasi agar foydalanuvchi fokusga ega bo'lgan ixtiyoriy tugmalar kombinatsiyasini bossa yoki qo'yib yuborsa sodir bo'ladi. Klavisha pastga(KeyDown) va Klavisha tepaga(KeyUp) hodisalaridan farqli bu hodisa funksional, kursor boshqarish va <Shift>, <Ctrl>yoki<Alt> klavishalari bosilsa ishlamaydi. Bundan tashqari bu hodisalar yuqori va quyi registrlar uchun xar xil. Hodisani qayta ishlash protsedurasi bitta argument:KeyAscii-bosilgan klavisha kodini ko'rsatuvchi butun son. Agar foydalanuvchi klavishasini bosib qo'yib yubormasa, u xolda Klavisha pastga(KeyDown) va Klavisha tepaga(KeyUp) hodisalari, u klavisha qo'yib yubormaguncha takrorlanadi. Shunda Klavisha tepaga(KeyUp) hodisasi bajariladi. Agar klavisha bosilish natijasi fokusni bir elementdan boshkasiga o'tkazish bo'lsa, u xolda Klavisha pastga(KeyDown) hodisasi birinchi element uchun sodir bo'ladi, Klavisha tepaga(KeyUp) va Klavisha bosilishi(KeyPres) hodisalari ikkinchi uchun bajariladi.

Agar tugma bosilishi natijasida dialog oynasi ko'rinsa Klavisha pastga(KeyDown) va Klavisha bosilishi(KeyPres) hodisalari bajariladi, Klavisha tepaga(KeyUp) hodisasi esa ishlamaydi.

Sichqoncha hodisasi

Bu hodisa formadagi yoki uning biror boshqarish elementida qandaydir harakat sichqoncha yordamida qilinsa ishga tushadi. Sichqoncha hodisalari hisobot boshqarish elementlariga, xamda guruhlardagi bayroqcha va qayta ulagichlar uchun aniqlanmagan. Ular faqat guruhlarga to'liqligicha aniqlangan.

Tugmani bosilishi(Нажатие кнопки)

Bu eng keng tarkalgan hodisa. U formada ham uning boshqarish elementlarida ham paydo bo'ladi.

Forma uchun Tugmani bosilishi hodisasi (**Click**) sichqoncha tugmasini nafaqat element ustida bosganda, balki unga biriktirilgan yozuvda bosganda ham sodir bo`ladi.

Lekin boshqarish elementi uchun u faqat sichqoncha tugmasini bosganda emas, boshqa ba'zi xollarda ham sodir bo`ladi:

Elementni ro`yxatdan tanlashda, uni sichqoncha bilan tanlanganligidan yoki kursor boshqarish klavishini keyinchalik <Enter> tugmasini bosish bilan tanlashdan kat'iy nazar;

<Bo`shlik> tugmasini bosishda, fokus bayroqchada, qayta ulagich yoki buyruq tugmasida o`rnatilgan bo`lsa;

formada Да qiymatli O`rnatilgan bo`yicha (Default) tugma xossasini <Enter> tugmasini bosishda. Shunda xuddi shu tugmaga fokus o`rnatiladi.

formada Да qiymatli Rad etish (Cancel)xossalar tugmasini o`z ichiga oluvchi, <Esc> tugmasini bosishda;

Murojaat imkoni tugmasini bosishda, agar ular formadagi tugmalar bilan bog`liq bo`lsa.

Shunday qilib Tugmani bosilishi (Click) hodisasini qayta ishlash protseduralari bu tugma qanday tanlanganligidan kat'iy nazar-sichqoncha tugmasini bosishda, <Enter> tugmasini bosishda yoki qabul tugmasi ishga tushadi. Hodisani qayta ishlash protsedurasi faqat bir marta ishga tushadi. Agar uni tugmacha bosilib turgunicha bir necha bor ishlashi talab etilsa, u xolda tugma uchun Avtomatik qaytarish (AutoRepeat) xossasidan foydalanish kerak. Agar sichqonchaning qaysi tugmasi bosilganligini bilish kerak bo`lsa, u xolda Tugma pastga (MouseDown) va Tugma tepaga (MouseUp) hodisalaridan foydalanamiz.

Tugmani ikki bosilishi(Двойное нажатие кнопки)

Tugmani ikki bosilishi(**DbClick**) hodisasi formadagi yoki boshqarish elementidagi ixtiyoriy tugmani ikki marta tez bosilishidan sodir bo`ladi. Bunda bosishlar orasidagi интервал Windows ning instrumentlar panelida ko`rsatilgan vaqtdan oshmasligi kerak. Tugmani ikki bosilishi(DbClick) hodisasi forma yoki forma boshqarish elementi, xuddi Tugma bosilishi hodisasi (Click) kabi aniqlangan. Lekin boshqarish elementlari uchun bu hodisa natijasi boshqarish elementining turiga bog`liq.

Ko`zda tutilgan xolda sichqonchaning matnli maydonda ikki marta bosilishi so`zning ajralishiga olib keladi, OLE ob'yektida esa shu ob'yektni muharrirlashga tushiradi. qayta ishlash protsedurasini xakikiy hodisaga kiritib, Accessning standart harakatlarini pereoredelit qilish mumkin. Protsedura bitta Cancel parametrga ega Agar unga protsedurada True qiymati berilsa, u xolda bu hodisani bekor qilish mumkin.

Sichqoncha tugmasini ikki marta bosilishi aslida ikkita hodisani chaqiradi:birinchi Tugma bosilishi (Click) hodisasi, so`ng Tugmani ikki bosilishi(DbClick) hodisasi.

Ko`rsatkichni ko`chirish(Перемещение указателя)

Bu hodisa tinmay generatsiya kilinadi, qachonki foydalanuvchi sichqoncha ko`rsatkichini forma ob'yektlari bo`ylab ko`chirganda. Ko`rsatgich ob'yekt chegarasida harakat qilguncha Ko`rsatkichni ko`chirish(**MouseMove**) hodisasi shu ob'yekt uchun, ko`rsatkich formaning bo`sh qismiga tushganda, yozuvni ajratish sohasiga tushganda yoki siljitish(prokrutka) chizig`ida forma uchun Ko`rsatkichni ko`chirish(MouseMove) hodisasi generatsiya qiladi. Hodisa shu bilan birga formani yoki boshqarish elementini ko`chirishda sodir bo`ladi. Misol uchun VBA protsedurasi yordamida, sichqonchaning qo`zg`almas ko`rsatkichida. Protsedurani qayta ishlash hodisasi to`rtta parametrga ega:

Button-hodisa bo`lish damidagi sichqoncha tugmasi xolatini aniqlaydi(ko`rsatkich ko`chishi sichqonchaning bir necha marta yoki umuman bosilmaganida sodir bo`lishi mumkin).

Shift-<Shift>, <Ctrl>, <Alt> tugmalari xolatini Button parametri bilan aniqlanuvchi, tugma bosilgandagi axvolini aniqlaydi.

X va Y- sichqoncha ko`rsatkichini turlardagi ayni damdagi koordinatalari;

Sichqoncha g`ildirakchasi(Колесико мыши)

Foydalanuvchi skrolling g`ildirakchasi yordamida sichqoncha ko`rsatkichini harakatlantirganda hodisa yuz beradi. Hodisani qayta ishlash protsedurasi ikki parametrga ega:

Page - ko`rsatkich boshqa sahifaga o`tganda True qiymat qabul qiladi;

Count - Formani sichqoncha g`ildirakchasi yordamida siljirilgandagi ko`rsatkich ko`chgan chiziqlar soni.

Tugma tepaga(Кнопка вниз) va Tugma pastga(Кнопка вверх)

Tugma pastga(**MouseUp**) va Tugma tepaga(**MouseDown**) hodisalari foydalanuvchi sichqoncha tugmasini bosganda yoki qo`yib yuborganda sodir bo`ladi, xamda Tugma bosilishi (Click) vaTugmani ikki bosilishi(DblClick) hodisalaridan farqli qaysi tugma bsoilganligini aniqlaydi. Bu hodisani qayta ishlash protsedurasi to`rtta parametrga ega:Button, Shift, X, Y. Bu parametrlar birinchi Button parametridan tashqari Ko`rsatkichni ko`chirish(MouseMove) hodisasiga uxshash. Chunki bu xolda aniq bir tugma bosiladi, Button parametri qaysi tugmaligini aniqlaydi. Agar foydalanuvchi birdaniga ikkita tugma bossa, birinchi va ikkinchi tugmalar uchun hodisalar bajariladi. Agar sichqoncha tugmasi ko`rsatkich formaning boshqarish elementlaridan birida bo`lgan bo`lsa, u xolda aynan shu ob'yektga sichqonchaning keyingi hamma hodisalari Tugma tepaga(MouseUp) hodisasini ham qo`shib hisoblaganda taaluqli bo`ladi.

Chop etish hodisalari

Chop etish hodisasi hisobot tomonidan va uning ixtiyoriy sohalaridan chop etishda yoki predvaritelniy ko`rishda sodir bo`ladi.

Formatlash(Форматирование)

Formatlash(**Format**) hodisasi hisobot uchun ma'lumotlar tanlanganidan keyin, lekin hisobotning deyarli xar bir qismi Formatlanishdan oldin sodir bo`ladi. Shu bilan birga ma'lumotlar qismi uchun bu hodisa hisobotning xar bir yozuvi uchun sodir bo`ladi. Bu xar bir yozuvni xar xil formatlash imkonini beradi. Hisobotdagi guruh sarlavhasi uchun hodisa xar bir guruh uchun sodir bo`ladi. Bu hodisani qayta ishlash protsedurasi ikkita-Cancel va FormatCount parametrga ega. Cancel ushbu qismni formatlashni bekor qilish imkonini beradi. Buning uchun unga True qiymatini o`zlashtirishni o`zi kifoya. Bu sizga sahifada bo`sh joy qoldirmay hisobot qismlarini o`tkazib yuborish imkonini beradi. FormatCount- Formatlash(Format) hodisasi necha marta sodir bo`lganligini hisoblaydigan hisoblagich.

Qaytish(Возврат)

Hodisa formatlashda Formatlangan qismga qaytish kerak bo`lib kolganda sodir bo`ladi. U Formatlash(Format) hodisasidan oldin, lekin Chop etish(Print) hodisasidan so`ngsodir bo`ladi. Bu hodisa qayta ishlash protsedurasi ixtiyoriy formatlab bo`linganni o`zgartirish imkonini beradi. Shu yo`l bilan hisobot elementlarini sahifadagi kerakli joylashishini ta'minladi. Qaytish(**Repeat**)hodisasi yuqori va quyi kolontitullar uchun aniqlanmagan.

Chop etish(Печать)

Chop etish(**Print**) hodisasi qism formatlangandan keyin, lekin chop etilmasidan oldin sodir bo`ladi. Bu hodisa chop etilmaydigan hodisalardan tashqari deyarli xar bir formatlashdan so`ngsodir bo`ladi. Bu hodisa ikkita:Cancel va PrintCount parametrga ega. Cancel kuzatilayotgan qismni chop etishni bekor qilishni ta'minlaydi. Buning uchun unga True qiymatini o`zlashtirish kerak. Lekin bunda sahifada bo`sh joy qoladi, shuning uchun bu protsedurani o`zgarishlar sahifa formatiga tegishli bo`lmaganda ishlatish mumkin. PrintCount-Chop etish hodisasi necha marta bajarilganligini hisoblovchi hisoblagich.

Sahifa(Страница)

Sahifa(**Page**) hodisasi sahifa formatlangandan keyin, lekin chop etilmasidan oldin sodir bo`ladi va qayta ishlash protseduralari yordamida sahifaga ba'zi rasmiylashtirish elementlarini qo`shish imkonini beradi. Misol uchun ramka.

Ma'lumotlar yo`qligi(Отсутствие данных)

Ma'lumotlar yo`qligi(**NoData**) hodisasi hisobot formatlangandan keyin, lekin chop etilmasidan oldin sodir(birinchi Sahifa(Page) hodisasigacha) bo`ladi va hisobot uchun ma'lumotlar yo`qligini aniqlash imkonini beradi. Bu xolda chop etishni bekor qilish mumkin. Bu hodisani qayta ishlash protsedurasi bitta Cancel parametrga ega. Unga chop etishni bekor qilish uchun True qiymatini o`zlashtirish kerak.

Filtr hodisalari

Filtr hodisasi formada filtr ishlatganda yoki uni o`chirganda ishlatiladi.

Filtr qo`llash(Применение фильтра)

Filtr qo`llash(**ApplyFilter**) hodisasi foydalanuvchi formada yozuvlarni mos menyu buyruqlari, kontekstli menyu yoki instrumentlar paneli tugmasi(filtrni qo`llash yoki o`chirish) yordamida filtrlaganda sodir bo`ladi. Bu hodisa qayta ishlash programmasini odatda filtrda shart tekshiruvi uchun yoki forma ko`rinishini filtr qo`llashdan oldin o`zgartirish kerak bo`lsa ishlatiladi agar ortiqcha maydonlarni berkitish yoki aksincha berkitilganlarni ko`rsatish kerak bo`lsa. Hodisa qayta ishlash programmasi ikkita:Cancel va ApplyType parametrga ega. Cancel filtrlash operatsiyasini, misol uchun shart noto`g`ri berilgan bo`lsa to`xtatish uchun ishlatiladi. Buning uchun unga true qiymatini uzlashtirish kerak. ApplyType bajarilayotgan harakatni aniqlaydi va 0, 1, 2 qiymatlarini qabul qiladi. 0-filtr o`chirilishiga ishora qiladi, 1-filtr qo`llanishiga, 2-filtr oynasi yopilishini ko`rsatadi.

Hodisa filtrlash operatsiyasini Filtr qo`llash(ApplyFilter), Formani ochish (OpenForm), Hamma yozuvlarni ko`rsatish (ShowAllRecords), bularga mos usullar ob'yekti DoCmd xamda Закрыть makrokomandasi yordamida filtr oynasini yopishda, makrokomandalari yordamida bajarishda sodir bo`lmaydi.

Filtrlash(Фильтрация)

Filtrlash hodisasi(**Filter**) filtr oynasini yoki kengaylashtirilgan filtr ochilishi bilan, foydalanuvchi Filtrni o`zgartirish(Filter by Form) buyrug`ini bajarmoqchi bo`lganda ochiladi. Bu hodisani ishlatish juda qulay, agar misol uchun ko`zda tutilgan xolda filtrga ba'zi shartlar kiritilsa yoki tanlov shartlariga ba'zi maydonlarni qabul qilishni taqiqlab qo`yish kerak bo`lsa. Tanlov shartiga biror maydonni filtr oynasida qo`shishni taqiqlab qo`yish uchun uni Filtrlash hodisasi(Filter)da yopib qo`yish yetarli. Bu faqat oddiy filtr oynasiga taaluqli, chunki kengaytirilgan filtrda hamma maydonlar chikariladi, shu bilan birga yopiqqlari xam.

Xattoki filtr standart oynasini o`zining foydalanuvchi tanlov shartlarini beruvchi oynasi bilan almashtirish mumkin. Hodisa

qayta ishlash protsedurasi 2 ta parametrga ega. Bular - Cancel va FilterType. Cancel filtr standart oynasini ochish imkonini beradi, agar uning o`rniga maxsus forma chikarilsa. Buning uchun uning unga true qiymatini berish yetarli. FilterType parametri qaysi oyna ochilayotganligini aniqlaydi va 0 va 1 qiymatlarini qabul qiladi. 0-oddiy filtrligini bildiradi, 1 - kengaytirilgan filtni.

Oyna hodisalari

Oyna hodisalari forma va hisobot ochilishida va yopilishida, xamda forma o`lchamlarini o`zgartirishda ishga tushiriladi.

Ochilish(Открытие)

Ochilish(**Open**) hodisasi forma yoki hisobot asosida yotuvchi so`rov bajarilgandan so`ngsodir bo`ladi, lekin birinchi yozuvdan so`ngyoki hisobot chop etilishidan keyin. Bu hodisa qayta ishlash protsedurasi bitta Cancel parametriga ega. Unga True qiymati o`zlashtirishda forma yoki hisobot ochilishi bekor kilinadi. Odatda Ochilish(Open) hodisasi qayta ishlash protsedurasi shartlarni tekshirish uchun va forma ochilishini oldini olish uchun ishlatiladi, chunki vaqt bo`yicha keyingi Yuklanish(Load) hodisasi, endi bekor kilina olmaydi.

Yopilish(Закрытие)

Yopilish(**Close**) hodisasi forma erandan yo`qolishidan oldin oxirgi hisoblanadi. Odatda uni boshqa forma ochilishi uchun ishlatiladi. Hisobot uchun hodisa Oldindan Ko`rib chiqish(Предварительный Просмотр) rejimi yopilganda yoki hisobot chop etilishi tugaganda sodir bo`ladi. Xuddi forma kabi uni foydalanuvchini keyingi harakatlarini aniqlash uchun ishlatiladi.

Yuklanish(Загрузка)

Yuklanish(**Load**) hodisasi Ochilish(open) hodisasidan so`ngdarhol sodir bo`ladi, lekin undan farqli bekor klinmaydi. Odatda uni forma ekranga chiqishidan oldin, forma xossalarining yoki boshqarish elementlarini dinamik o`zgartirish uchun ishlatiladi.

Yuklanishni qaytarish(Выгрузка)

Yuklanishni qaytarish(**Unload**) hodisasi formani hisobotdan oldin yopilishida sodir bo`ladi va bekor qilinishi mumkin.

Odatda bu hodisa formani yopish mumkinligini aniqlaydigan, xar xil shartlarni tekshirish uchun ishlatiladi.

Bu hodisani qayta ishlash protsedurasi bittata Cancel parametriga ega. Agar unga True qiymati berilsa forma yopilishi bekor qilinadi.

Agar Cancel parametriga True qiymati o`matilgan, Yuklanishni qaytarish(Unload) hodisasini ishlatiyotgan bo`lsangiz, forma yopilishi uchun hamma shartlarni bajarish kerak bo`lsa, unga False qiymatini o`matishni unutmang. Aks xolda bu parametriga True qiymati o`matilgandan so`ngformani hech kachon yopib bo`lmaydi.

O`lchamni o`zgartirish(Изменение размера)

O`lchamni o`zgartirish(**Resize**)hodisasi forma ochilishida va uning o`lchamlari o`zgarishida sodir bo`ladi. Uni odatda, boshqarish elementlarini o`lchamini formaning o`zgarayotgan o`lchamlariga moslashtirish kerak bo`lsa yoki qaytadan hisoblanayotgan elementlarni hisoblash uchun ishlatiladi. Agar formaning o`lchamlari xar o`zgarishida ekran yangilanishi kerak bo`lsa, bu hodisa qayta ishlash protsedurasida Repaint metodidan foydalaning.

Ku p xolda hodisalar qayta ishlash protseduralari yaratilishida, qaysi ikki hodisa -Ochilish(Open) hodisasi yoki Yuklanish(Load) hodisasini tanlash(yoki mos ravishda Yopilish(Close) hodisasi yoki Yuklanish(Load) hodisasi) xaqida shubxalar tug`iladi. Quyidagi farazni ko`zda tutish tavsiya kilinadi. Agar hodisani bekor qilish ikoni kerak bo`lsa Ochilish(Open) hodisasi yoki Yuklanish(Load) hodisasini ishlating aks xolda xoxlaganingizni ishlating.

Xatolar hodisalari

Xato(**Error**) hodisasi forma yoki hisobotni Access yadrosi bilan qayta ishlash jarayonida xato tug`ilsa sodir bo`ladi. Bu hodisa qayta ishlash protsedurasida Access beruvchi xato xaqidagi standart xabarni ushlash va bu xakda xabar berish mumkin. Protsedura ikkita parametrغا ega-DataErr va Response DataErr parametri xato kodini saklaydi. Response parametri esa ikkita qiymatga ega-0 va 1. Bulardan 0 xato xaqidagi standart xabarni bekor qiladi va 1 ko`rsatish imkonini beradi. Bu hodisa VBA kodida uchraganda chiqmaydi.

Taymer hodisasi

Taymer(**Timer**) hodisasi xar gal vaqt интервалidan so`ngsodir bo`ladi. U Таймер интервалі (TimerИнтервал) xossasi bilan beriladi. U taymer signali bo`yicha davriy ravishda sodir bo`lishi kerak bo`lgan harakatlarni aniqlashga imkon beradi. Odatda ko`p foydalanuvchili prilojeniyalarida ekranni xar gal yengilash uchun ishlatiladi. Bunda Taymer(Timer) hodisasini qayta ishlash protsedurasida Requery metodini ishlatish kerak. U formaning ma'lumotlar manbayining qayta so`rovini bajaradi.

ACCESS hodisalari ketma-ketligi

Hodisani qayta ishlash protseduralarini yozishda ular qaysi tartibda bajarilishini tushinish judayam muhim, chunki Access ilovasi hodisalar bilan boshqariladi va ish natijasi bu protseduralarni bajarilish tartibiga bog`liq.

Forma boshqarish elementlari uchun fokus hodisalari ketma-ketligi.

Fokusni boshqarish elementiga sichqoncha tugmasini bosish bilan o`rnatishda, <Tab> tugmasini bosishda yoki formani ochishda bajariladi:

Kirish (Enter)=>Fokus olish (GotFocus).

Element fokus yo`qotganda, misol uchun forma yopilishida yoki fokusni shu formaning boshqa boshqarish elementiga utganda quyidagi hodisalari sodir bo`ladi:

Chiqish(Exit)=>Fokusni yo`qotish (LostFocus)

Fokusni boshqa formaga o`tkazishda birinchi forma elementi fokus yo`qotadi, lekin Chiqish (Exit) hodisasi sodir bo`lmaydi. Mos ravishda fokus orkaga qaytishida, ya'ni forma aktiv bo`lganda fokusga ega bo`lgan elementga utsa Kirish (Enter) hodisasi bajarilmay, faqat Fokus olish (GotFocus) hodisasi ro`y beradi.

Boshqarish elementidagi ma'lumotlar o`zgarish hodisalar ketma-ketligi

Boshqari elementga ma'lumotlar kiritilganda yoki o`zgartirilganda, so`ngesa keyingi boshqarish elementiga utilsa, quyidagi hodisalar zanjiri sodir bo`ladi:

{Klavisha pastga=> Klavisha bosilishi =>O`zgarishlar kiritilgan=> O`zgarish => Klavisha tepaga}> Yangilashgacha =>Yangilashdan so`ng=>Chiqish=> Fokusni yo`qotish.

Keltirilgan ifodadagi figurali qavslar, ajratilgan hodisalar zanjiri klaviaturadagi tugmalarni xar gal bosishda sodir bo`lishini anglatadi.

Yozuvlar bilan bog`liq forma hodisalari ketma-ketligi

Agar faqatgina formada yozuvlar ko`rilayotgan bo`lsa, u xolda xar yangi yozuvga o`tishda forma elementlaridagi fokus o`rnatilishi bilan bog`liq bo`lgan forma va hisobot uchun forma elementlarida Joriy yozuv (Current) hodisasi sodir bo`ladi. Agar yozuvda ma'lumotlar o`zgarayotgan bo`lsa, u xolda uzgrishlarni saqlab kolish faqat keyingi yozuvga o`tishda yoki forma yopilishida(agar kerakli yozuvni topgan bo`lsangiz o`zgartirib formani yoping) sodir bo`ladi. Yozuvdagi hamma o`zgarishlar formani boshqarish elementida sodir bo`lgani uchun bir vaqtda forma elementlarida mos hodisalar ro`y beradi.

Shunday xolatni ko`rib chiqamiz, formada yozuv chikkanda uning boshqarish elementlari bo`yicha kerakli elementgacha kuchadi, undagi ma'lumotlarni o`zgartiradi va keyingi yozuvga utadi. Bu xolda hodisalar chiqish ketma ketligi quyidagi ko`rinishda bo`ladi:

Joriy Yozuv (forma)> {Kirish(element)>Fokus olish(element) =>Chiqish(element)> Fokusni yo`qotish(element)}> Kirish(element)>Fokus olish (element)> Yangilashgacha(element)> Yangilashdan so`ng(element)> Yangilashgacha(forma)> Yangilashdan so`ng(forma) => Chiqish(element) => Fokus yo`qotish(element) => RecordExit => Joriy Yozuv (forma) => Kirish(element) => Fokus olish (element)

Bu zanjirda soddalik uchun klaviaturaning hamma hodisalari olib tashlangan, va figurali qavslar esa forma boshqarish elementlari orasida sodir bo`ladigan hodisalar zanjirini ajratadi. Ma'lumotlarini o`zgartirishda Yangilashgacha(BeforeUpdate) va Yangilashdan so`ng(AfterUpdate) hodisalari avval element uchun so`ngforma uchun sodir bo`ladi. so`ngoxirgi element (bizning xolda o`zgarishlar sodir bo`lgani) fokus yo`qotadi, RecordExit(Выход из Записи) hodisasi ro`y beradi, keyingi yozuv chiqadi va fokus shu yozuvning birinchi elementiga o`rnatiladi. Yangilashgacha(BeforeUpdate) va Yangilashdan so`ng(AfterUpdate) hodisalari xar gal keyingi yozuvga o`tishda sodir bo`ladi. Bundan so`ng yozuv (bundan avval fokusi bo`lgan boshqarish elementi) fokus yo`qotadi.

Agar yozuv Запись menyusining Сохранить запись(Records, Save Records) buyrug`i bilan saklanayotgan bo`lsa u xolda oxirgi Chiqish (Exit) va Fokus yo`qotilishi(LostFocus) hodisasi ruy bermaydi. Buni bilish bu hodisalarni chetlab o`tish kerak bo`lganda kerak bo`ladi.

Yozuvni o`chirishda quyidagilar sodir bo`ladi:

O`chirish=> Del tasdiqlanganicha=> Del tasdiqlanganidan so`ng=> Joriy Yozuv => Kirish=> Fokus olish

Agar O`chirish (Delete) hodisasi bekor qilinsa, u xolda qolgan hodisalar sodir bo`lmaydi. Yozuv o`chirilgandan so`ngfokus keyingi yozuvga o`tadi, shuning uchun forma Joriy Yozuv (Current) va Kirish (Enter) va shu yozuvning birinchi elementi Fokus olish (GetFocus) hodisalari yuz beradi. Yangi yozuv qo`shish yangi yozuv, birinchi simvolini kiritgandan so`ngsodir bo`ladi. Bunda hodisalar quyidagi tartibda sodir bo`ladi:

Joriy Yozuv (forma)> Kirish (element) =>Fokus olish (element) =>Joylashdan oldin (forma) =>O`zgartirish (element)>Yangilashgacha (forma) =>Yangilashdan so`ng (forma) =>Joylashdan so`ng (forma)

Bu zanjirda klaviatura va forma elementlari uchun Yangilashdan oldin (BeforeUpdate), Yangilashdan so`ng (AfterUpdate) hodisalari qo`yib yuborilgan, chunki ular bundan avval ta'riflagan. O`zgartirish (Shange) hodisasi agar yangi yozuv birinchi simvoli tekst maydoniga kiritilsa sodir bo`ladi.

Formadagi hodisalar ketma-ketligi

Forma ochilishida quyidagi hodisalar sodir bo`ladi:

Ochilish=> Yuklanish=>Kattalikni o`zgartirish =>Ulanish=>Joriy Yozuv =>Kirish =>Fokus olish

Kirish (Enter) va Fokus olish (GetFocus) hodisalari birinchi yozuv birinchi elementi uchun sodir bo`ladi.

Agar forma ko`rinarli yoki mumkin bo`lgan elementlarga ega bo`lmasa, hodisalar ketma-ketligi boshkacharak:

Ochilish=> Yuklanish=>Kattalikni o`zgartirish =>Ulanish=>Fokus olish =>Joriy Yozuv

Bu xolda Fokus olish (GetFocus) hodisasi formaga taaluqli bo`ladi

Forma yopilishida hodisalar ketma-ketligi quyidagicha:

Chiqish=>Fokusni yo`qotish =>Yuklanishni yo`qotish=>Uzilish=>Yopilish

Ya'ni avval elementni formadagi oxirgi element yo`qotadi, so`ngforma uchun hodisalar bajariladi. Agar formada mumkin bo`lgan element yoki ko`rinarli bo`lmasa, ketma-ketlik boshkacha:

Yuklanishni yo`qotish=> Fokusni yo`qotish (forma)> Uzilish=>Yopilish

Ikkita forma orasida o`tishda ketma-ketlik quyidagicha:

Fokusni yo`qotish (forma elementi 1) => Uzilish (forma 1) =>Ulanish (forma 2) => Kirish (forma elementi 2) => Fokus olish (forma elementi 2)

Agar ochiq formadan boshqa forma ochishsa, u xolda ochilayotgan formada Ochilish (Open), Yuklanish (Load) va Kattalikni o`zgartirish (Resize) hodisalari sodir bo`ladi va faqat shundan so`ngbirinchi formada Uzilish (Deactivate) hodisasi yuz beradi:

Ochilish (forma 2)> Yuklanish (forma 2) =>Kattalikni o`zgartirish (forma 2) =>Uzilish (forma 1) =>Ulanish (forma 2) =>Joriy Yozuv (forma 2)

Bu ikkinchi forma ochilishini boshqarish imkonini beradi. Bundan tashqari Ulanish (Astivate) hodisasi xar gal, forma fokus qabul kilganda sodir bo`ladi, Ochilish (Open) va Yuklanish (Load) hodisalari esa sodir bo`lmaydi, agar forma ochilgan bo`lsa va xattoki bu formaga o`tish Formani ochish (OpenForm) makrokomandasi yordamida bajarilayotgan bo`lsa xam. Forma uchun Uzilish (Deactivate) hodisasi bajarilmaydi agar fokus dialog oynasiga ko`chirilayotgan bo`lsa xam.

Agar ochilayotgan forma tobe formalarga ega bo`lsa, u xolda avval Включение(Activate) hodisasidan tashqari odatda forma ochilishida sodir bo`ladigan tobe formalar ochiladi va undagi hamma hodisalar sodir bo`ladi. so`ngodatiy tartibda bosh forma yuklanadi va unga Включение(Activate) hodisasi yuz beradi. Shu kabi shunday forma yopilishida ham avval hamma tobe

formalar yuklanadi, lekin ularda Uzilish (Deactivate) hodisasi yuz bermaydi. Hodisalar quyidagi tartibda bajariladi:

Tobe forma elementlari uchun hodisa. Misol uchun Chiqish (Exit) va Fokus yo`qotish (LostFocus).

Bosh forma boshqarish elementlari hodisasi, shu bilan birga tobe formani tashkil etuvchi elementlar ham shu jumladan.

Bosh forma uchun hodisa.

Tobe forma uchun hodisa.

Klaviatura va sichqoncha hodisalari ketma-ketligi

Klaviaturada ixtiyoriy klavisha bosilganda yoki qo`yib yuborilganda, agar fokus forma boshqarish elementlarinig birida turgan bo`lsa, quyidagi xodisalar zanjiri xosil bo`ladi:

Klavisha pastga => Klavisha bosilishi => Klavisha tepaga

Sichqoncha tugmasini forma boshqarish elementida bosganda mos ravishda:

Tugma pastga => Tugma tepaga => Tugma bosilishi

Hisobotdagi hodisalar ketma-ketligi

Hisobotda hodisa faqat hisobot o`zi uchun va uning bo`limlari uchun sodir bo`ladi. Hisobot maydonlari uchun hodisa bo`lmaydi. Hisobotni chop etishda hodisa quyidagi ketma-ketlikda sodir bo`ladi:

Ochilish=>Ulanish=>{Formatlash =>Chop etish} =>Yopilish=>Uzilish

Formatlash (Format) va Chop etish (Print) hodisalari hisobot bo`lim lariga tegishli va xar bir bo`lim qayta ishlanishida sodir bo`ladi. Qo`shimcha hodisalar quyidagi hodisalar bo`lishi mumkin:

Ma'lumotlar yo`qligi(NoData). Bu birinchi Chop etish (Print) hodisasidan so`ngsodir bo`ladi agar yozuvlar anbas i yozuvga ega bo`lmasa.

Sahifa(Page). U hisobot sahifasi uchun hamma Chop etish (Print) hodisasidan so`ngsodir bo`ladi

Qaytarish (Retreat) hodisasi sahifa uchun Chop etish (Print) hodisasidan oldin, avvalgi bo`lim ga qaytish kerak bo`lganda hisobot sahifasini formatlash jarayonida sodir bo`ladi.

Hodisalarni qayta ishlash protsedurasini yaratish

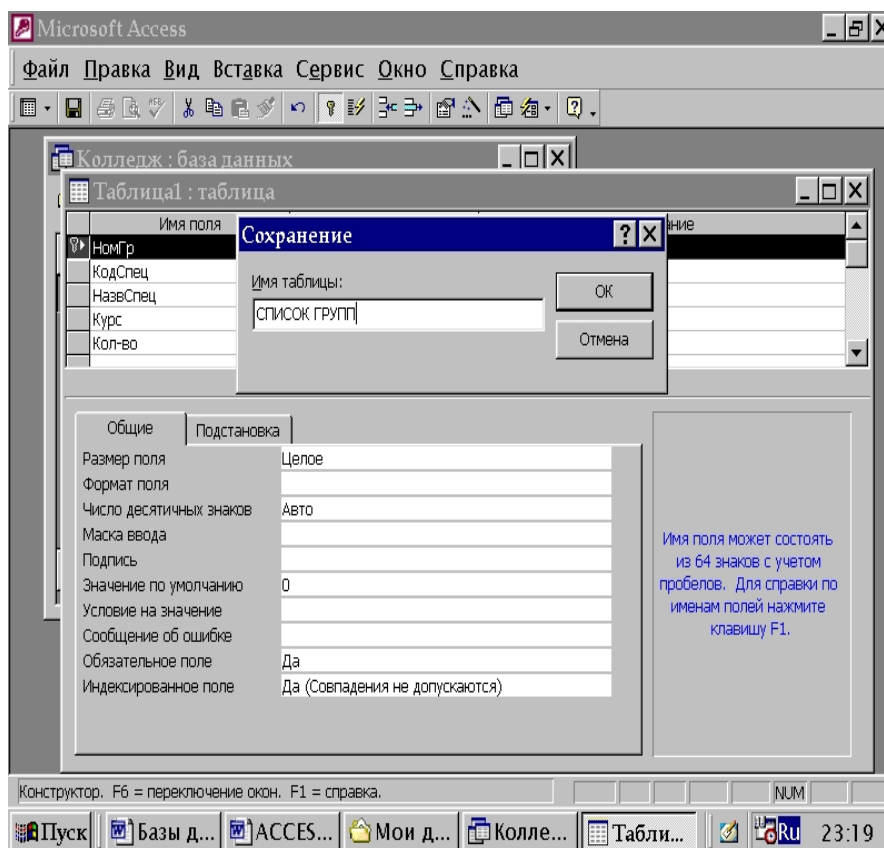
Hodisalarni qayta ishlash protseduralarini yaratishda ko`pchilik forma boshqarish elementlari uchun xamdaformaning uziga va hisobot uchun standart harakatlar quyidagilar:

Конструктор rejimida forma oching. Agar bunda xossalar oynasi ekranda bo`lmasa, panel insturmentovdagi Свойства(Properties) tugmasini bsoing.

Kerakli boshqari elementini tanlang(yoki sichqoncha bilan forma chap yuqori burchagida kora kichkina kvadrat tugmasini bosing, shunda butun forma chiqadi). Xossalar oynasida tanlangan element xossalari ko`rinadi.

События qo`shimcha sahifasini oching(Events). Qayta ishlash protsedurasi yaratilishi kerak bo`lgan hodisani tanlang va unda sichqonchani o`ng tugmasini bosing.

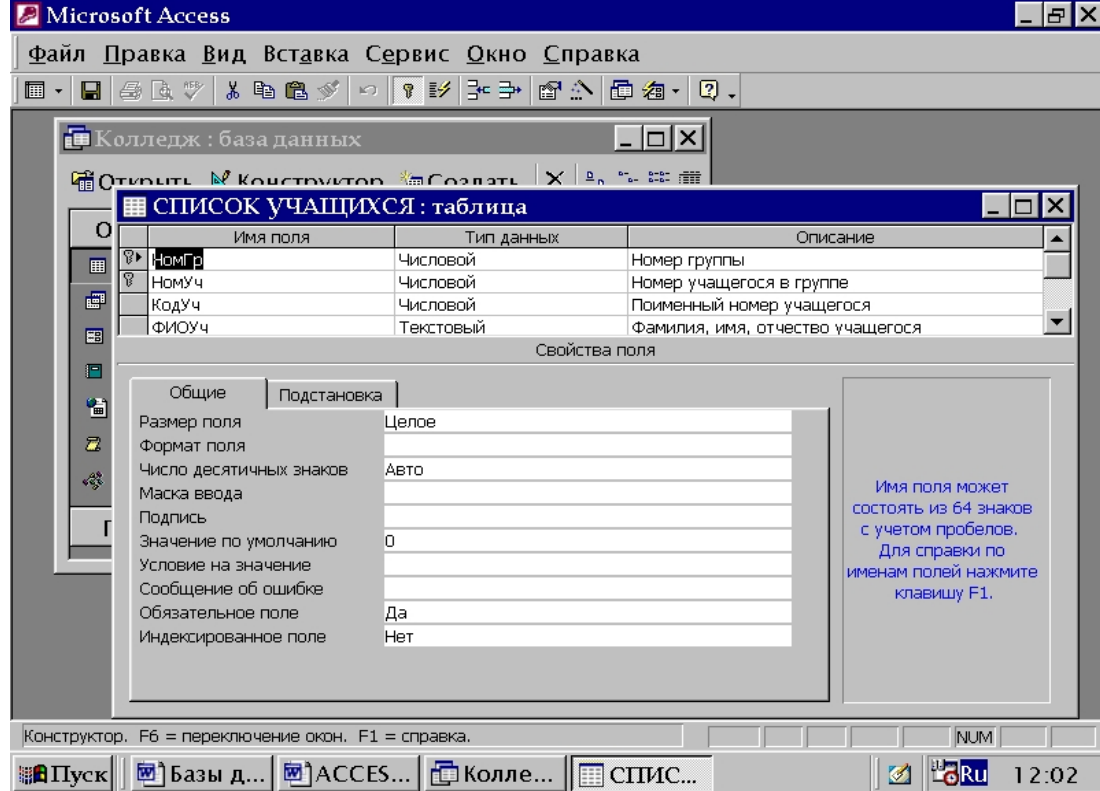
Kontekst menyusidan Yaratish (Построить(Build) punktini tanlang. Ochilgan Yaratuvchi(Построитель)(Choose Builder) dialog oynasida elementlar ro`yxatidan Dastur elementi(Элемент Программы)(Code Builder)ni tanlang va OK Kнопkasini bosing. VBA redaktor oynasini oching. Unda protsedurani birinchi va oxirgi qatorlari chiqadi.



Forma hodisalari dialog oynasi

Agar tanlangan hodisa qayta ishlash protsedurasi argumentga ega bo`lsa, ular ham protsedura sarlavhasida chiqadi.

Endi shu ikki qator orasiga protsedura kodini kiritish kerak. Pastda hodisa qayta ishlash protseduralariga misollar keltirilgan.



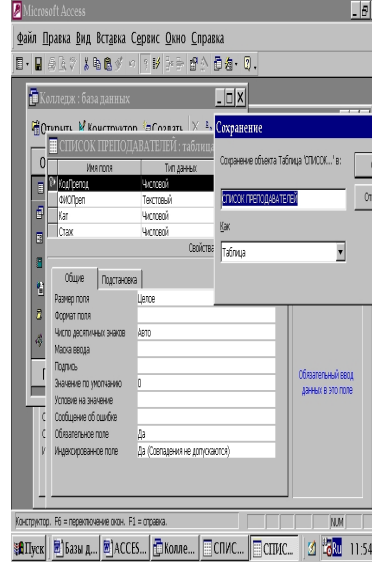
ACCESS dasturlash muhiti

Access 2002 dasturlash muhiti VBA protseduralarini sozlash dasturlarini yozish uchun kerak bo`ladigan hamma vositalarga ega. Bu muhit eng muhim komponentasi Visual Basic Editor(VBE) kodi redaktori hisoblanadi. U programma kodini kiritishni yengillashtiradigan ko`p imkoniyatlarga ega. Bundan tashqari dasturlash muhiti yana ko`pgina dialog oynalariga ega. Ularda ham yaratishda, xamda kodni sozlashda qo`shimcha ma'lumotlar ko`rinadi.

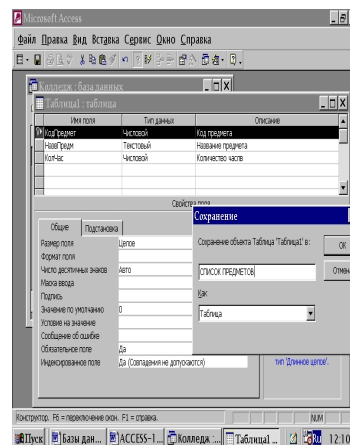
Kodni tahrirlash oynasi

Tahrirlash oynasini ochish uchun Accessning ixtiyoriy modulini ochish yetarli. Odatda tahrirlash oynasida uchta panel ishlatiladi.

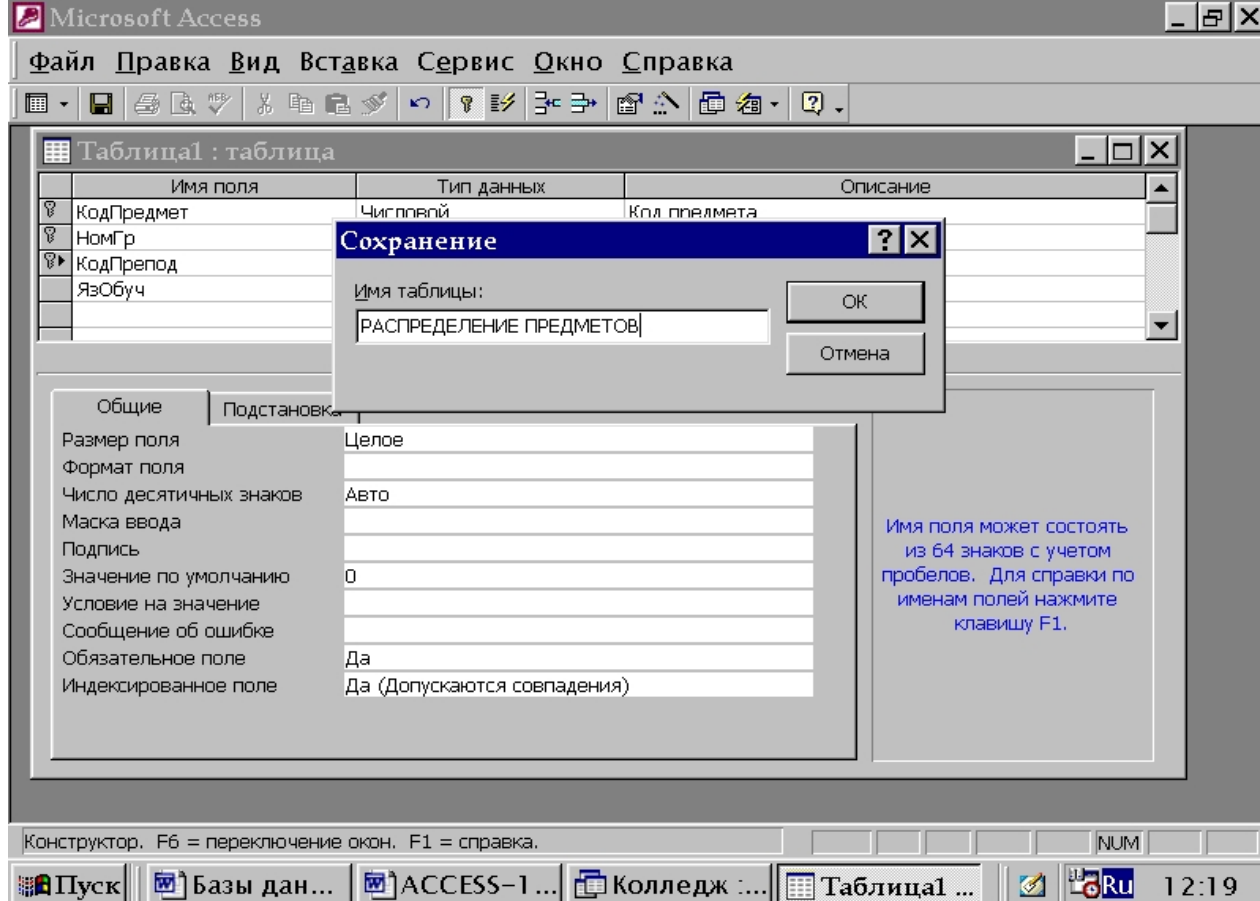
Project (Loyiha paneli) tahrirlovchining tepa chap burchagida joylashgan bo`ladi. Unda ilovalar modullarining ierarxik daraxti aks etadi. Agar bu oyna noaktiv bo`lsa, View, Project Explorer komandasini bajaring yoki <Ctrl>+<R> tugmalari kombinatsiyasini bosing.



Properties (Xossalar paneli), loyiha panelining tagida joylashgan. U loyihaga kiruvchi xar xil ob'ektlar xossasini o'zgartirish va ko'rish imkonini beradi. U loyihalar panelida aks etadi. Xossalar ro'yxati ham alfavit tartibda, ham kategoriyalar bo'yicha aks etishi mumkin.



Code(Kod tahriri paneli). Bu oyna ekranning katta qismini egallaydi va "ko'pdokumentli" hisoblanadi, ya'ni bir vaqtda shu turdagi bir nechta oynani xar xil modul uchun ochish imkonini beradi. U yuqori intellektual protsessor bo'lib, VBA kodi yozilishini ancha yengillashtiradi.



Ilova loyihasi modullardan tashkil topgan. Ular 3 kategoriyaga bo`linadi:

Microsoft Access Class Object(Access sinflari moduli)- forma va hisobotlarning hamma modullarini o`z ichiga oladi;

Modules(Modullar)-standart moduli;

Class Modules(Sinflar moduli) - foydalanuvchilar moduli sinflari, agar ular ilovalarda qatnashsa.

Loyiha oynasidagi ob'ektlar ro`yxati ierarxik bo`lib, loyiha modullaridan tashqari, u tashqi bibliotekadagi ob'ektlarga va bu biblioteka modullari ro`yxatiga ko`rsatkichlarni o`z ichiga olishi mumkin.

Loyiha paneli yuqori qismida uchta tugma joylashgan:



View Code(Kodni ko`rish). Bu tugmani bosishda sichqoncha ko`rsatkichi oyna kodiga kuchadi. Unda loyiha ob'yekti ierarxiyasida ajratilgan kod aks etadi.

View object(Ob'yektni ko`rish). Bu tugma loyiha daraxtida faqat forma yoki hisobot moduli ajratilgan bo`lsa murojaat qilish mumkin. Shunda u mos ob'yekt moduliga, ya'ni forma yoki hisobot Конструктор oynasiga o`tish imkonini beradi.

Toogle Folder(Daraxt tasvirlanishini o`tish turlari). Bosilgan tugmada loyiha daraxtidagi hamma elementlar papkalarda kategoriyalar bo`yicha guruhlanadi, bosilmay turgan tugmada esa ular alfavit tartibda bo`ladi.

Xossalar paneli loyihaga kiruvchi xar xil ob'ektlar xossalarini ko'rish va o'zgartirish imkonini beradi. Ob'yekt xossalarini ko'rish uchun uni doyixa daraxtida ajratish kerak yoki, forma(hisobot) Konstruktori oynasida. Ko'pincha bu panel forma va hisobotdagi Xossalar(Свойства)(Properties) dialog oynasini ikkilaydi(dubliruet).

U xolda xam, bu xolda ham ob'yekt xossasi o'zgarishi statik, shuning uchun ular boshlang'ich yoki o'zgarimas ob'yekt xossalarini berish uchun ishlatilishi mumkin.

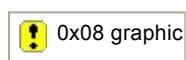
Ob'yekt xossasi qiymatini o'zgartirish uchun:

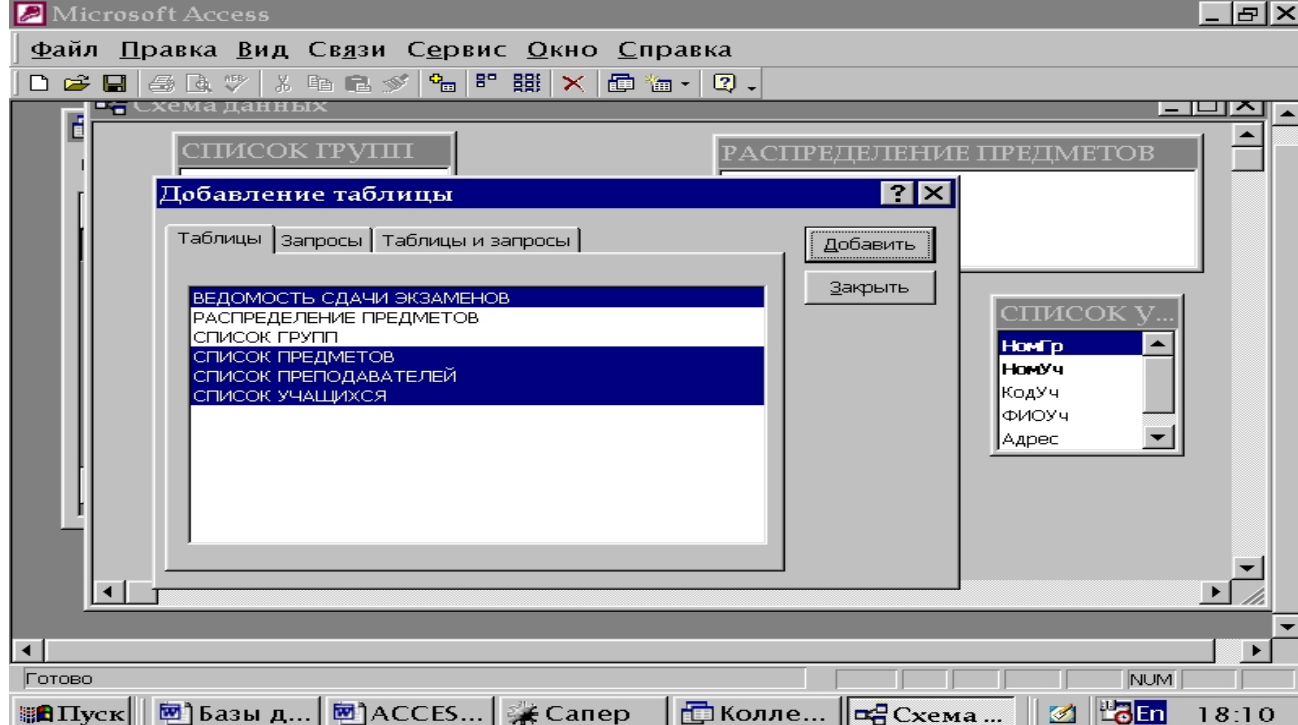
Chap ustun xossa nomini ajratish.

O'ng ustun qiymatini uni qo'l bilan kiritib yoki ro'yxatdan tanlab o'zgartirish.

Xar bir moduldagi kod bo'limlardan tashkil topgan. Ular orasidagi o'tish(переключение) Object(Ob'yekt) ro'yxatidan qiymat tanlash yo'li bilan bajariladi. U chap yuqori oynada joylashadi. Bo'limlar ro'yxati xar xil turdagi modullar uchun turlidir.

Forma moduli uchun bu ro'yxat General(Общая область) bo'limini o'z ichiga oladi. U o'z ichiga o'zgaruvchilar, o'zgarimaslar, maxsus turlar, tashki protseduralar e'lonini oladi. Bu bo'lim kod oynasini ochishda po umolchaniyu ochiladi. Standart modul uchun bu bo'lim yagona-u hamma protseduralar ta'rifini o'z ichiga oladi. Sinf moduli General va Class bo'lim lariga ega. Forma moduli General bo'limidan tashqari forma o'zining bo'limiga va xar boshqarish elementlari uchun bo'lim larga ega. Hisobot moduli o'z ichiga Report(Hisobot) ob'yekti uchun bo'limni va hisobot xar sohasi uchun bo'limlarga ega. Hisobot maydonlari uchun hodisalar aniqlanmagan, shuning uchun kod modulidagi mos bo'limlar ham yo'q.. Xar bir bo'lim o'ng yuqori oynada Procedure ro'yxatidan tanlanadigan bir nechta bo'lim ni o'z ichiga olishi mumkin. Standart modulda protseduralar ro'yxati Declarations(Tavsiflar) o'z ichiga oladi va modul hamma protseduralar nomlarini alfavit tartibda o'z ichiga oladi. Forma moduli uchun protseduralar ro'yxati o'z ichiga forma hamma hodisalari va elementlari ro'yxatini oladi. Protседuralar ro'yxati, modul klassi klass hodisalari uchun faqat ikkita protseduraga ega: Initialize va Terminate.





Oynaning chap pastki burchagida joylashgan tugmalar yordamida kodni ekranda tasvirlash turlaridan birini tanlash mumkin

Procedure View(Protsedura ko`rinishi), oynada faqat bitta tanlangan protsedura aks etsa.

Full Module View(To`liq modul ko`rinishi(predstavleniya))oynada bir-biridan ingichka chiziqli bilan ajratilgan bir nechta protsedura birdaniga aks etsa.

"Параметры"(Parametrlar) oynasi

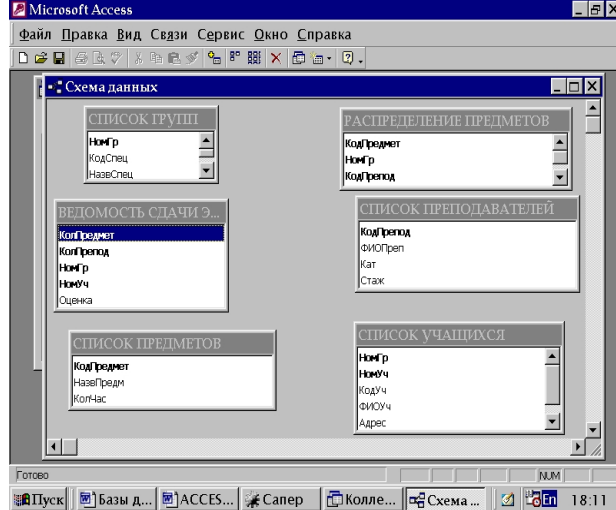
Tahrirlovchi parametrlarini to`g`rilash uchun Options(параметры) oynasi ham ishlatiladi. Uni Tools menyusidagi Options buyrug`ini tanlab ochish mumkin. Bu faqat kod tahrirlovchisini parametrlarini ta'rifiyladi va kod tahrirlovchisi muhitidan chaqiriladi.

U to`rtta qo`shimcha sahifaga ega.

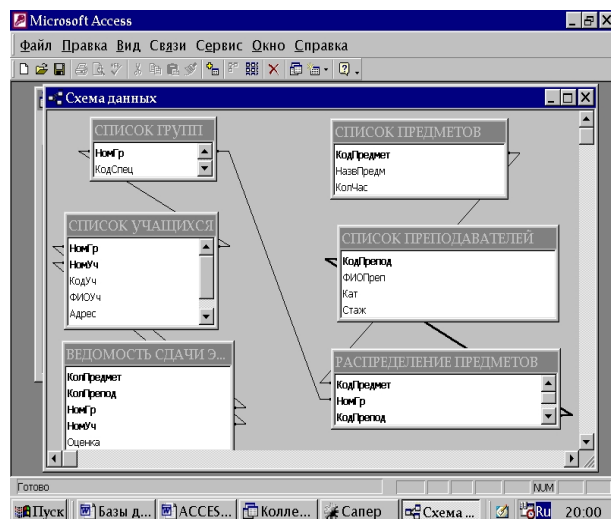
Birinci Editor qo`shimcha sahifasi ikkita parametrlar guruhidan tashkil topgan:Code Settings va Window Settings.

Code Settings guruhi(dasturlash) oltita bayroqchaga ega. Ular muharrirning intellektual funksiyalarni o`chirish va yoqish imkonini beradi, ya'ni avtootstup, sistaksisni avtomatik tekshirish, foydalanuvchi dastur yozishida yordam(podskazka).

Window Settings o`z ichiga uchta bayroqchani oladi. Ular kodni muharrir oynasida namoyish qiladi. Bu bayroqchalar to`liq modul va protseduralar tasavvurlari orasida qayta o`tish, ekranga protseduralarni ajratuvchi ingichka chiziqlarni chiqarish yoki chiqarmaslik, matnni ko`chirishi funksiyasini o`chirish yoki yoqish, ya'ni sichqoncha bilan ajratilgan kod bo`lagini ko`chirish imkonini beradi.



Ikkinchi qo`shimcha sahifa **Editor Format**(Muharrir formati) matnni xar xil sohalaridagi shrift, o`lchami va rangini aniqlash imkonini beradi. Ixtiyoriy modulni ochib, siz dastur matni bir necha rangda berilganligini ko`rishingiz mumkin:ko`zda tutilgan bo`yicha izoh yashil rangda, kalit so`zlar ko`k rangda beriladi. Dasturni sozlashda matn qismlarini ajratishda boshqa ranglar ham ishlatiladi:to`xtash nuqtasi, joriy buyruq va boshqalar. Bu hamma ajratish usullari shu oynadagi sizning didingizga mos ravishda to`g`rilanishi mumkin.



Uchinchi qo`shimcha sahifa **General(Общая)** bir nechta qayta ulagich va bayroqchalar guruhiga ega.

Form Grid Setting guruhi Microsoft Visual Basic 6.0 va Office boshqa komponentalarida forma yaratilishga taaluqli va Accessda ishlatilmaydi.

Show ToolTips bayroqchasi instrumentlar panelidagi tugmalar uchun so`zib chiquvchi yordamchi matnlarni yoqish yoki o`chirish imkonini beradi.

Collapse Proj. Hides Windows bayroqchasi loyiha panelidagi loyiha yopilishida muharrir oynasini ochilishini aniqlaydi.

Notify Before State Loss bayroqchasi, qilinayotgan harakatlar joriy loyiha xolatini yo`qolishiga olib kelishi xaqidagi xabarni foydalanuvchi olish yoki olmasligini aniqlaydi.

Error Trapping qayta ulagichi, dasturdagi xatolarni qayta ishlashda ishlatiladigan ko`zda tutilgan xolda ko`rsatmalarni aniqlaydi:

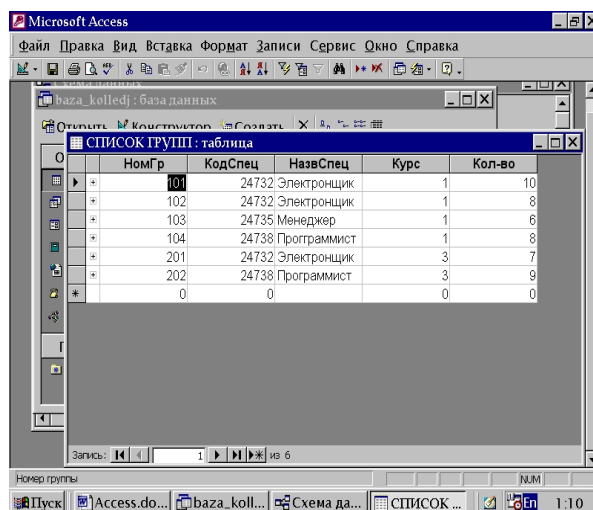
Break on All Errors-dastur tomonidan qayta ishlanayotganligidan qat'iy nazar, ixtiyoriy xato. Dastur uzilishiga olib keladi.

Break in Class Module- Modul klasslaridagi ixtiyoriy qayta ishlanmagan xato. Dastur uzilishiga olib keladi.

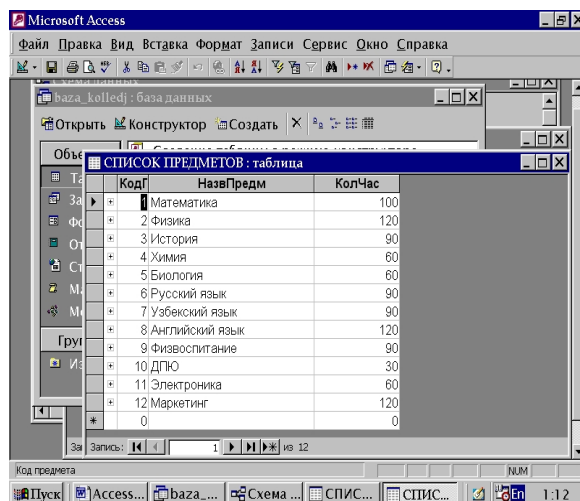
Oxirgi bayroqchalar guruhi modul loyihasi kompilyatsiya qilinayotganligini aniqlaydi.

1. **Compile On Demand**- aniqlaydi, loyihada hamma modullar kompilyatsiyalanadimi ish boshida, yoki faqat talab bo'yichami, birinchi xolda ilovani ishga tushirish ancha vaqt oladi;

2. **Background Compile**-modullar kompilyatsiyasi fon rejimida bajarilishini aniqlaydi;



Doking- oxirgi qo`shimcha varaqasida bayroqchalar asosiy oynalarni ekrandagi xolatini mustaxkamlaydi.



Visual Basic muharriri quyidagi asosiy xususiyatlarga ega :

1. Sintaksisni avtomatik tekshirish - komandani kiritish muvaffaqiyatli; va hamma so`zlar kerakli rangda ajratiladi, faqat jumlaning sintaksisi to`g`ri bo`lsa, aks xolda siz tovush signali eshitasiz, matn qizil rangda ajraladi va xato to`g`risida ma'lumot paydo bo`ladi.

2. **Auto List Members** ning komponentalar ro`yxatining avtomatik chiqishi ob'ektlar xususiyatlari turini, o`z ichiga olgan

konstantalarni, o`zgaruv chan xolini kirgazishda xato qo`yilmaslikning oldini oladi; ayniksa bu funksiyasi xususiyatlarni o`rnatishda foydali, chunki xususiyatning to`g`ri ismini eslabgina qolmay balki ob'yektning qaysi xususiyatlari shu paytda ko`rish oson.

3. **Auto Quick Info**-qisqa ma'lumotlar avtomatik chiqishi xuddi namoyon bo`lgan ichki funksiyalar va makrokomaglalar sintaksisi xaqida qo`shimcha ma'lumot.

Parametrlar kirgizilishida keyingi parametr qalin shriftda ajralib turadi.

Quyidagi ko`rsatilgan funksiyalardan tashqari kontekst menyuda boshqa juda foydali funksiyalar - kodning tez yozilishini ta'minlaydigan List Constants, Parameter Info, Complete Word, yana ikkita komanda, programmalar moduli orasida navigatsiyani juda soddalashtiradigan:

1. **Definition**. Agar protsedurada boshqa protsedura chaqirilishi bajarilsa, unda kursorni chaqirilayotgan protsedurada o`rnatildi va kontekst menyudan komanda Definition tanlanadi(undan ham tezrok tugmalar kombinatsiyasi <Shift>+<F2>ni bosib, siz chaqirilayotgan protseduraning tanasiga tushasiz).

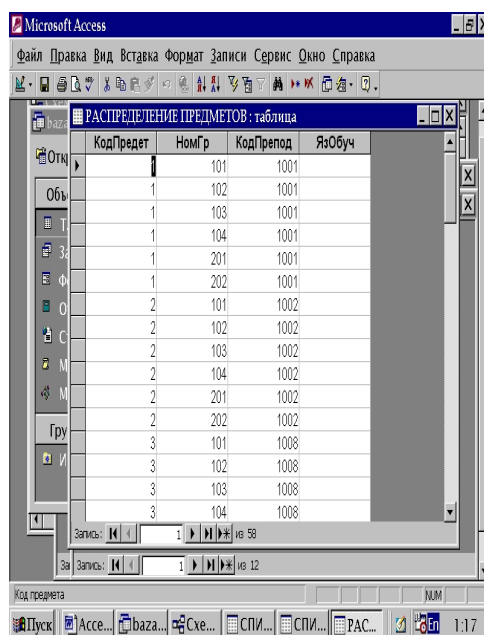
2. **Last Position**-bunday o`zgarishdan so`ngasli xolatga qaytishga ijozat beradi. Definition komanda ham tezda Object Brouser oynasini ochishga va kerakli xususiyatlarni tavsifini, usulini yoki ob'yekt nuqtasiga ijozat beradi. Buning uchun kursorni qiziqtirilayotgan ob'yektga o`rnatiladi, xususiyatlar, usul va <Shift>+<F2> tugmalar kombinatsiyasini bosish kerak.

Sozlash Instrumental vositalari

Sozlash instrumental vositalari dasturlash professional muhitda matnning intellektual redaktoridan tashqari instrumental vositalari ham bor. so`zlashning instrumetal vositalari mutaxassisga uning dasturi kay darajada ishlaganligi xaqida tula ma'lumot beradi. Va mutaxassisning vazifasi kul ostidagi barcha ma'lumotlar asosida tez xatoni tanishdan iborat. so`zlashning vositalar to`plami Accessda juda keng :bu va maxsus menyu Debug(Otladka), ko`p jixatdan uni qaytaruvchi tugmalar instrumentlar kanalida va maxsus sozlash oynalari. Quyida xar bir vositaning qisqa tavsifi berilgan.

Menyu Debug

Menyu Debug (sozlash) va maxsus Debug(sozlash) instrumental paneli rasmda ko`rsatilgan. Komandalar menyusining tavsifi quyidagi jadvalda belgilangan:



Debug menusi komandalari

Komanda	Vazifasi
Compile (Компиляция)	Joriy ma'lumotlar bazasida hamma modullarni kompilyatsiya qiladi
Step Into (Шаг с заходом)	Kod navbatdagi satrini protseduraga kirib bajaradi
Step Over (Шаг с обходом)	Kod navbatdagi satrini protseduraga kirmasdan bajaradi
Step Out (Шаг с выходом)	Joriy protsedura koldigini bajaradi va shu protsedura chakirig`idan so`ng keyingi satrda to`xtab qoladi
Run to Cursor (Запуск до курсора)	Joriy satrdan to kursor turgan satrgacha kod bajariladi. Bu satr oldida to`xtaydi
Add Watch (Добавление контрольного значения)	Kontrol qiymat qo`shish oynasini ochadi
Edit Watch (Изменение контрольного значения)	Kontrol tugmasini o`zgartirish oynasini ochadi
Quick Watch (Быстрый просмотр)	Maxsus oynada to`xtash nuqtasida ifoda qiymatini chiqaradi
Toggle Breakpoint (Установка/сброс точек останова)	To`xtash nuqtasini kursor turgan nuqtaga o`rnatadi/oladi
Clear All Breakpoints (Сброс всех точек останова)	Moduldagi hamma to`xtash nuqtalarini o`chiradi
Set Next Statement (Установка следующего предложения)	Navbatdagi bajariladigan komandani kursor turgan satrga o`rnatish
Show Next Statement (Показ следующего предложения)	Muharrir oynasida bajarish uchun navbatdagi komandani ko`rsatadi

Instrumental paneldagi tugmachalar asosan tavsif qilingan komandani qaytaradi. Bu tugmachalardagi va komanda yonidagi belgilardan ko`rinib turibdi. Undan tashqari bu yerda Run menyusiga taaluqli tugmalar bor:

RunSub/User Form-protseduraning to`xtash nuqtasidan so`ng bajarilishini davom ettiradi, qadamma qadam rejimni yechib, agar u o`rnatilgan bo`lsa;

Break-protseduraning bajarilishini to'xtatish ;

Reset-protseduraning bajarilishini to'xtatadi va boshlang'ich ma'nosini o'zgaruvchanligini o'zlashtiradi.

Instrumental panelning yana bir necha tugmachalari so'zlashning ixtisoslashtirilgan oynasini ochishga ijozat beradi.

Locals oynasi

Sozlashning ixtisoslashtirilgan oynasi foydalaniladi asosan berilganlar oraliqdagi qiymatini ko'rish uchun: o'zgaruvchilar, ifodalar, ob'ektlar xossalari va hokazo.

Ularga ikkita oyna kiradi: **Locals**(lokal) va **Watches**(kontrol).

Kontrol nuqtani o'rnatish uchun:

1. Protseduraning berilishini ochish kerak, xuddi yuqorida redaktor kodining oynasida tavsif etilganidek.

2. Kursorni kerakli satrga o'rnatish kerak.

3. Instrumentlar panelida qo'l rasmi tugmachani bosish kerak yoki Toogle Breakpoint Debug menyusidagi komandani tanlash kerak. Satr jigar rangdagi fonda ajratiladi va panelning chap tarafida, shu satrning karshisida belgi paydo bo'ladi- qalin jigar rang nuqta.

Eng sodda va tabiiy to'xtash nuqtasining o'rnatilishi: sichkon kul rang polosada bosiladi, muharrir oynasi chap tarafida. Nuqtani yo'qotish uchun sichqoncha tumasini kerakli satr ro'parasidagi jigar rang nuqtani bosish kerak.

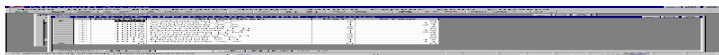
Endi protsedurani bajarayotganda, bajarish kerakli satrga yetganda to'xtash sodir bo'ladi, ekranda kod muharririning oynasi paydo bo'ladi, u yerda belgilangan satr sariq rangda ajratilgan, bu xozirgi paytdagi komanda. Local Window tugmasini bosib yoki kerakli komandani muharrirning kontekst menyusidan tanlang. Shunda ekranning past qismida yangi oyna Locals(lokal) paydo bo'ladi. Bu oynada hamma lokal o'zgaruvchilarni va xozirgi payt konstantalar protsedurasini, shuningdek hamma ob'ektlar formasining xossalarni ierarxik ro'yxatini ko'rish mumkin. Mutaxassisni aynan shu xossalar qiziqtiradi, chunki ko'pchilik xossalarda o'zgaruvchanlik mazmunini to'xtash nuqtasida kursorni muharrirning oynasida o'zgaruvchanlikka keltirib ko'rish mumkin.

Agar siz muolajani qadamma qadam bajarsangiz, u xolda xossalar ma'nosining o'zgarishini kuzatishingiz mumkin.

Call Stack oynasi

Call Stack oynasi(Chaqirishlar steki) o'z ichiga hamma bajarilmagan protseduralar ilovasining ro'yxatini oladi va qo'shilgan protseduralarning trassirovkasida foydalaniladi. Shunda eng avval chaqirilgan protseduraning birinchisi stekning pastki qismida

joylashadi, eng so`nggisi - yuqori qismida protseduraning nomi uch qismdan iborat : loyiha nomi, modul nomi(modul formasi shu jumladan), protsedura nomi;



Call Stack dialog oynasi

Stekda kerakli protsedurani tanlab, Show(ko`rsatish) tugmachasini bosish mumkin. Shunda Call Stack oynasi yopiladi va modul kodi oynasida kerakli protsedura paydo bo`ladi. Bunda kursor kod satriga o`rnatiladi, stekning keyingi protsedurasini chaqiradi

Agar xozirgi paytdagi protsedura tanlansa, unda kursor xozirgi paytdagi komandaga o`rnatiladi.

Call Stack oynasi bir necha usulda chaqiriladi:

Instrumentlar panelida Call Stack tugmasini bosish;

View, Call Stack komandasini tanlash;

<Ctrl>+<L> tugmalar kombinatsiyasini bosish;

Locals oynasi o`ng burchagida uch nuqtali tumani bosish.

Bu oyna faqat to`xtash rejimida ochilishi mumkin.

Hodisani qayta ishlash protsedurasini yaratish

Makroslar yoki makrokomandalardan emas Visual Basic komandalaridan iborat hodisani qayta ishlash protsedurasini yaratish uchun quyidagilar bajariladi:

Hodisa ushlab olinadigan ob'yekt xossalari oynasini ochish.

«События» qo`shimcha sahifasida kerakli hodisa qarshisidagi satrda «protsedura obrabotki» punktini tanlash.

Paydo bo`lgan satr yonidagi tahrirlash nuqta ustida sichqoncha tugmasini tez bosish. Visual Basic muharriri ochilib, unda qayta ishlash protsedurasi shabloni ko`rinadi.

Qayta ishlash kodini yozish.

Visual Basic Sintaksisi

Protseduralar va funksiyalar, ularning VBA chaqirilishi va parametrlarning uzatilishi

VBAda programmaning asosiy komponentlari protseduralar va funksiyalar hisoblanadi. Ular programma kodining qismi hisoblanadi, operatorlar Sub va EndSub Visual Basic yoki Function va EndFunction oralarida tuzilgan bo'ladi. VBA protsedurasi quyidagi ko'rinishda bo'lishi mumkin:

Sub<protsedura nomi>(<argument1>, <argument2>,...)

<operatorVisual Basic1><operatorVisual Basic2>EndSub

funksiyaning protseduradan farqi shundaki, uning nomi o'zgaruvchi sifatida chiqadi va funksiyaning qaytishi chaqirish nuqtasi ma'nosida foydalaniladi. Bu shunday ko'rinishda bo'lishi mumkin:

Function<funksiya nomi>(<argument1>, <argument2>,...)

<operatorVisual Basic1><operatorVisual Basic2>

<funksiya nomi>=<kaytariluvchiQiymat >

EndFunction

Quyida yozilgan protsedura yoki funksiyadan foydalanish uchun uni chaqirish lozim. Argumentlar ro'yxati bo'shmas protsedurani faqat boshqa protsedura yoki funksiyadan, unda uning nomidan va argumentlar xaqiqiy qiymatlari ro'yxatidan VBA operatorlaridan biri sifatida foydalanib chaqirish mumkin. Funksiyani esa faqatgina VBA operatorlari yordamidagina emas, balki uning nomini argumentlar xaqiqiy qiymatlari ro'yxati bilan birga to'g'ri formulaga yoki VBA programmasining ifodasi yoki masalan so'rov hisoblanuvchi maydonchasi to'g'ri formulasiga, forma va Access hisobotiga joylash mumkin. Argumentlar bo'sh ro'yxatli protsedura (komanda makrosi) faqatgina boshqa protseduradan yoki funksiyadan emas, balki tez chaqiriladigan klavishlar kombinatsiyasi, ochiladigan komandalar menyusi yoki instrumentlar paneli tugmachalari yordamida ham chaqirilishi mumkin. Yana bu protsedurani xar xil hodisalar bajarilishi bilan bog'lash mumkin. Masalan, forma ochilishi yoki hisobot, sichqonchaning tugmachasini formada elementlar boshqarish formasi, aksariyat boshqarish elementlari ActiveX. Bunday protseduralarni hodisalarni qayta ishlash protseduralari deyiladi. Argumentlar o'tkazilishiga muhtoj funksiya yoki proeduralarni bu yo'l bilan chaqirish mumkin emas.

Agar chaqirilayotgan protseduraning unikal nomi bo'lsa va shu modulda joylashgan bo'lsa, nima va chaqirilayotgan protsedura u xolda uning chaqirilishi uchun uning nomini ko'rsatish va asli ma'noli argumentlar ro'yxatini ko'rsatish kifoya, uni qavsga olmagan xolda protsedurani chaqirishni ikkinchi usuli operator Calldan foydalanishdir. Avval operator Call keladi so'ng protsedura nomi va parametrlar ro'yxati, bu xolda albatta qavslar ichida bo'lgan xolda. Funksiyani xuddi shu yo'l bilan chaqirish mumkin, xuddi protseduradagidek faqat ko'pinchalik boshqa ro'yxatlar parametri operatorning o'ng qismida o'zlashtiriladi.

mana protsedurani chaqirish masalasi CrossRC nomi ostida unga ikki argumentlar uzatilishi bilan(konstanta va ifoda);

CrossRC 7, i+2 yoki CallCrossRC(7, i+2)

Mana ikkinchi funksiya chaqirish masalasi Left va Mid, va ular qaytargan qiymatni ifodada foydalanilishi:yStr=Left(y, 1)&Mid(y, 2, 1)

O'zgaruvchini protseduraga yoki funksiyaga ikki xar xil uzatish yo'li ruxsat etiladi. Qiymat bo'yicha va ilova bo'yicha. Agar o'zgaruvchi ilova bo'yicha uzatilsa, u xolda bu protsedura yoki funksiyaga bu o'zgaruvchining adresi xotirasiga uzatiladi. Bunda protseduraning formal argumentlarning o'xshashligi sodir bo'ladi va unga asli parametrlar uzatiladi. Shu bilan chaqirilayotgan protsedura asli parametrlarning qiymatini o'zgartirishi mumkin: agar protsedurani formal argumenti o'zgarsa u xolda uzatilgan unga asli parametri chaqirilganda uning qiymatiga ta'sir qiladi. Agar asli parametr qiymatiga qarab uzatilsa, u xolda formal argumenti chaqirilgan protseduraning yoki funksiyaning faqat asli parametr ma'nosini oladi, lekin o'zgaruvchining o'zini emas, faqat shu

parametr sifatida oladi.

Bu bilan barcha o'zgarishlar formal argumentining qiymati o'zgaruvchi qiymatiga ta'sir qilmaydi, u asli parametrdir.

Protseduralar parametri yoki funksiyalar uzatilishi yo'llari uning tavsifnomasida uning argumentlari ko'rsatiladi: argument nomiga uning tavsifi joriy etish uzatish yo'li tavsiflovchi ByRet masalani yuborish bo'yicha uzatadi, ByVal esa ma'nosi bo'yicha. Agar parametr uzatish yo'li yo'q bo'lsa u xolda ko'zda tutilgan xolda uzatish bo'ladi.

Aytilganlarni quyidagi misolda tushuntiramiz.

Quyida ikki protsedura tavsifi berilgan:

```
Sub Main()
```

```
a=10
```

```
b=20
```

```
c=30
```

```
callExample(a, b, c)
```

```
call msg Box(a)
```

```
call msg Box(b)
```

```
call msg Box(c)
```

```
endSub
```

```
sub example(x, By val y, By RetZ)
```

```
x=x+1
```

```
y=y+1
```

```
z=z+1
```

```
call msg box(x)
```

```
call msg box(y)
```

```
call msg box(z)
```

```
endsub.
```

Yordamchi protsedura example foydalanadi formal argumentlar sifatida uchta o'zgaruvchi, xar xil tavsif etilgan.

Keyinchalik bu protsedura tanasida ularning xar biri birga oshadi, so`n gesa uning qiymati ekranga chiqadi msnbox funksiya yordamida. Asosiy protsedura Main o`zgaruvchilar a, b, c ning qiymati o`matadi, so`n gesa asli argumentlar sifatida example l protsedurasiga uzatadi. Bunda birinchi argument ilova bo`yicha uzatiladi, ikkinchisi qiymat bo`yicha, uchinchisi -yana ilova bo`yicha. Protседura example qaytgandan so`ngasosiy protsedurani ham ekranga uchta o`zgaruvchi argument sifatida uzatadi. Hammasi bo`lib ekranga oltita qiymat:

Avval bu 11, 21 va 31(hamma olingan qiymatlar birga oshgan va example protsedurasida chikariladi).

Keyin esa sonlar 11, 20 va 31 bu qiymatlar Main- protsedurasi orqali chiqariladi, aksariyat o`zgaruvchilar ilovada o`tkazilgan oshadi, o`zgaruvchi esa qiymati o`tkazilgan o`zgarmaydi.

Dastur ko`p protsedura va funksiyalardan iborat. Ular bitta yoki bir nechta modullarda joylashishi mumkin. Modullar loyihaga birlashgan va bir loyihada bir nechta xar xil dastur bo`lishi mumkin. Ular umumiy modul va protseduralardan foydalanishi mumkin.

Protседuralarning xar biri bir modulda joylashib unikal nomga ega bo`lishi kerak, lekin loyihada xar xil modul bo`lishi mumkin. Aslida unikal nomlar bir loyihada foydalanish tavsiya etiladi, lekin boshqacha ham mumkin.

Agar loyihada xar xil protsedura bir xil nom bilan bo`lsa, u xolda nomini aniqlashtirish uchun protsedura chaqirilganda quyidagi sintaksis qo`llanadi:

<Modul nomi><Protsedura nomi>

Agar bu xolda modul nomi bir necha so`zdan iborat bo`lsa, bu nomni kvadrat qavslarga olish kerak. Masalan, agar modul "Grafik protseduralar"-deb atalsa, protsedura-"Krestcha" deb atalsa, chaqiriq quyidagi shaklda bo`lishi mumkin:

[Grafik protseduralar].Krestcha

Boshqa loyihalarda joylashgan protseduralardan ham foydalanish mumkindir. Bu xolda nomni aniqlashtirish yana bir darajasi kerak bo`ladi

<Loyiha nomi><Modul nomi><Protsedura nomi>

O`zgaruvchilar, konstantalar va ma'lumotlar turlari

Boshqa dasturlash trlari kabi VBA tilida ham vaqtinchalik qiymatlarni saqlash, parametrlarni uzatish va hisoblashlar olib borish o`zgaruvchilardan foydalaniladi. VBA tilida o`zgaruvchilarni ta'riflash va ulardan foydalanish asosiy xususiyatlariga qisqacha tuxtalib utamiz. Odatda o`zgaruvchilarni ishlatishdan oldin, uni e'lon qilishadi, ya'ni shu Visual Basic va o`z dasturingizda o`zgaruvchilar nomlaridan foydalanishingiz xaqida oldindan xabar berasiz, bunda o`zgaruvchida saqlanishi lozim bo`lgan ma'lumot turi ham e'lon qilinadi. VBA tilida oddiy Basic kabi Dim operatoridan foydaliniladi. Bu operator sintaksisi :

Dim<o`zgaruvchi nomi> [As<ma'lumot turi>]

VBA tilida o`zgaruvchilarni nomlashning quyidagi qoidalari mavjud. O`zgaruvchi nomi 255 simvoldan uzun bo`lmasligi kerak, u xarfdan boshlanishi kerak. Bu xarfdan so`ng xarf raqam va ostiga chiqish simvollar kelishi mumkin. O`zgaruvchi nomi oxiriga quyidagi olti simvol ma'lumotlar tillari ta'riflaridan biri qo`shilishi mumkin.

! # % & @

Bu simvollar o`zgaruvchi nomining qismi emas. Agar dasturda string1\$ va string1 nomlardan foydalanilsa, ular bitta satrli o`zgaruvchiga ko`rsatadi. O`zgaruvchi nomini xar xil tipini aniqlovchi simvollar ishlatib bo`lmaydi yoki bir vaqtning o`zida ma'lumot turini e'lon qilib, shu e'longa mos kelmaydigan simvoldan foydalanish mumkin emas.

Ma'lumotlar turini muxokama qilganda bu masalaga biz yana qaytib kelamiz. O`zgaruvchilar nomlari sifatida VBA kalit so`zlari va standart ob'yektlar nomlaridan foydalanish mumkin emas. Shuning uchun o`zgaruvchilar nomlarini katta emas kichik

xarflardan boshlash lozimdir. VBA kalit soʻzlarida va standart ob'yektlari nomlari birinchi xarflari kiritishda avtomatik ravishda katta xarfga oʻzgartirilgani uchun, siz bilmasdan oʻzgaruvchi taqiqlangan nomidan foydalanish xavfidan xalos boʻlasiz. Oʻzgaruvchi nomlarida lotin alfavitdan tashqari kirilchadan foydalanish mumkin. Bu rus foydalanuvchilari uchun qulay boʻlib, oʻzgaruvchi nomlarini rus tilida berishga imkon beradi. Koʻpincha dasturlash tillarida, masalan Pascalda oʻzgaruvchilar albatta taʼriflangan boʻlishi shart va bu taʼriflar kompilyator tomonidan oʻzgaruvchilarga xotiradan joylashtirish uchun foydalaniladi. VBA tilida esa oʻzgaruvchilarni eʼlon qilish shart emasdir. Uning ajdodi Basic tili kabi eʼlon kilinmagan oʻzgaruvchilardan foydalanishga yoʻl qoʻyiladi.

Oʻzgaruvchilarga xotiradan joy ajratish dinamik ravishda amalga oshirilishi mumkin, oʻzgaruvchilarda saqlanayotgan maʼlumotlar turi oxirgi simvolga qarab aniqlanishi mumkin. Oʻz paytida oʻzgaruvchilarni taʼriflashdan voz kechish hisobiga yozuvlarini qisqartirish tarafdorlari va hamma umzgaruvchilarni majburiy ravishda taʼriflash tarafdorlari orasida uzoq bajarilib borilgan edi. Ikki tomonning dalillari ham jiddiydir- ikkinchidan xam, koʻp xollarda hamma oʻzgaruvchilarni taʼriflash talabi xar bir oʻzgaruvchi taʼrifini erinmasdan yozib chikishni talab etadi.

Masalan:

Dim i As Integer, j As Integer

Dim x As Double

Boshqa tomondan taʼriflanmagan oʻzgaruvchilardan foydalanishga imkon beruvchi dasturlash tillarida topilishi qiyin boʻlgan xatolar nomlarni yozishdagi xatolardir. Baʼzida bu xatolarni topishga ketadigan vaqt, hamma oʻzgaruvchilarni taʼriflashga ketadigan vaqtdan ancha koʻpdir.

VBA tilida bu muammoni daturchining oʻzi xal qiladi. Tilda **Option Explicit** operatori mavjuddir. Agarda modulingizni shu operatoridan boshlasangiz (bu operator modulda eng birinchi protseduradan oldin kelishi kerak)VBA shu modulda hamma oʻzgaruvchilarni taʼriflashni talab qiladi va xar gal taʼriflanmagan umzgaruvchiga duch kelganda kattalik xaqida maʼlumot beradi. Bundan tashqari agarda hamma modullaringiz uchun bu talab shart bumlishini xoxlasangiz VBA muharriri Options(parametrlar) dialog oynasining Editor(Muharrir) qoʻshimcha sahifasida Require Variable Declaration(oʻzgaruvchilarni oshkora taʼriflash) parametrini oʻrnatishingiz mumkin.

Bu parametr oʻrnatilgandan soʻng Visual Basic muharriri xar yangi yaratilayotgan modul boshiga Option Explicit operatorini avtomatik qoʻyadi. Lekin bu bayroqqa oldin yaratilgan modullarga taʼsir qilmaydi-ularga operatorni qoʻlda kiritishingizga toʻgʻri keladi.

«VBA da maʼlumotlar tiplari» Jadvali

Ma'lu-motlar tipi	Ta'rif
Array	O'zgaruvchilar massivi, massiv konkret elementiga ilova qilish uchun indeks ishlatiladi. Talab qilinadigan xotira: massiv tipiga bog'liq
Boolean	Ikki qiymatdan birini qabul qiladi: True yoki False. Talab qilinadigan xotira: 2 bayt
Byte	Ishorasiz son, 0 dan 255 gacha. Talab qilinadigan xotira: 1 bayt
Currency	Yaxlitlash xatoligini olish kerak bo'lganda pul hisoblarni o'nli nuqtadan so'ng fiksirlangan belgilar soni bilan olib borish uchun ishlatiladi. Qiymatlari diapazoni: -922 337 203 685 477, 5808 dan 922 337 203 685 477, 5807gacha. Talab qilinadigan xotira: 8 bayt. Tip ta'riflash ko'zda tutilgan simvoli: @
Date	Sanalarni saqlash uchun ishlatiladi. Qiymatlari diapazoni 1 yanvarya 0100 y. dan 31 dekabrya 9999 y. gacha. Talab qilinadigan xotira: 8 bayt
Double	Ikkilik aniqlikdagi so'zuvchi nuqtali sonlar. Qiymatlari diapazoni manfiy sonlar uchun: -1, 797693 13486232E308 dan -4, 94065645841 247E-324 gacha. Qiymatlari diapazoni musbat sonlar uchun: Qiymatlari diapazoni musbat sonlar uchun 4, 94065645841 247E-324 dan 1, 7976931 3486232E308 gacha. Talab qilinadigan xotira: 8 bayt. Tip ta'riflash ko'zda tutilgan simvoli: #
Integer	Qisqa butun sonlar. Qiymatlari diapazoni: -32 768 dan 32 767 gacha. Talab qilinadigan xotira: 2 bayta. Tip ta'riflash ko'zda tutilgan simvoli: %
Long	Uzun butun sonlar. Qiymatlari diapazoni: ot -2 147 483 648 dan 2 147 483 647 gacha. Talab qilinadigan xotira: 4 bayt. Tip ta'riflash ko'zda tutilgan simvoli: &
Object	Faqat ob'yektlarga ilovalar saqlash uchun ishlatiladi. Talab qilinadigan xotira: 4 bayt
Single	Oddiy aniqlikdagi so'zuvchi nuqtali son qiymatlar. Qiymatlari diapazoni manfiy sonlar uchun: -3. 402823E38 dan -1, 401 298E-45 gacha. Qiymatlari diapazoni musbat sonlar uchun: 1, 401 298E-45 dan 3. 402823E38gacha. Talab qilinadigan xotira: 4 bayta. Tip ta'riflash ko'zda tutilgan simvoli: !
String	Satrlri qiymatlarni saqlash uchun ishlatiladi. Satr uzunligi: 0 dan 64 Kbaytgacha. Talab qilinadigan xotira: 1 bayt na simvol. Tip ta'riflash ko'zda tutilgan simvoli: \$
Variant	Xaar xil tipdagi ma'lumotlarni saqlash uchun ishlatilishi mumkin: sana/vaqt, suzuvchi nuqtali sonlar, butunsonlar, satrlar, ob'yektlar. Talab qilinadigan xotira: 16 bayt, plyus 1 bayt xar bir satrli simvol uchun Tip ta'riflash ko'zda tutilgan simvoli: yo'q
Foydalanuvchi tomoni-dan ta'riflanadigan tip	Foydalanuvchi tomonidan ta'riflanadigan ma'lumotlar tip, vazifasi va ajratiladigan xotira hajmi ta'rifga bog'liq. Bu tipdagi o'zgaruvchida xar xil tipdagi xar xil qiymatlar to'plamini saqlashga imkon beradi

O'zgaruvchining tavsifida ma'lumotlar turining ko'rsatmasi tushirib qoldirilishi mumkin. O'zgaruvchining turi bunda o'zgaruvchi nomining oxirgi simvoli bilan belgilanadi:@, #, %, &, /, \$ (Currency, Double, Integer, Long, Single, String). Masalan, agar simvol \$ qator ma'lumotlari turini aniqlash simvoli bo'lsa, u xolda text\$ nomli o'zgaruvchi avtomatik xolda "simvollar qatori" "o'zgaruvchi turi" bo'lib qoladi

Keyinchalik bu maxsus simvol ma'lumotlar turining ko'rsatmasi tushirib qoldirilishi mumkin, lekin o'zgaruvchining nomida doimo turni aniqlash simvolining ishtirok etishi, qaysi ma'lumotlar turiga tegishli ekanligini eslatib turadi-bu esa bir biriga zid ma'lumotlar turidan foydalanish xato qilmaslikka yordam beradi.

Agar oxirgi simvol yuqorida ko'rsatilganlarning hech biriga kirmasa va tur ko'rsatmalaridan foydalanilmasa foydalanilmasa, u xolda o'zgaruvchilar ko'zda tutilgan ma'lumotlar turiVariantda belgilanadi, bu esa unda barcha ma'lumotlarni saqlashga imkon beradi.

Bir xil protsedurada bir biridan faqat maxsus simvoli bilan ajralib turadigan o'zgaruvchilar tomonidan foydalanish mumkin

emas o`zgaruvchan oxirida. Masalan o`zgaruvchining baravar bir vaqtda ishlatilishi mumkin emas var\$ va var%. Shuningdek simvol aniqlash turi bo`lgan o`zgaruvchining nom oxirida e`lon qilinishi mumkin emas, tavsiyachi yordamida As<tip peremennoy> (agar bu aniqlash oddiy simvol aniqlash turini qo`llashga qarshilik qilmasa) Masalan, agar siz xato xaqida ma'lumot olsangiz, quyidagi barcha aniqlovchilarni kirgizib:

Dim var 1% as string

Dim var2% as integer

Protsedura yoki funksiyaning argumentlar ma'lumoti tipini aniqlash uchun protsedura yoki funksiyaning sarlavhasi qatorida ma'lumotlar tipi tavsifi foydalaniladi.

Masalan, protseduraning keyingi sarlavha satri uning parametrlarini o`zgaruvchining satr tipida tavsiflaydi:

Sub splitstr(str1 as string, str2 as string, str3 as string)

Ma'lumotlar tipini aniqlash qaytayotgan funksiyaning qiymati funksiyaning satrini yakunlaydi, masalan:

Function Find split space(str1 as string) as integer

tavsiflaydi qaytayotgan funksiyaning qiymatini xuddi o`zgaruvchining kalta butun tipi.

Dastur tezroq ishlashi uchun va kam xotira egallashi uchun, mumkin bo`lganda o`zgaruvchining konkret tiplaridan foydalanish tavsiya etiladi. Universal tip Variant emas. o`zgaruvchining Variant tipini qayta ishlash uchun faqat qo`shimcha xotira emas, balki qo`shimcha vaqt kerak bo`ladi; qaysi konkret ma'lumotlar tipiga qarashli ekanligini bu o`zgaruvchining qayta ishlash vaqtida talab qilinadi, yana kerak bo`lsa ma'lumotlar tipini o`zgartirishni bajarish.

Nomlangan konstantalar tavsifi uchun operator Const qo`llaniladi o`zgaruvchi Dim tavsifi operatori bilan o`xshash. Bu operatorning sintaksisi:

Const<konstanta nomi>(as<ma'lumotlar tipi>)=<ifoda>

<ifoda>-istalgan qiymat yoki formula qaytayotgan qiymat konstanta sifatida foydalanishi kerak bo`lgan. Masalan keyingi operator butun konstanta tipini aniqlaydi:

const maxLen%=30

Xuddi o`zgaruvchidagi kabi konstantada ma'lumotlar tipining xar xil qiymati bor, lekin bunda ular o`z qiymatini dastur bajarishda o`zgartirmaydi.

Agar siz programmangizda ma'lum bir konstantadan foydalanmoqchi bo`lsangiz, u xolda bu konstantaga ma'noli nom berish tavsiya etiladi va uni modul boshida tavsif etish, keyinchalik faqat nomlangan konstantalardan foydalanish tavsiya etiladi. Bu dasturni tushunarligina qilib qolmay balki soddalashtiradi va sozlashda xamrox qiladi. Ko`pinchalik u yoki bu konstantaning qiymatini o`zgartirish talab qilinadi (xech bo`lmaganda sozlash davrida) va shunda faqatgina bitta qiymatni nomlangan konstantada o`zgartirish kifoya qiladi. Agarda dasturning matn kodida aynan shu qiymat foydalanilgan bo`lsa, unda u qiymatning hamma kelib chiqishini o`zgartirish ancha murakkab bo`ladi.

Massivlar

Massiv-bu o`zgaruvchi, unda bir vaqtda bir tipdagi bir necha qiymatlar saqlanadi. U o`zini bir tipdagi indeksli o`zgaruvchilar yig`indisi hisoblanadi. Massivning qo`llaniladigan indeks soni xar xil bo`lishi mumkin. Ko`pinchalik bir yoki ikki indeksli massivlar qo`llaniladi, ba'zan -uch indeksli, undan ko`p sonli indekslar kam uchraydi. VBA 60tagacha indekslar qo`llashi mumkin. Massivning indekslar soni massivning razmerini belgilaydi. Bir indeksli massiv bir o`lchovli massiv deyiladi. Ikki indeksli - ikki o`lchovli va hokazo.

Katta sonli massivlar katta xotirani egallaydi, shuning uchun ularni qo'llashda ehtiyot bo'lish kerak. Massiv qo'llashdan avval uni operator Dim yordamida e'lon qilish lozim va unda massivda qiymatlar qaysi tip da saqlanishini ko'rsatish kerak. Hamma qiymatlar massivda bir tipdagi ma'lumotlar turiga qarashli bo'lishi majburiy. Bu chegaralashni amaliyotda chetlab o'tish mumkin, buning uchun massiv e'lon qilinganda Variant tipidan foydalanish mumkin, bunda massiv elementlari xar xil tipdagi qiymatga ega bo'lishi mumkin.

Quyida massivni e'lon qilish operatorining sintaksisi:

Dim<massiv nomi>(<razmer1>, <razmer2>,...)As<ma'lumotlar tipi>

bunda qavsda ko'rsatilgan kattaliklar <razmer1>, <razmer2> va hokazolar massiv kattaligini - indekslar soni shu indeks uchun maksimal mumkin bo'lgan qiymatni beradi. Bunda massivning elementlar indeksatsiyasi o'rnatilgan bo'yicha noldan boshlanadi. Shunda **Dim Arrayl(9) as integer** ni e'lon qilinishi butun tipdagi o'zgaruvchi bo'lgan 10 elementdan iborat bir o'lchovli massivni aniqlaydi.

Dim Array2(4, 9) as Variantning e'lon qilinishi esa Variant universal tipidagi o'zgaruvchi bo'lib (5*10) ellik elementdan iborat ikki o'lchovli massivni aniqlaydi. Pastki chegaralar standart qiymati sifatida mumkin bo'lgan qiymatlar indeksi uchun faqatgina nol qo'llanishi mumkin. Bu standart qiymatni operator Option Bass yordamida o'zgartirish mumkin. Agar modul boshiga Option Bass 1 operatorni joylashtirsak, u xolda massiv elementlari indeksatsiyasi o'rnatilgan bo'yicha noldan emas, birdan boshlanadi.

Massiv e'lon qilinganda faqatgina indeksning yuqori chegarasini emas, balki uning pastki chegarasini ham ko'rsatish mumkin, ya'ni massivning konkret indeksining diapazon o'zgarishi beriladi, bunda chegara ixtiyoriy butun son bo'lishi mumkin, nomanfiy bo'lishi shart emas.

Mana bu aniqlashning sintaksisi:

Dim<massiv nomi>(<min1>to<maks1>,...) As <ma'lumotlar tipi>

Masalan, agar siz meteorologik ma'lumotlar massivi bilan ishlamoqchi bo'lsangiz, oxirgi ikki hafta ichidagi o'rtacha kunduzgi xaroradni ko'rsatuvchi, u xolda massivning quyidagi aniqlanishi qulay bo'ladi.

Dim Temperature(-14 to 0) As Single

Bunda, masalan temperature(-2) o'tgan kungi xaroratga to'g'ri keladi, kerakli indeksni aniqlash uchun esa sizni qiziqtirayotgan kun uchun kerakli sana farqidan foydalanish kifoya.

Yuqorida ko'rilgan misollarda gap massivning mustaxkam o'lchovlari xaqida borgan edi, unda elementlar soni massivning operator Dim tavsifida aniq ko'rsatilgan. Bunday massivlar statik massiv deyiladi.

VBA da dinamik massivlar ham qo'llanilishi ruxsat etiladi, bunda ularning o'lchamlari tavsifi qayd etilmaydi. Dinamik massiv o'lchamlari muayyan dastur bajarilayotganda o'rnatishim mumkin.

Dinamik massiv aniqlanayotganda operator Dim da massiv nomidan so'ngbo'sh qavslar turadi va o'zgaruvchan tipi tavsifi keladi. Indekslar soni va ular o'zgarishining diapazoni berilmaydi. Lekin massiv foydalanilishidan oldin, operator ReDim bajarilishi kerak, u dinamik massiv indeksining o'lchamlari va diapazon o'zgarishini beradi.

Dinamik massivning e'lon va o'lchamlari aniqlashining sintaksisi:

Dim<massiv nomi>() As <ma'lumotlar tipi >

ReDim<massiv nomi>(<razmer1>, <razmer2>,...)

dinamik massivda e'loni, o'lchamlarini aniqlash va foydalanish, o'lchamlari va kattaligi o'zgarishi quyidagicha bo'lishi mumkin:

Dim dArray()As Variant

ReDim dArray(1, 2)

dArray(0, 0)=2

dArray(0, 1)=3

k=dArray(0, 0)+dArray(0, 1)

ReDim dArray(k)

dArray(0)="Stroka1"

Bu masalada massiv dArray boshida xuddi ikki o'ldamli olti elementdan iborat massivdek aniqlanadi so'ngboshqatdan xuddi bir o'ldamli massivdek aniqlanadi, bunda yuqori chegara indeksi o'zgaruvchi k ning qiymatida beriladi.

Massivning shu paytdagi yuqori va pastki chegaralarini aniqlash uchun **LBound** va **UBound** funksiyalaridan foydalanish mumkin albatta.

Ko'zda tutilgan xolda massivning o'ldamlari o'zgarganda unga yangidan xotira ajratiladi va uning elementlarining qiymatlari yo'qoladi.

Massivning joriy qiymatini yo'qotmaslik uchun uning o'ldovlari o'zgartirilganda Preserve so'zi ishlatiladi. Masalan massivning dArray o'ldamini bir elementiga oshirish uchun bor elementlarning qiymatini yo'qotmagan xolda, quyidagidek bajarish mumkin.

ReDim Preserve dArray(UBound(dArray)+1)

O'zgaruvchining va konstantaning harakat sohasi

Hamma protseduralar, funksiyalar, O'zgaruvchilar va konstantalar VBAda o'z harakat sohasiga ega.

Bu ularning faqat dastur kodining ma'lum bir joyida - ayni ular tavsifida foydalanishi mumkinligini bildiradi.

Masalan, agar o'zgaruvchi A operator Dim yordamida tavsif etilgan bo'lsa, protsedura tanasida Proc1 nomi bilan aynan shu protsedura uning harakat sohasi hisoblanadi.

Shunday qilib agar boshqa protsedura Proc2 bo'lsa, u xolda siz bunda shu o'zgaruvchidan foydalana olmaysiz. Agar siz shunga harakat qilib ko'rsangiz u xolda xato xaqida ma'lumot olasiz tavsif qilinmagan o'zgaruvchidan foydalanish uchun, (agar operator Option Explictdan foydalanilsa) yoki boshqa o'zgaruvchini olasiz - xuddi shu nom bilan, lekin bir xil nomli o'zgaruvchi, birinchi protsedura bilan hech qanday bog'liqligi yo'q.

Dasturning qaysi joyida va aynan qanday tavsif etilgan o'zgaruvchi, buni uning harakat sohasi aniqlaydi va u qancha davr xotirada va o'zining qiymatini saqlaydi. O'zgaruvchining harakat sohasini aniqlashda uch xil darajaga ega:

1. Protseduraning darajasi

2. Modul darajasi

3. Loyiha darajasi

Protsedura darajasida o`zgaruvchini aniqlash uchun uning tavsifi shu protsedura tanasiga joylanadi, shunda shunda bu lokal o`zgaruvchi bo`ladi.

Protsedura darajasida o`zgaruvchini aniqlash va uni shu bilan birga tushunish uchun oson qilib, birgalikda modulning hamma protseduralarida ishlatilishi uchun uning tavsifini modulning e'lonlar bo`limida -biror bir protsedura yoki funksiya matni oldidan. Bunda yaqqol harakat sohasining yaqqol tavsifidan ham foydalanish mumkin :Dim kalit so`zi **Private** kalit so`zi bilan almashtiriladi. Qaysi tavsifdan foydalanishning farqi yo`q va nixoyat o`zgaruvchining loyiha darajasida tavsif qilish uchun uning tavsifini modul loyihaining biron-bir e'lonlar bo`limida joylashtirish kerak va albatta kalit so`zi **Public** foydalanilishi shart. Shunday qilib tavsif etilgan o`zgaruvchilar loyihaning ixtiyoriy modulida ishlatilishi mumkin.

Yuqorida aytilganlarning hammasi harakat sohasining konstant va massivlarining tavsifiga va aniqlanishiga kiradi.

O`zgaruvchilar uchun ular tavsifining yana bir usuli bor, ular darajasini o`zgartirmaydigan, lekin o`zgaruvchi qiymatini saqlab qoladigan protsedura darajasida tavsif qilingan protseduraning ishi tugagandan keyin. Buning uun tavsifchi **Static**dan foydalanishi lozim, bu bilan statik o`zgaruvchiligini aniqlab. Bundan o`zgaruvchi o`zi uchun ajratilgan joyni xotirada saqlab qoladi va o`zining qiymatini o`zi tavsif qilingan va foydalangan protsedura tamom bo`lgandan so`ng xam. Shunga qaramay statik o`zgaruvchi boshqa protseduralarda foydalanilmaydi. Uning faqatgina xayot davri o`zgaradi, harakat sohasi emas. (Agar shu protsedura qayta chaqirig`i bo`lsa)

Agar statik o`zgaruvchi tavsif etilgan protsedura qayta chaqirilsa, unda bu o`zgaruvchi o`zining oldingi qiymatini saqlab qoladi, oldingi chaqiriqda, ish yakunida qay xolda bo`lsa. Oddiy, statik bo`lmagan o`zgaruvchi xar gal o`zgaruvchini tashabbusga chorlaydi va protseduraga kirishda bo`sh qiymat oladi.

Protsedura va funksiyaning harakat sohasi

Protsedura va funksiyalar faqat ikita darajali harakat sohasiga ega:Modul darajasi va loyiha darajasi.

Loyiha darajasi yashirinch a foydalaniladi. Shunday qilib protsedura yoki funksiya ixtiyoriy boshqa protsedura yoki funksiya orqali chaqirilishi mumkin bu loyihada.

Protsedura va funksiya tavifida loyiha darajasida nomajburiy kalit so`zi **Public**dan foydalanish mumkin. Bu so`zning bor yoki yo`qligi ta'sir ko`rsatmaydi.

Agar faqat modul darajasida protsedura tavsif etilishi kerak bo`lsa buning uchun kalit so`zi **Private** qo`llaniladi. E'tiborga oling, bunday tavsif faqatgina protseduraning harakat sohasini toraytiribgina qolmay, lekin va uning mustaqil protsedura sifatida foydalanishni taqiqlaydi-uni faqat boshqa protseduradan chaqirish mumkin. Nixoyat, protsedura yoki funksiyaning tavsifida kalit so`zi **Static** dan foydalanish mumkin. Bu hech protsedura harakat sohasiga ta'sir qilmaydi, lekin hamma o`zgaruvchiga ta'sir qiladi, shu protsedura yoki funksiyaning ichida tavsif qilingan.

Bu xolatda hamma lokal o`zgaruvchilar Static statusini oladi va shu bilan protsedura tugagandan so`nguning xotirasida qoladi va chaqirilganda avvalgi qiymatini saqlab qoladi.

Quyidagicha boshlanadigan modul misolini ko`rib chiqamiz:

Public Al As String Private A2 As Integer

Dim A3 As Single

Sub Prod ()

Dim A4 As Integer

Static A5 As Integer

A1 = "Tekstovaya stroka 1"

A2= 2

A3 = 3.14

A4 = A4 + 4

A5 = A5 + 5

MsgBox A4

MsgBox A5

End Sub

Sub Proc2 () Proc1

MsgBox A1

MsgBox A2

MsgBox A3

MsgBox A4

MsgBox A5

Proc1 End Sub

Bu misolda o`zgaruvchi A1 butun loyiha darajastda aniqlangan(kalit so`zi Public foydalanilgan) o`zgaruvchilar A1 va A3 modul darajasida aniqlangan,, o`zgaruvchi A4 faqat protsedura Proc1, darajasida, o`zgaruvchi A5 protsedura tanasida aniqlangan Proc1, lekin statik o`zgaruvchidek tavsif qilingan.

Protsedura Proc2 chaqirilganda quyidagi sodir bo`ladi: bu protseduradan o`z navbatida protsedura Proc1 chaqiriladi, u hamma 5 o`zgaruvchining kuymatini tayinlaydi A1, A2, A3, A4, A5 so`ng o`zgaruvchi A4 va A5 -ning joriy qiymatini ko`rsatadi dialog oynasida.

Bu protseduraning yakunidan so`ng o`zgaruvchining joriy qiymati A1-A5 protsedura Proc2 dan chiqariladi.

Bunda o`zgaruvchilar A1-A3 o`z qiymatlarini saqlab qoladi, chunki ular modul darajasida tavsif etilgan o`zgaruvchilar A4 va A5 esa bo`sh qiymat qabul qiladi, chunki bu o`zgaruvchilarning harakat sohasi protseduradir, ular u yerda foydalaniladi. Protseduralar birining ichidagi bu o`zgaruvchilarning o`zgarishlari shu o`zgaruvchilarga mos boshqa protseduralardagi o`zgaruvchilarga bog`liq emas-aslini olganda bular xar xil o`zgaruvchilardir, faqatgina ularning nomlari bir xil. so`ngProc1 protsedurasini yana bir chaqirilish codir bo`ladi va u yana A4 A5 qiymatini o`zgartirib ekranga chiqaradi.

Bunda o`zgaruvchi A4 yana 4 qiymatini oladi, chunki protseduraning yangi chaqirig`ida bu o`zgaruvchi uchun boshqatdan xotira ajratiladi va u initsiyalizatsiyalanadi bo`sh qiymatda.

A4 dan farqliroq o`zgaruvchi A5-statik o`zgaruvchidek tavsif etilgan, o`zining oldingi qiymatini saqlaydi. Protseduraning oldingi chaqirig`idan, buning oqibatida uning qiymati qayta chaqiriqda 10ga teng bo`ladi.

Tarmoqlanish

Tarmoqlanishning boshqaruv konstruksiyasi ba'zi bir shartlarni tekshirishga imkon beradi va bu tekshirish natijalariga qarab u yoki bu grupp operatorlarini bajarishi mumkin. VBA da tarmoqlanishning tashkillashtirish uchun tarmoqlash va operatorlarining xar xil formalaridan foydalaniladi **If** va tanlash operatori **Select Case**.

U operatorining eng oson formasi bitta shartni tekshirish uchun ishlatiladi, so`ng yoki bajarilish yoki bitta ta operatorni o`tkazish yoki bir necha operatorlar uchun bloki ishlatiladi. If tarmoqlanish operatori kalta formasi bir satrga va blok formasiga ega. If kalta formasi bir satrga quyidagicha yozilishi mumkin:

```
if<shart>then<operator>
```

Blok formada kalta tarmoqlanish quyidagicha bo`ladi

```
if< shart >then
```

```
<operator1>
```

```
<operator2>
```

```
endif
```

Shart sifatida mantiqiy ifoddan foydalanish mumkin, qiymat qaytaruvchi True yoki False yoki ixtiyoriy arifmetik ifodadan foydalanish mumkin. Agar arifmutik ifoda qo`llanilsa, bu ifodaning 0 qiymati mantiqiy False qiymatga ekvivalent, ixtiyoriy 0 ga teng bo`lmagan qiymat True ekvivalent. Shart False qiymatini qaytarsa, u xolda operator yoki operatorlar bloki kalit so`zlari then va endif oralig`ida joylashgan va tarmoqlanish kalta operatorning tanasini tashkil qilsa bajarilmaydi.

Tarmoqlanishning kalta operatori yozuВида bir satrda kalit so`zlari endif foydalanilmaydi.

If operatorining to`liq formasi shunday hodisalarda foydalaniladiki, unda ikki xar xil operatorlar bloki va tekshirishlar natijasidagi shartlarda ulardan birinchi bajarish kerak. Ifning bunday formasi bir satrga yozilishi mumkin emas va xar doim yozuvning blok formasiga ega.

```
if < shart >then
```

```
<operatorlarblogi1>
```

else

<operatorlarblogi2>

endif

Agar shart rost bo'lsa operatorning birinchi bloki bajariladi, kalit so'zlari then va else oralig'ida joylashgan aksariyat xolda - ikkinchi blok kalit so'zlari else va endif oralig'ida joylashgan.

Protsedurangizning matni tushunarli va qulay bo'lishi uchun operatorlar gruppasiga chekinma tavsiya etiladi, xuddi ularning intaksisi tavsifida ko'rsatilganidek.

VBAda chekinmalarni o'zgartirish qulay vositasi mavjud-<Tab>klavishasi bosilishi chekinmani o'ngga oshiradi, <Shift>+<Tab> klavishlar kombinatsiyasini bosish bu chekinmani kamaytiradi. Ba'zida bir necha shartlarni tekshirish asosida bitta operatorlar blokini bloklar guruhidan tanlab olishga to'g'ri keladi. Buning uchun if..... Then..... Elseif tarmoqlanish operatorlari zanjiridan foydalanish mumkin.

If <shart1> Then

<Operatorlarblogi!>

Elseif < shart 2> Then

< Operatorlarblogi 2>

Elseif < shart 3> Then

< Operatorlarblogi Z>

Elseif < shart N> Then

< Operatorlarblogi N> Else

< Operatorlarblogi _Else>

End If

Bunday if..... then..... elseif operatorlar zanjiri hamma muammolarni xal qilishga imkon beradi, lekin bir necha imkoniyatlardan birinchi tanlash xar vaqt bitta ifoda xar xil qiymatlariga asoslangan bo'lsa, buning uchun maxsus mo'ljallangan Select case tanlash operatoridan foydalanish qulaydir. Bu operator quyidagicha sintaksisga ega.

Select Case <tekshirilayotganIfoda>

Case <Qiymatlarro'yxati1>

<Operatorlarblogi1>

Case < Qiymatlarro`yxati2>

< Operatorlarblogi2>

Case < Qiymatlarro`yxatiZ>

< OperatorlarblogiZ>

Case Else

< OperatorlarblogiElse>

End Select

Tekshirilayotgan ifoda Select Case operatori ishi boshlanishida hisoblanadi. Bu ifoda ixtiyoriy tipdagi qiymat qaytarishi mumkin, masalan mantiqiy, sonli yoki satrli. Ifodalar ro`yxati, vergul bilan ajratilgan bir yoki bir necha ifodalardan iborat. Operator bajarilganda shu ro`yxat elementlaridan biri tekshirilayotgan ifodaga tengligi tekshiriladi.

Ifodalar ro`yxati elementlari quyidagi shakllardan biriga ega bo`lishi mumkin:

<ifoda>bu nolda tekshirilayotgan ifoda qiymatiiga tengligi tekshiriladi.

<ifoda1>To<ifoda2 >bu xolda tekshirilayotgan ifoda qiymati ko`rsatilgan qiymatlar diapazoniga tegishligigi tekshiriladi.

Is<mantiqiy operator><ifoda> uchinchi xolda tekshirilayotgan ifoda ko`rsatilgan qiymat bilan berilgan mantiqiy operator yordamida tekshiriladi, masalan i1>=10 bajarilgan hisoblanadi, agar tekshirilayotgan ifoda 10dan kichik bo`lmasa.

Agar ro`yxat elementlaridan biri tekshirilayotgan ifodaga tug`ri kelsa, mos operatorlar guruhi bajariladi va shu bilan Select case tugallanadi, ifodalar ro`yxatining qolgan elementlari tekshiriladi, ya'ni birinchi to`g`ri keladigan element o`matiladi.

Agar biror element tekshirilayotgan ifodaga to`g`ri kelmasa, operatorlari guruhi bajariladi, agar ular mavjud bo`lsa.

Sikllar

VBA da sikllarni tashkil qilish turli usullari mavjud bo`lib ularni 2 guruhga ajratish mumkin Do-Loop shartli sikllar va sanovchi sikllar Tor-Next Do-Loop turidagi sikllar sikl tanasini tashkil qiluvchi operatorlar bloki necha marta qaytarilishi oldindan ma'lum bo`lmaganda ishlatiladi.

Bu sikl ma'lum shart bajarilmaguncha o`z ishini davom ettiradi.

Do-Loop operatorlarining shart turi va tekshirish vaqti bilan farq qiluvchi 4ta ko`rinishi mavjud.

Do sikl operatorlari sintaksisi

Konstruksiya	Ta'rif
<i>Do While <shart> <Operatorlar blogi> Loop</i>	Sikl tanasini tashkil qiluvchi operatorlar guruhi bajarilishidan oldin shart tekshiriladi. Sikl tanasini tashkil qiluvchi operatorlar guruhi bajarilguncha shart tekshiriladi. Agar shart bajarilsa (ya'ni True qiymatga ga bo'lsa sikl bajariladi), ya'ni bu konstruksiyada sikl ishini davom etish sharti ko'rsatiladi
<i>Do Until < shart > <Operatorlar blogi > Loop</i>	Sikl tanasini tashkil qiluvchi operatorlar guruhi bajarilishidan oldin shart tekshiriladi. Agar shart bajarilsa (ya'ni True qiymatga ga bo'lsa) sikl bajarilmaydi, ya'ni bu konstruksiyada sikl ishini to'xtatish sharti ko'rsatiladi
<i>Do < Operatorlar blogi > Loop Until < shart ></i>	Sikl tanasini tashkil qiluvchi operatorlar guruhi juda bo'lmasa bir marta bajarilgandan so'ng shart tekshiriladi. Agar shart bajarilsa (ya'ni True qiymatga ga bo'lsa) sikl bajarilmaydi, ya'ni bu konstruksiyada sikl ishini to'xtatish sharti ko'rsatiladi
<i>Do < Operatorlar blogi > Loop While < shart ></i>	Sikl tanasini tashkil qiluvchi operatorlar guruhi juda bo'lmasa bir marta bajarilgandan so'ng shart tekshiriladi. Agar shart bajarilsa (ya'ni True qiymatga ga bo'lsa) sikl bajariladi, ya'ni bu konstruksiyada sikl ishini davom etish sharti ko'rsatiladi

Sanovchi sikl operatori For..... Next 2 shaklli mavjuddir. Ko'p xollarda massivlarni qayta ishlashda, xamda biror operatorlar guruhini berilgan marta bajarish lozim bo'lsa hisoblovchili sikl For..... Next qo'llaniladi. Bu siklda Do... Loop siklidan farqli maxsus hisoblovchi o'zgaruvchi ishlatilib, sikl tanasi xar gal bajarilganda bu o'zgaruvchi qiymati ma'lum oshadi yoki ko'payadi.

Bu o'zgaruvchi berilgan qiymatga teng bo'lganda sikl bajarilishi to'xtaydi.

Bu turdagi sikl sintaksisi quyidagi ko'rinishga ega (kvadrat qavslar ichida sintaksis konstruksiya shart bo'lmagan elementlari):

For < sanovchi> = <boshlang'ichQiymat>

To <oxirgiQiymat>

[Step <oshirma>]

<Operatorlarblogi> Next [<sanovchi>]

Keltirilgan ta'rifga tushuntirgilar.

<oshirma>-musbat yoki manfiy son bo'lishi mumkin. Agar manfiy qiymat ishlatilsa, oxirgi qiymat boshlang'ich qiymatdan kichik yoki teng bo'lishi kerak, sikl tanasi juda bo'lmasa 1 marta bajarilishi uchun.

For..... Next sikli turgandan so'ng sanovchi chifatida ishlatilgan o'zgaruvchi agar qo'shilma musbat bo'lsa albatta oxirgi qiymatdan katta qiymat oladi va qo'shimcha manfiy bo'lsa oxirgi kichik qiymat qabul qiladi.

Agar boshlang'ich va oxirgi qiymatlar teng bo'lsa sikl tanasi faqat bir marta bajariladi.

VBAda ko'p ishlatiluvchi For..... Next siklining yana bir turi bir jinsli ob'yektlar massivi yoki oilasini tashkil qiluvchi

ob'yektlarni qayta ishlashda ko'p ishlatiladi. Siklning bu turida sanovchi bo'lmaydi sikl tanasi ob'yektlar massivi yoki oilasi xar bir elementi uchun qo'llanadi.

Bu sikl sintaksisi :

For Each <element> In <jamlanma>

<Operatorlarblogi>

Next [<element>]

bu yerda :

<element>-ob'yektlar oilasiga ilova sifatida ishlatiluvchi o'zgaruvchi;

<jamlanma>-massiv yoki oila nomi;

Shunday sikldan foydalanishga misol keltiramiz. Quyidagi protsedura joriy ochiq ma'lumotlar bazasi hamma jadvallari hamma maydonlari ro'yxatini bosmaga chiqaradi:

```
Public Sub EnumerateAllFields()
```

```
Dim MyBase As Database
```

```
Dim tdf As TableDef, fid As Field
```

```
Set MyBase = CurrentDb()
```

```
For Each tdf In MyBase. TableDefs
```

```
Debug. Print "Таблица: " & tdf. Name
```

```
For Each fid In tdf. Fields
```

```
Debug. Print " Pole: " & fid. Name Next fid
```

```
Next tdf
```

```
Set MyBase = Nothing
```

```
End Sub
```

Dim operatorlarida My Base o'zgaruvchini "DAO ma'lumotlar bazasi" sifatida e'lon qilgan edik tdf va fid bo'lsa -jadval va jadval maydoni sifatida. Set operatori My Base o'zgaruvchisiga joriy ochiq ma'lumotlar bazasini tayinlaydi. Xar bir jadval ta'rifi uchun jadval nomi bosmaga chiqariladi, shundan so'ng, xuddi shu turdagi ichki joylashgan sikl hamma maydonlari nomlarini bosmaga chiqaradi.

Shunga o'xshash For Each... Next sikl operatorini qayta ishlashga qo'llashni ko'rib chiqamiz. Bizda 1000 elementdan iborat(o'Ichovlari 10*10*10) sonli massiv mavjud bo'lsin va biz bu massivni 0 dan 1 gacha tasodifiy sonlar bilan

to'ldirmoqchimiz. Agarda biz oddiy hisoblovchili For... Next sikllarini qo'llasak, bu masalani yechish uchun 3 ta ichki joylashgan For... Next sikli yozishimiz kerak bo'lar edi:

```
Dim tArray{9, 9, 9} As Single
```

```
Dim i%, j%, k%
```

```
Randomize
```

```
For i=0 To 9
```

```
For j=0 To 9
```

```
For k=0 To 9
```

```
tArray(i, j, k) = Rnd()
```

```
Next k
```

```
Next j
```

```
Next i
```

Aslida bitta sikl yetarli agar hisoblovchili sikl o'rniga For Each... Next siklidan foydalanilsa.

```
Dim tArray(9, 9, 9) As Single
```

```
Dim elem As Variant
```

```
Randomize
```

```
For Each elem In tArray
```

```
elem = Rnd()
```

```
Next
```

Yana bir marta tarmoqlanish operatorlar yozuВидagi kabi, sikllar yozuВида chekinmalardan foydalanishni maslaxat beramiz.

So`rovlar yordamida ma'lumotlarni tahlil qilish

Kesishgan so`rovlar(Перекрестные запросы)

Kesishgan so`rovlar bu ma'lumotlar statistik qayta ishlab olib boriluvchi so`rovlar bo`lib ularning natijasi Excel natijaviy jadvaliga o`xshaydi. Kesishgan so`rovlar quyidagi ustunliklarga ega:

ko`p hajmdagi ma'lumotlarni qayta ishlash va ularni avtomatik ravishda grafiklar va diagrammalar yaratishga moslashgan formatda chiqarish

soddaligi va murakkab bir necha detalizatsiya darajalariga ega bo`lgan so`rovlarni tez qayta ishlash

Lekin ular kamchiliklarga ega masalan natijalar jadvalini ustunlardagi qiymatlar bo`yicha tartiblash

mumkin emas, chunki bir vaqtning o`zida ustunlardagi qiymatlarni hamma satrlar bo`yicha tartiblash mumkin emas. Lekin kamayish yoki o`sih bo`yicha tartiblashni satrlar sarlavhalari uchun berish mumkin.

Kesishgan so`rovlar ma'lumotlarni jadval sharlida tasvirlasyga qulay, lekin ularda ustunlar soni xar xil shuning uchun ular yordamida hisobotlar yaratih murakkab.

Kesishgan so`rov yaratish

Misol tariqasida "Borey" ma'lumotlar bazasi uchun kesishgan so`rov yaratamiz xar oylik mahsulot sotuvini chiqarish uchun. Ma'lum davrdan so`ng masalan xar oy yoki kvartaldan so`ngnatijalardan iborat jadvallar grafiklar yaratish uchun ishlatiladi. Quyidagi kadamma kadam kesishgan so`rov yaratish jarayonida so`rovlarda yangi hisoblanuvchi maydonlar yaratish uchun ifodalardan foydalanish ko`rsatilgan. Shunday qilib satrlarda mahsulotlar ularga mos ustunlarda oylik sotuv hajmlari chikariluvchi kesishgan so`rov yaratamiz.

Buning uchun:

Конструктор rejimida yangi so`rov yarating va unga **"Товары"** (Products), **"Заказы"** (Orders) и **"Заказано"** (Order Details) jadvallarini qo`shing.

"Товары" jadvalining **"КодТовара"** (ProductID) и **"Марка"** (ProductName) maydonlarini so`ngra **"Товары"** jadvali **"ДатаРазмещения"** (OrderedDate) maydonini so`rov blanki birinchi uchta ustuniga

ko`chiring.

Запрос, Перекрестный (Query, Crosstab Query) menyuu komandasini tanlang. Заголовок окна запроса Запрос1: на выборку (Query1: Select Query) so`rovning oynasi sarlavhasi Запрос1: перекрестный запрос (Query1: Crosstab Query) ga o`zgaradi. Bundan tashqari so`rovlar blankiga Перекрестная Таблица (Crosstab) va Групповая операция (Total) satrlari qo`shiladi va hamma ustunlarga avtomatik Группировка (Group By) operatsiyasi kiritiladi.

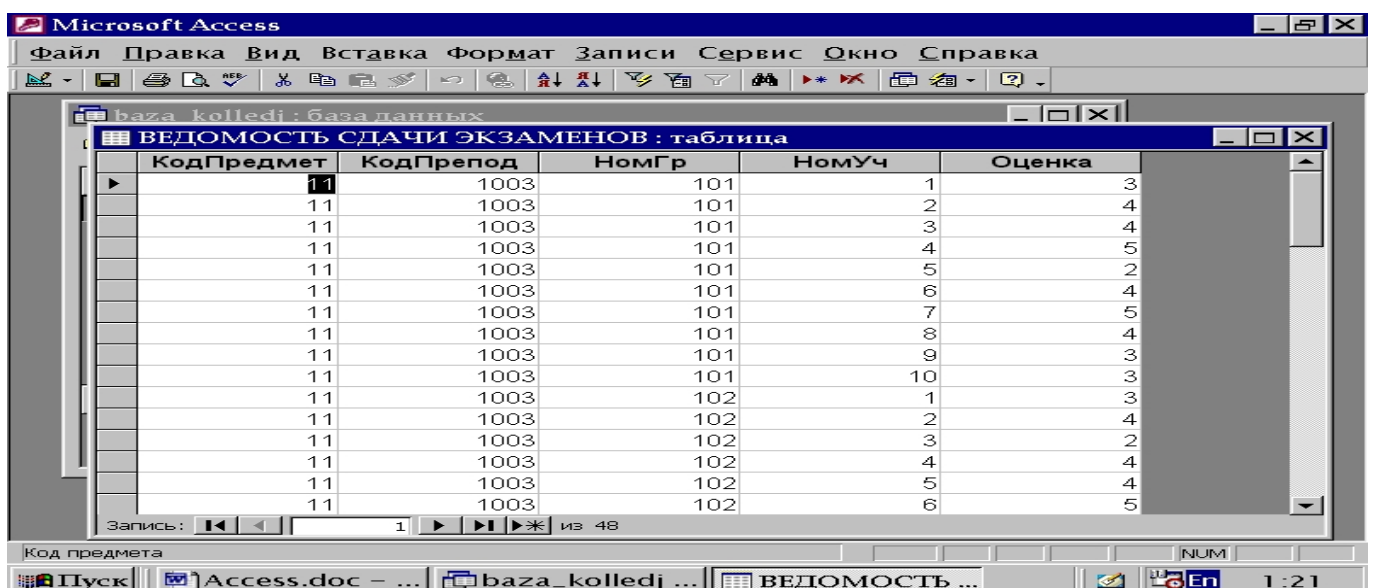
"КодТовара" ustuni Перекрестная Таблица yacheykasi ro`uxatida Заголовки строк (Row Heading) qiymatini tanlang. Xuddi shuni "Марка" ustuni uchun bajaring. Bu ustunlar kesishgan jadvali satrlarining kerakli sarlavhalaridir.

"ДатаРазмещения" ustuni Групповая операция yacheyka ro`uxatida Условие (Where) qiymat tanlang. Shu ustunning Условие отбора (Criteria) yacheykasida kesishgan jadvalda 2005 yil uchun ma'lumotlarni chiqarish uchun <=#31. 12. 05# And >=#01. 01. 05# ifoda kiriting.

Kursorni keyingi bo`sh ustunning Pole (Field) yacheykasiga o`mating va quyidagi ifodani kiriting: **Объем продаж: Sum([Количество]*[Заказано]. [Цена]).** Bu qiymati ifoda yordamida hisoblanadigan, nomi bo`lsa - "Объем продаж" maydon bo`ladi. Ifodada so`rovga kiritilgan jadvallardan maydonlar ishlatiladi, lekin maydonlar o`zi so`rov natijasiga kiritilmagan. Bu shuni bildiradiki hisoblanuvchi maydon yaratilganda ifodaga faqat so`rov maydoniga ilova emas, so`rov natijasiga kirmaydigan maydonlarga ham ilovalar kiritish mumkin. Muhimi ular berilgan jadvallarda bo`lishi. Ifodada "Цена" maydoniga ilovada "Заказано" jadval nomini ko`rsatdik, "Количество" maydoniga ilovada ko`rsatmadik. Jadval nomini shuning uchun ko`rsatdikki "Цена" nomli maydon "Товары" jadvalda ham, "Заказано" jadvalida ham mavjud.

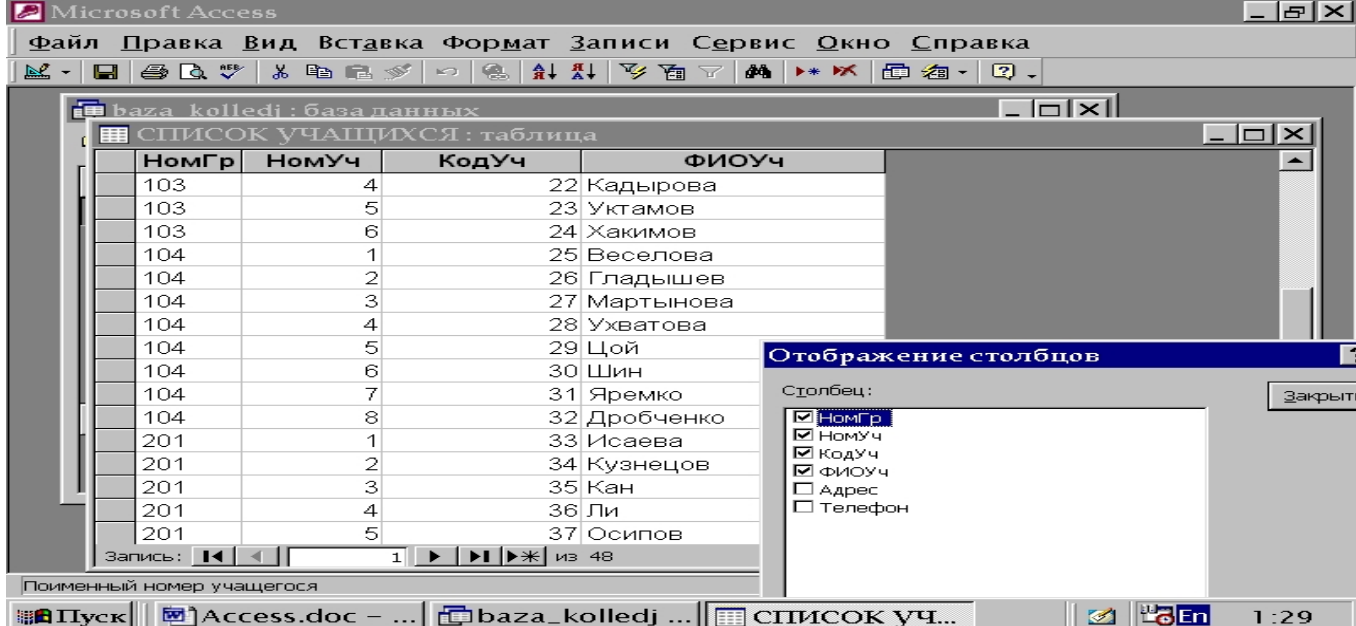
Shu ustunning Групповая операция (Total) yacheykasida Выражение (Expression) qiymati kiriting, so`ng Перекрестная таблица yacheykasida- Значение (Value) qiymati. "Объем продаж" ustunida so`rov natijaviy jadvali har bir yacheykasiga kiritiladigan har bir mahsulot uchun umumiy buyurtmalar hajmi hisoblanadi.

Kursorni keyingi bo`sh ustun Поле (Field) yacheykasiga o`mating **Format([ДатаРазмещения];"mmm")** ifoda kiriting. Funktsiya Format O bu xolda "ДатаРазмещения" ustunidagi sana qiymatidan oyning birinchi uch xarfini qaytaradi. Bu funktsiya urdamida biz so`rovda yana bir hisoblanuvchi maydon yaratdik, ifoda kiritish tugagandan so`ng Access avtomatik bu maydon nomini yaratadi - "Выражение!" (Expr1). Biz hisoblanuvchi maydonni aniqlamaganimizda doim shu yuz beradi. Bu ustunning Перекрестная Таблица yacheykasi ro`uxatidan Заголовки столбцов (Column Heading) qiymat tanlang. Bu shuni bildiradiki bu so`rov ustuni natijaviy jadval ustunlari sarlavhalari satriga aylantiriladi.



КодПредмет	КодПрепод	НомГр	НомУч	Оценка
11	1003	101	1	3
11	1003	101	2	4
11	1003	101	3	4
11	1003	101	4	5
11	1003	101	5	2
11	1003	101	6	4
11	1003	101	7	5
11	1003	101	8	4
11	1003	101	9	3
11	1003	101	10	3
11	1003	102	1	3
11	1003	102	2	4
11	1003	102	3	2
11	1003	102	4	4
11	1003	102	5	4
11	1003	102	6	5

Запуск (Run) tugmasini bosing. Natijaviy yozuvlar to`plami chiqadi.



Ustunlar fiksirlangan sarlavhalaridan kesishgan so`rovlarda foydalanish

Ko`rilgan so`rovda ustunlarni oylar kalendar navbati bo`yicha chiqarish uchun ustunlar fiksirlangan sarlavhalarni berish kerak:

Конструктор rejimiga o`ting va ekranga so`rov xossalarini chiqaring. Buning uchun instrumentlar panelida **Свойства** (Properties) tugmasini bosing va so`rov jadvallari aks etayotgan panel bo`sh maydonida sichqoncha chap tugmasini cherting. So`rov xossalari dialog oynasida **Заголовки столбцов** (Column Headings), xossasi chiqadi, u faqat kesishgan so`rovlarda bo`ladi.

Заказано : таблица

	Код заказа	Товар	Цена	Количество	Скидка
	10248	NuNuCa Nuss-Nougat-Creme	140,00p.	23	3%
	10248	Chartreuse verte	35,30p.	1	0%
▶	10248		98,00p.	10	0%
	10248		345,00p.	4	5%
	10248		348,00p.	5	0%
	10249		186,00p.	9	0%
	10249		424,00p.	40	0%
	10250		77,00p.	10	0%
	10250		168,00p.	15	15%
	10250		135,00p.	35	15%
	10251		168,00p.	6	5%
	10251		156,00p.	15	5%
	10251		168,00p.	20	0%
	10252		648,00p.	40	5%
	10252		20,00p.	25	5%
	10252		272,00p.	40	0%
	10253		100,00p.	20	0%
	10253	Cote de Blaye	144,00p.	42	0%
	10253	Chocolade	160,00p.	40	0%
	10254	Outback Lager	36,00p.	15	15%
	10254	Tourtiera	192,00p.	21	15%

Заголовки столбцов xossasi maydoniga yil xar bir oyining qisqacha nomini kiriting. Nomlar Format () funksiyasidan qanday qaytarilsa ustunlar sarlavhalarini standart akslantirishda qanday ko`rinsalar shunday ko`rinishlari kerak, aks xolda noto`g`ri berilgan oylar uchun ma'lumotlar akslanmaydilar. Nomlarni qavslarga olish kerak va nuqta vergul simboli bilan ajratish kerak. 12 oy nomi kiritib bo`lingandan so`ng so`rovlar oynasini berkiting.

So`rovda detalizatsiya darajasini o`zgartirish

Tahlil uchun qulay tasavvur yeki qulay grafik yaratish uchun kurilgan so`rov satrlari va ustunlari sonini qisqartirish mumkin. Buning uchun mahsulot tipi bo`yicha xar kvartal foydani chiqaradigan kesishgan so`rovni yaratamiz. Bunday so`rovni yaratish uchun:

Yaratilgan so`rovga