

O'zbekiston Respublikasi
Oliy va o'rta maxsus ta'lim vazirligi

Namangan muhandislik – pedagogika instituti

Kasb ta'limi fakulteti

Kasb ta'limi (Informatika va AT) kafedrası

Visual dasturlash tillari

fanidan

Ma'ruzalar matni

Namangan

«Visual dasturlash tillari» fanidan tayyorlangan ushbu ma`ruzalar matni 5111000-Kasb ta`limi (Informatika va axborot texnologiyalari) ta`lim yo`nalishi bo'yicha ta`lim olayotgan kunduzgi bo`lim talabalariga mo'ljallangan bo'lib, NamMPI ilmiy kengashi tomonidan tasdiqlangan na'munaviy o`quv dasturi asosida tayyorlandi.

Mualliflar:	Kasb ta`limi (Informatika va AT) kafedrası dotsenti O.Jakbarov Kasb ta`limi (Informatika va AT) kafedrası assistenti E.Qosimov
Taqrizchilar:	“Informatika va AT” kafedrası Dotsenti P.Karimov NamDU, “Informatika va amaliy matematika” kafedrası mudiri, dotsent A.Imomov

Ushbu ma`ruzalar matni Kasb ta`limi (Informatika va AT) kafedrası kafedrasining ilmiy uslubiy seminarida ko`rib chiqilgan va institut ilmiy-uslubiy kengashiga ko`rib chiqish uchun tavsiya etilgan.

(__ sonli majlis bayoni «__» ____ 2014 yil).

Namangan muxandislik-pedagogika instituti ilmiy-uslubiy kengashi tomonidan qo`rib chiqilgan va chop etishga tavsiya etilgan

(__ sonli majlis bayoni «__» ____ 2014yil).

Mavzu. Kirish. Visual Basic ni o'rnatish va ishlatish Visual Basic muhiti

Kompyuter texnologiyalari rivojlanib borayotgan ayni paytlarda turli imkoniyatlarga boy bo'lgan hamda o'zining aloxida yutuqlari bilan ajralib turuvchi dasturlash tillari ishlab chiqarilmoqda va ishlab chiqarildi. Dunyodagi keng dasturchilar ommasida foydalanilayotgan dasturlash tillarining aksariyati Windows operatsion sistemasida ishlaydi.

Jumladan Visual Basic dasturlash tili ham aynan Windows operatsion sistemasida ishlaydi.

Dasturlar tuzish sermashaqqat jarayon, lekin Visual Basic bu ishni sezilarli darajada soddalashtiradi. Visual Basic ning asosiy yutug'i shundaki, u Microsoft Office dasturlar paketi va Internet resurslari bilan ishlay oladi. Visual Basic bir necha versiyasida ishlab chiqariladi:

- Learning edition – boshlovchilar uchun juda mos bo'lib, Windows da ishlovchi dasturlar tuzish uchun zarur bo'lgan barcha elementlarni o'zida jamlagan;
- Professional edition – kompilyator dasturlari ishini tezlashtiruvchi ActiveX qo'shimcha komponentlarini va ma'lumotlar bazalarini boshqarishning kengaytirilgan vositalarini o'z ichiga oladi. Bu versiya jiddiy ilovalarni ishlab chiqarish uchun mo'ljallangan;
- Enterprise edition – ko'p foydalanuvchili ma'lumotlar bazalariga ega ilovalarni ishlab chiqishga imkogn beradi va ilovalarni dasturchilar guruhlar hamkorlikda ishlab chiqish vositalarini o'z ichiga oladi;
- Visual Basic for Applications - Visual Basicning mazkur versiyasi Microsoft Office dasturi tarkibiga kiradi. Agar sizda Office ning to'liq versiyasi yoki bu paketning istalgan komponenti bo'lsa, u olda sizda Visual Basic ning ilovalar uchun versiyasi ham bor bo'ladi.

Odatda Visual Basic ni o'rnatuvchi dasturlari to'plami kompakt diskda joylashgan bo'ladi. Agar bu dasturlar to'plami kompyuterda mavjud bo'lsa, uni o'rnatish uchun setup.exe faylini ishga tushirish lozim. Kompakt diskdan o'rnatish uchun diskni CD-ROM ga qo'yilishi bilan o'rnatuvchi dastur avtomatik tarzda ishga tushadi.

Visual Basic o'rnatilayotgan kompyuter kamida quyidagi talablarga javob berishi kerak:

- 486 yoki Pentium tipidagi processor;
- Windows operatsion sistema;
- Kamida 16 Mbayt operativ xotira;
- Vinchester 35 Mbaytdan yuqori xajmli.

Eslatma: Visual Basic ni o'rnatish jarayonida barcha aktiv dasturlar ishini yakunlash kerak

Dasturni o'rnatishda Setup dasturi o'rnatish variantlarini tanlashni so'raydi:

- Typical – bunda setup dasturi o'rnatish uchun zarur komponentlarni o'zi tanlaydi;
- Custom – kerakli komponentlar dasturchi tomonidan tanlab o'rnatiladi;
- Compact - Visual Basicni ishlashi uchun minimal imkoniyatli qilib o'rnatiladi.

Agar siz Custom (Maxsus) variantini tanlamagan bo'rsangiz, dasturni o'rnatish avtomatik tarzda davom etadi.

Agar Visual Basic ning qaysi komponentlari o'zingizga zarur ekanligini bilmasangiz, u holda Typical (Oddiy) variantini tanlash maqsadga muvofiqdir.



Visual Basicning o'rnatish variantlarini tanlashda kompyuterning texnik imkoniyatlariga e'tibor berish kerak, deb o'ylaysizmi?

Visual Basic 6.0 ni ishga tushirish

Windows asosiy menyusidan dasturni ishga tushirish uchun quyidagilarni bajarish kerak:

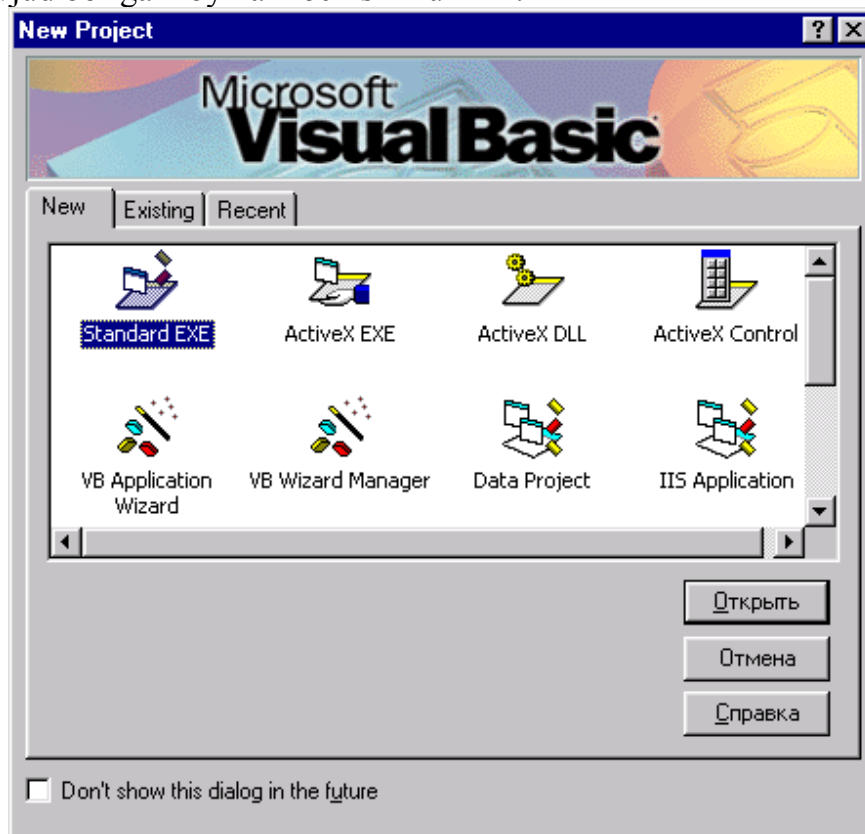
1. Ekraning quyi qismida joylashgan «Пуск» (Start) tugmasini bosing.

2. Windows asosiy menyusidagi «Программы» (**Programs**) ni tanlang. Menyuda ishga tushirish buyrug'i paydo bo'ladi.

3. **Microsoft Visual Studio 6.0** opsiyasini tanlang.

4. Navbatdagi menyusdan **Microsoft Visual Studio 6** ni tanlang.

Visual Basic 6 ni ishga tushirganda ekranda **New Project** muloqot oynasi chiqadi. Uning yordamida yangi loyiha uchun shablon tanlash, loyiha yaratish masterini ishga tushirish yoki mavjud bo'lgan loyihani ochish mumkin.



1-rasm

Bu oyna uchta ilovadan tashkil topgan:

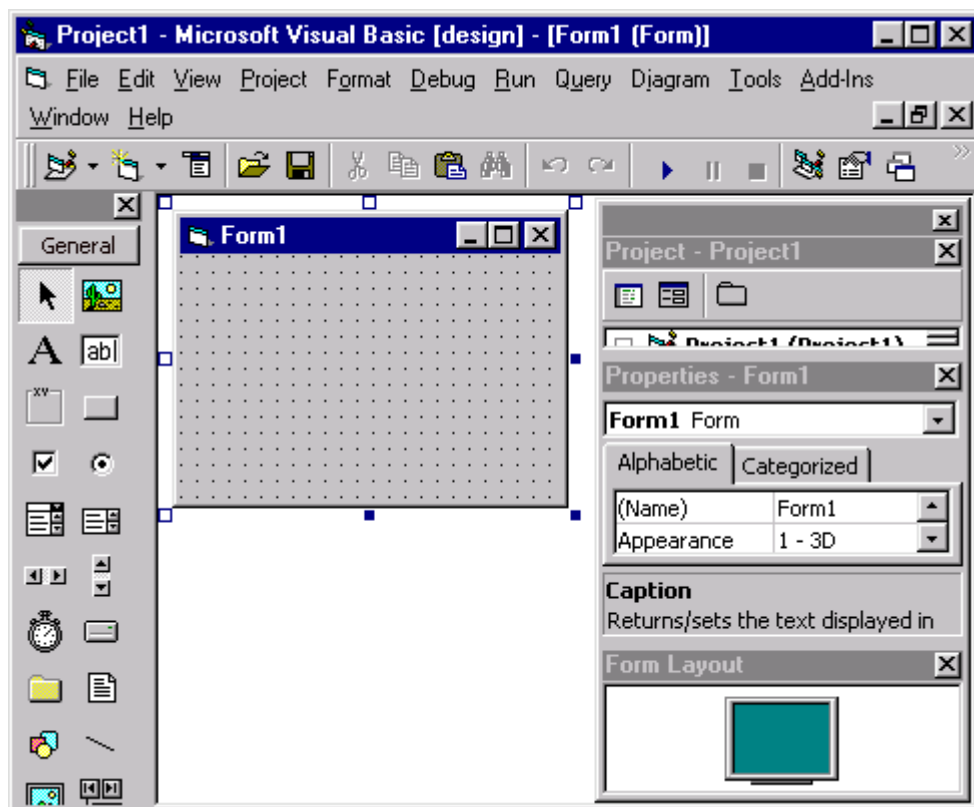
- ✓ **New** (Yangi) — shablonlardan va yangi loyiha yaratish uchun masterdan tashkil topgan;
- ✓ **Existing** (Mavjud) — ilgari yaratilgan loyihani va Visual Basic 6 ni loyiha-misollarini ochishga imkon beradi. Ushbu ilova yoyiladigan menyusga ega va u yordamida kompyuterda barcha mavjud bo'lgan papkalarni tanlash mumkin.
- ✓ **Recent** (Yaqinda yaratilgan) — Oxirgi vaqtda ochilgan loyihalarni o'z ichiga oladi.

Yangi loyihani yaratishda New ilovasi ishlatiladi. Unda mavjud bo'lgan loyiha shablon turlarini tanlash mumkin, lekin boshlang'ich bilim olish maqsadida standart ilovani tanlaymiz:

Standard EXE —bajariladigan standart ilovalar.

Integrallashgan ishlab-chiqarish muhiti

Integrallashgan ishlab-chiqarish muhiti (IDE) bizga ma'lim bo'lgan Microsoft ilovalarining boshqa turdagi grafik interfeysni namoyon qiladi. Uning tashqi korinishi 2-rasmda ko'rsatilgan.



2-rasm

Loyihalash muhitining tarkibiga quyidagi asosiy elementlar kiradi:

- ✓ Asosiy menyu;
- ✓ asboblarning standart paneli (**Standard**);
- ✓ boshqarish elementlari paneli;
- ✓ loyiha yurituvchi oynasi (**Project**);
- ✓ forma konstruktori;
- ✓ menyu tahrirlagichi (**Menu Editor**);
- ✓ xususiyatlar oynasi (**Properties**);
- ✓ forma maketining oynasi (**Form Layout**);
- ✓ ob'ektlarni ko'rish oynasi (**Object Browser**);
- ✓ dastlabki kodni tahrirlagich.

Asosiy menyu

Asosiy menyu Microsoft ilovalaridagi kabi ochilib-yopiluvchi qism menyularidan iborat qatordan tashkil topgan.

U quyidagi asosiy buyruqlardan iborat:

- ✓ **File** (fayl)
- ✓ **Edit** (Tahrirlash)
- ✓ **View** (Ko'rinish)
- ✓ **Project** (Loyiha)
- ✓ **Format** (Format)
- ✓ **Debug** (Rostlash)
- ✓ **Run** (Ishga tushirish)
- ✓ **Query** (So'rov)
- ✓ **Diagram** (Diagramma)
- ✓ **Tools** (Servis)

- ✓ **Add-Ins** (Sozlash)
- ✓ **Window** (Oyna)
- ✓ **Help** (Yordam)

Asosiy menyu ko'rinishi 3-rasmda keltirilgan.



3-rasm

Menyu buyruqlarining asosiylari bilan tanishib chiqamiz.

File menyusi:

<i>Buyruq</i>	Vazifasi
New Project	YAngi proekt yaratish
Open Project	Mavjud proektni ochish
Add Project	Bir nechta proektlar bilan parallel ishlash uchun guruxga yangi proekt qo'shish.
Remove Project	Proektlar guruxidan proektni o'chirish
Save Project Group	Proektlar guruxini diskka saqlash
Save Project Group As	Proektlar guruxini diskka boshqa nom bilan saqlash
Save <forma nomi>	Forma ko'rsatilgan nom bilan saqlash
Save <foma nomi> As	Formani boshqa nom bilan saqlash
Print	Print muloqot oynasini ochish
Print Setup	CHop etishni sozlash muloqotoynasini ochish
Make<proektnomi>.exe	Joriy proektning bajariluvchi (ishchi) faylini xosil qilish
Make Project Group	Proektlar guruxini tashkil etish.
Exit	Visual Basic dan chiqish.

Edit menyusi:

<i>Buyruq</i>	Vazifasi
Undo	Oxirgi bajarilgan buyruqni bekor qilish
Redo	Oxirgi bekor qilingan buyruqni qayta tiklash
Cut	Matn qismini qirqib olish
Copy	Matn qismidan almashish buferiga nusxa olish
Paste	Almashish buferidagi matnni kursor turgan joyga qo'yish
Remove	Belgilanganni ko'chirish
Delete	Belgilanganni o'chirish
Find	Izlash muloqot oynasini yuklash
Replace	O'rniga qo'yish muloqot oynasini yuklash

View menyusi:

<i>Buyruq</i>	Vazifasi
Code	Kodlarni taxrirlash oynasini ochish
Object	Forma konstruktori oyanasini ochish

Object Browser	Ob`ektlar brauzeri oynasini ochish
Properties Window	Ob`ekt xususiyatlari (Properties) oynasini ochish
Form Layout Window	Formalar maketi oynasini ochish
Toolbox	Boshqaruv elementlari oynasini ochish

Project menyusi:

<i>Buyruq</i>	Vazifasi
Add Form	Proekt oynasiga yangi forma qo`shish
Add Module	Dasturiy modul qo`shish
Add File	Fayl qo`shish
Remove	Proektdan faylni o`chirish
Component	Boshqaruv elementlari paneliga yangi komponentlarni o`rnatish uchun Component muloqot oynasini ochish
Project Properties	Proekt xususiyatlarini o`gartirish imkoniyatini beruvchi Project Properties muloqot oynasini ochish

Format menyusi:

<i>Buyruq</i>	Vazifasi
Align	Ob`ektlarni joylashtirish buyruqlari joy olgan menyuni ochish
Make Same Size	Formadagi ob`ektlarning o`lchamlarini boshqaruvchi buyruqlar joy olgan menyuni ochish
Horizontal Spacing	Ob`ekt va gorizontal yeo`nalish orasidagi intervalni o`rnatish
Vertical Spacing	Ob`ekt va vertikal yeo`nalish orasidagi intervalni o`rnatish
Center in Form	Formadagi ob`ektlarni markazga joylashtirish
Order	Formadagi ob`ektlarni bir-biriga nisbatan yuqori yoki quyi qismiga yeo`naltiradi

Debug menyusi:

<i>Buyruq</i>	Vazifasi
Step Into	Proceduralarni bajarilishini qadamba-qadam tashkil etish (murojaat etiluvchi proceduralarni ham)
Step Over	Proceduralarni bajarilishini qadamba-qadam tashkil etish (murojaat etiluvchi proceduralardan tashqari)
Step Out	Joriy proceduralardan chiqquncha proceduralarni bajarilishini qadamba-qadam tashkil etish

Toggle Breakpoint	Bajarilishni to'xtatish nuqtasini o'rnatish
Clear all Breakpoints	Barcha to'xtatish nuqtalarini bekor qilish

Run menyusi:

<i>Buyruq</i>	Vazifasi
Start	Dasturni ishga tushirish
Start with Full Compile	Dasturni to'liq kompilyatsiya bilan ishga tushirish
Break	Dasturni ishchi xolatidan to'xtatish
End	Dasturning ishini bekor qilish
Restart	Dasturni boshqatdan ishga tushirish

Tools menyusi:

<i>Buyruq</i>	Vazifasi
Add Procedure	Prosedura qo'shish
Procedure Attributes	Prosedura atributlarini o'rnatish
Menu editors	Menyu taxrircisi oynasini ochish
Options	Visual Basic parametrlarini sozalsh uchun Options muloqot oynasini ochish

Asosiy menyuning ko'pchilik buyruqlari Windows ilovalarida ishlatiladi (masalan, Microsoft Word yoki Microsoft Excel). Shu sababdan ular alohida ko'rib chiqilmaydi. Zarur bo'lgan tushintirishlar materiallarni yetqazib berish jarayonida beriladi.

Menyu pastida joylashgan, ko'proq ishlatiladigan buyruqlar menyu standart asboblari panelida kichik rasmchali tugmalar shaklida tasvirlangan.

Mavzu. Visual Basicda boshlang'ich amallar va proektlar. Dastur matnining umumiy ko'rinishi.

Standart asboblari paneli

Standart asboblari paneli asosiy menyu ostida joylashgan. Asboblarning standart panelida ko'p ishlatiladigan menyu buyruqlarini chaqirish uchun tugmachalar joylashgan.



4-rasm

Standart asboblari panelining ko'pchilik tugmachalari tavsiyalari boshqa Windows ilovalari tavsiyalari bilan mos keladi. Qator tugmachalar visual dasturlash muhitining o'ziga xos maxsus funksiyalarni bajaradi (masalan,  **Menyu Editor** (Tahrirlash menyusi) yoki



Toolbox (Boshqarish elementlar paneli)). Quyida standart panelning tugmalarining tavsiyalari jadvali keltirilgan.

Asbobarning standart panelining tugmachalari

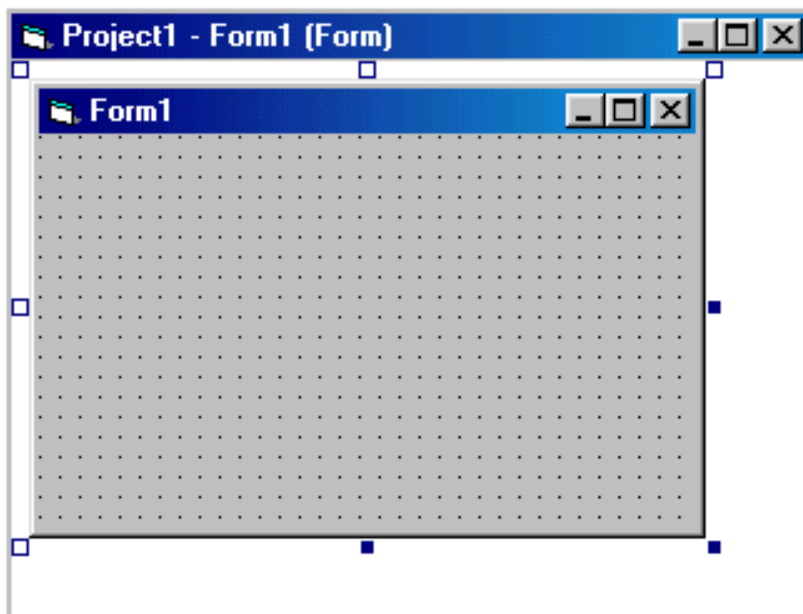
Tugma	Nomi	Tavsiya qilinishi
	Add Standard EXE Project (Standart loyiha qo'shish)	Standart exe-loyiha qo'shadi
	Add Form (Forma qo'shish)	Loyihaga forma qo'shadi
	Menu Editor (Tahrirlash menyusi)	Tahrirlash menyusini chaqiradi
	Open Project (Loyihani ochish)	Loyihani ochadi
	Save Project (Loyihani saqlash)	Loyihani saqlaydi
	Cut (Kesib olish)	Almashish buferiga axborotni qirqib oladi
	Copy (Nusxalash)	Almashish buferiga nusxa oladi
	Paste (Qo'yish)	Almashish buferidan axborot qo'yadi
	Find (Qidirish)	Kontekst bo'yicha axborot qidirishni amalgam oshiradi
	Can't Undo (oldingini bekor qilish)	O'dingi holatni bekor qiladi
	Can't Redo (Takrorlashni bekor qilish)	Bekor qilingan holatni tiklash
	Start (Ishga tushirish)	Dasturni ishga tushiradi
	End (Yakunlash)	Dastur bajarilishini yakunlaydi
	Break (To'xtatish)	Dastur bajarilishini to'xtatadi
	Project Explorer (Loyihalar sharhlovchisi)	Loyihalar sharhlovchisining oynasini ochadi
	Properties Window (Xususiyatlar oynasi)	Xususiyatlar oynasini ochadi
	Form Layout Window (Forma maketi oynasi)	Forma maketi oynasini ochadi
	Object Browser (Ob'ektlar brauzeri)	Ob'ektlar brauzeri oynasini ochadi
	Toolbox (Boshqarish elementlari paneli)	Boshqarish elementlari panelini ochadi
	Data View Window (Ma'lumotlarni ko'rish oynasi)	Ma'lumotlarni ko'rish oynasi ochadi
	Visual Component Manager (Vizual komponentlarini boshqarish)	Visual Component Manager vizual komponentlarini boshqarish oynasini ochadi

Forma konstrukturi oynasi

Ilovalarni visual loyihalashda forma konstruktori oynasi asosiy ishchi oyna hisoblanadi (5-rasm). Bu oynani chaqirish uchun asosiy menyuning **View** (Ko‘rinish) menyusidagi **Object** (Ob‘yekt) buyrug‘i yoki loyihalar sharxining **Forms** guruhidagi ob‘yektning kontekst menyusining **View Object** buyrug‘i tanlanadi.

Formalar konstruktori oynasidagi barcha formalar va ilova ob‘yektlari konstruksiya qilinadi. Formadagi ob‘yektlarni aniq pozitsiyalashda oynada to‘r hosil bo‘ladi.

Formaning va sichqonchaning ajratgich markerinidan foydalanib oynadagi formaning o‘lchamlarini o‘zgartirish mumkin. Formaning o‘lchamlarini o‘zgartirish uchun sichqoncha ko‘rsatkichini markerga o‘rnatish kerak, u ikki yo‘nalishli strelka ko‘rinishiga ega bo‘lganda uni talab etilgan o‘lchanlargacha siljutiladi.



5-rasm



6-rasm

Boshqarish elementlari paneli

Tugma	Nomi	Tavsiya etilishi
	Pointer (Ko‘rsatkich)	Sichqoncha markerini (ko‘rsatkichini) pozitsiyalash uchun ishlatiladi
	PictureBox (Grafik oyna)	Guruhga elementlarni birlashtirish uchun, unga grafik tasvirlarni, matnni, grafik elementlarni va animatsiyani chiqarish uchun mo‘ljallangan grafik oynani formaga joylashtiradi.
	Label (Belgi)	Formaga matnli axborotlarni, yozuvlarni va eslatmalrni chiqarish uchun mo‘ljallangan ob‘yektlarni joylashtiradi.
	TextBox (Matnli maydon)	Formaga matnli axborotlarni, sonlarni va vaqtlarni kiritish uchun foydalaniladigan matnli maydonlarni joylashtiradi.
	Frame (Ramka)	Ob‘yektlarni mantiqiy guruhga oladigan sarlavhali ramkani formaga joylashtiradi.
	CommandButton (Boshqarish tugmasi)	Formaga initsiyatsiya qilish, buyruqlarni bajarish, dasturni ishga tushirish uchun mo‘ljallangan boshqarish tugmalarini joylashtiradi.

Boshqarish elementlari paneli – bu ilova formasini visual ishlab-chiqishda asosiy ishchi asbobi hisoblanadi (6-rasm). Boshqarish elementlari paneli **View** (Ko‘rinish) menyusidagi

Toolbox buyrug'ini tanlash orqali chaqiriladi. Bu panelni chaqirishning yana bir usuli standart asboblarning panelining **Toolbox** tugmasi bosiladi.

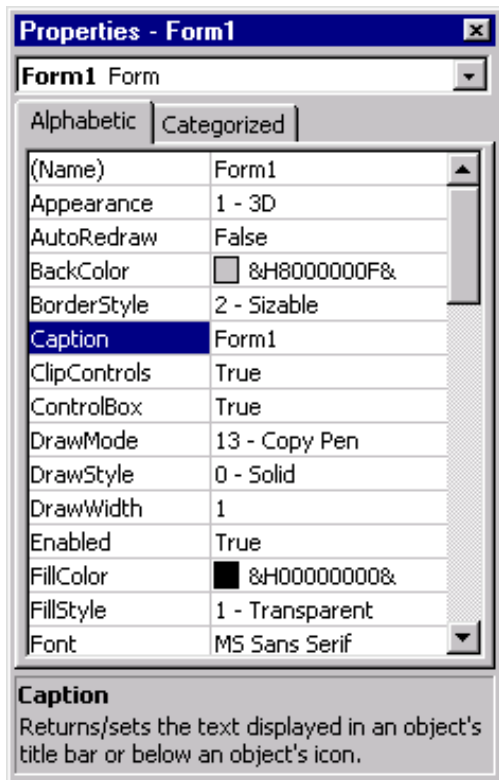
Boshqarish elementlar paneli tarkibiga formani boshqarishning asosiy elementlari – belgilar, matnli maydonlar, tugmalar, ro'yxatlar va forma maketini tezroq visual loyihalash uchun boshqa elementlar kiradi.

	CheckBox (Bayroqcha)	Dasturning bajarilishi uchun shartlarni shakllantirish yoki “ha/yo‘q” tamoyilida ishlovchi qandaydir sozlashlar uchun bayroqchalarni formaga joylashtiradi.
	OptionButton (O‘chirib-yondirg‘ich)	Ishlash rejimini tanlash yoki dasturning bajarilishini sozlash uchun o‘chirib-yondirg‘ichlarni formaga joylashtiradi.

Boshqarish elementlari paneli tugmalari

Boshqarish elementlarini formaga elementlar paneli yordamida joylashtirish quyidagicha

	ComboBox (Поле со списком)	Formada birvaqtning o‘zida kiritish maydonidan va ochilib-yopiluvchi ro‘yxatdan tashkil topgan ob‘yektini yaratadi.
	ListBox (Ro‘yxat)	Formada tavsiya etilgan ro‘yxat qiymatlarining bittasini yoki birnechtasini tanlash uchun mo‘ljallangan ro‘yxat yaratadi.
	HScrollBar (Gorizantal chizg‘ich)	Formaga berilgan diapazondagi qiymatni tanlash uchun siljitgich sifatida ishlatiladigan gorizantal kesmani joylashtiradi.
	VScrollBar (Vertikal chizg‘ich)	Formaga berilgan diapazondagi qiymatni tanlash uchun siljitgich sifatida ishlatiladigan vertical kesmani joylashtiradi.
	Timer (Taymer)	Formaga taymerni joylashtiradi.
	DriveListBox (Qurilmalar ro‘yxati)	Formada qurilmalar ro‘yxatini yaratadi.
	DirListBox (Papkalar ro‘yxati)	Formada papkalarining daraxt ko‘rinishini yaratadi.
	FileListBox (Fayllar ro‘yxati)	Formada fayllar ro‘yxatini yaratadi.
	Shape (Shakl)	Formada geometrik shakllarni – to‘g‘riburchaklik, kvadrat, aylana, ellips, doira, aylana burchakli to‘g‘riburchaklik va kvadrat yaratadi.
	Line (Chiziq)	Chiziqlar yaratadi.
	Image (Tasvir)	Formaga grafik tasvirlarni joylashtirish uchun mo‘ljallangan maydon yaratadi.
	Data (Ma’lumotlar)	Formada yozuvlar bo‘yicha siljishlar va navigatsiya natijasini tasvirlash uchun ma’lumotlar bazasida ma’lumotlarni boshqarish elementlarini yaratadi.



7-rasm

amalgam oshiriladi:

1. Sichqoncha yordamida talab qilingan boshqarish elementini tanlangiz.
2. Forma qurish (konstruktor) oynasiga o'tingiz. Shuning bilan sichqoncha ko'rsatkichi jo'ylashtirilgan ob'yekt joyini o'rnatishga imkon beradigan krest holatiga o'tadi. Sichqonchaning chap tugmasini bosgan holda yangi ob'yektning pozitsiyasi o'lchamlarini kiritingiz.

Xossalar oynasi

Xossalar oynasi (**Properties**) forma va unga joylashtirilgan ob'yektlarning xossalarini sozlash va tasvirlash uchun qo'llaniladi. Unga, masalan, belgilab olingan ob'yekt xossasi formadagi joylashgan pozitsiyasi, balandligi, kengligi, rangidan tashkil topgan (7-rasm).

Xossalar oynasidagi Caption xossasiga «Yozuv matnini o'zgartirish» tugmasining sarlavhasini kiriting. Sarlavhani bosish jarayonida & belgi maxsus qiymatga ega va aks ettirilmaydi. ALT tugmasi va tagi chizilgan belgi yordamida siz ob'yektni sichqonsiz tanlash imkoniga ega bo'lasiz. Bu klavishlarning ekvivalent kombinatsiyasi deyiladi.

1-misol: “Kubik tashlash”

Navbatdagi dastur yordamida biz boshqarish paneli elementlarini ishlatishni o'rganamiz:

Formada “1” dan “6” gacha bo'lgan butun sonlar chiqadigan ikkita belgi ob'yekti yaratiladi – o'yin kubigini tashlash imitatsiyasi. Rnd datchigi yordamida tasodifiy sonlar olinadi. Agar ikkita sonning yig'indisi 6 dan katta bo'lsa, yutuq haqida xabar va unga mos tasvir chiqadi, aks holda – yutqazganligi haqida xabar va boshqa tasvir chiqadi.

Dasturdagi a va b o'zgaruvchilarining tahlili, yutuq shartini tekshirish operatori, Rnd funksiyasi darslikning kelasi boblarida keltirilgan. Shu sababli dastur kodiga kiritiladigan buyruqlarni o'zgarishlarsiz o'tkazish tavsiya etiladi. Hozircha bizni “Belgi” (**Label**) va “Tasvir” (**Image**) kabi ob'yektlarni yaratish e'tiboringizga jalb qilinadi.

1. Visual Basicni ishga tushiramiz va yangi standart ilova yaratamiz.
2. Boshqarish panelining “Belgi” (**Label**) ob'yektini tanlab olamiz. Formada belgini chizamiz. **Label1** ob'yekti yaratiladi va shunga uxshash yo'l bilan **Label2**, **Label3** belgi ob'yektlari yaratiladi.
3. Boshqarish panelidan buyruq tugmasi (**Command Button**) tanlanadi va uni formada chizamiz (**Command1**). Yana bitta buyruq tugmasi yaratilib, formada chizamiz (**Command2**).
4. Boshqarish panelidan “Tasvir” (**Image**) tugmasi tanlanadi. Formaga tasvirni joylashtirish uchun unga oyna yaratiladi. Bu amallarni bajargandan keyin forma quyidagicha ko'rinishga ega bo'ladi:

5. **Label1**, **Label2**, **Label3** ob'yektlarining **Caption** xossasi (xossalarning mos maydonlari tozalanadi) va **Command1**, **Command2** ob'yektlarining ham **Caption** xossasi (tugmalar nomini yozish orqali) o'zgartiladi. Mazkur ob'yektlarning barobarlashtirish va shrift xossalri o'zgartiriladi. Tasvir (**Image**) ob'yektining "Cho'zish" (**Stretch**) xossasini ham o'zgartiramiz. Bu xossasini o'rnatilayotganda uning **True** qiymatida kartina tasvir oynasida aniq chiziladi.

6. **Command1** buyruq tugmasiga quyidagicha dastur kodi yoziladi:

```
Private Sub Command1_Click()  
    Dim a, b  
    Label1.Caption = Int(Rnd * 6) + 1  
    a = Int(Label1.Caption)  
    Label2.Caption = Int(Rnd * 6) + 1  
    b = Int(Label2.Caption)  
    If a + b > 6 Then  
        Label3.Caption = "YPA!!! Siz yuttingiz!!!"  
        Image1.Picture = LoadPicture("c:\Image\smile.gif")  
    Else  
        Label3.Caption = "Afsus... Siz yutqazdingiz..."  
        Image1.Picture = LoadPicture("c:\Image\cry.gif")  
    End If  
End Sub
```

Bu dasturning kodida **Label1.Caption = Int(Rnd * 6) + 1** qatorida 0 dan 5 gacha bo'lgan intervaldagi tasodifiy butun sonni hisoblaydi, 1 ga oshiradi, olingan qiymat esa **Label1** ob'yektining **Caption** xossasiga o'zlashtiriladi

Image1.Picture = LoadPicture("c:\Image\smile.gif") qatorida **Image1** ob'yektining **Picture** xossasiga LoadPicture funktsiyasi qaytaradigan qiymatni o'zlashtiradi.

Mavzu. Vizual Basic dasturlash tili elementlari. O'zgaruvchilar

O'zgaruvchilar operativ xotirada ma'lumotlarni vaqtincha saqlash uchun o'zini rezervga olingan holda e'lon qilinadi. Har bir o'zgaruvchi shaxsiy nomlarga ega bo'ladi. O'zgaruvchiga qiymat o'zlashtirilgandan keyin biz uni va qiymatini dasturda ishlatishimiz mumkin.

O'zgaruvchilar **Dim** (dimension – o'lcham degan so'zdan kelib chiqqan) operatori yordamida e'lon qilinadi. **Dim** operatori o'zgaruvchi bilan operativ xotiraning aniqlangan sohasini rezervga oladi. O'zgaruvchini elon qilish uchun o'zgaruvchi nomini Dim operatoridan keyin yozish darkor. O'zgaruvchining nomidan keyin uning turini ko'rsatish maqsadga muvofiq keladi.

Masalan, Dim X - X o'zgaruvchisini turi ko'rsatilmadan e'lon qilinadi,

Dim X As Integer - X o'zgaruvchisini turi ko'rsatilgan holda ya'ni butun deb e'lon qiladi.

Visual Basic dasturlash muhitida ixtiyoriy turdagi ma'lumotlarni qabul qiladigan Variant turiga ega. Variant turiga mansub o'zgaruvchi foydalanishga juda qulay, lekin o'zgaruvchilarni aniq bir turda e'lon qilish operativ xotirani tejam ishlatilishiga imkon beradi va dasturning ishlashini tezlashtiradi.

O'zgaruvchi nomlari

O'zgaruvchilar o'qishga qulay va birqancha ko'rgazmali bo'lishi uchun qandaydur ma'noga ega bo'lgan nomlar beriladi. O'zgaruvchilar nom berishda bir nechta qoidalar mavjud:

- O'zgaruvchi nomi 255 dan oshmagan belgilardan iborat bo'lishi mumkin;
- O'zgaruvchi nomi ixtiyoriy harf va raqamlardan tashkil topishi mumkin;
- O'zgaruvchi nomining birinchi belgisi harf bulishi shart;
- O'zgaruvchi nomida probellar bo'lmasligi kerak;
- O'zgaruvchining nomi uzunligi unikal va ko'rish chegarasidan oshmasligi kerak.

Masalan, o'zgaruvchi nomlari quyidagicha bo'lishi mumkin: CurrentNum, Total, Date_of_birth. Navbatdagi o'zgaruvchi nomlari bo'lishi mumkin emas: 1Time, \$Total, Date of birth.

Ma'lumotlarning turiga qarab o'zgaruvchi nomlarining prefikslari tavsiya qilinadi. Ular quyidagi jadvalda keltirilgan:

2.2.	Ma'lumotlarning turi	Prefiks	Misol
	Boolean	bin	binSuccess
	Currency	cur	curPrice
	Date	dtm	dtmFinish
	Double	dbl	dblSum
	Integer	int	intQuantity
	Long	lng	lngTotal
	Single	sng	sngLength
	String	str	strLastname
	Variant	vnt	vntValue

Ma'lumotlarning turi

Visual Basic dasturiy muhitida quyidagi ma'lumlarning asosiy turlarini ishlatishingiz mumkin:

- sonli (**Integer, Long, Single, Double, Currency**);
- qator (**String**);
- sana (**Date**);
- mantiqiy (**Boolean**);
- ixtiyoriy (**Variant**);

Ma'lumotlarning asosiy turlarining jadvali

Ma'lumotlarning turlari	O'lchami (baytlarda)	Qiymatlar diapazoni	Misollar
Integer (Butun)	2 bayt	-32 768 dan + 32 767 gacha	Dim intX as Integer intX = 23546
Long (Uzun butun)	4 bayt	-2 147 483 648 dan +2 147 483 647 gacha	Dim lngY as Long lngY = 1000000
Single (Bir karra aniqlikdagi siljuvchi nuqtali o'nlik son)	4 bayt	-3.402823E38 dan +3.402823E38 gacha	Dim sngA as Single sngA = 4.781
Double	8 bayt	-1.79769313486232D308	Dim dblB as

(Ikki karra aniqlikdagi siljuvchi nuqtali o'nlik son)		dan +1.79769313486232D308 gacha	Double dblB = 1.000000000001
Currency (Pullik)	8 bayt	от -922337203685477.5808 до +922337203985477.5807	Dim curM as Currency curM =25456.20
String (Qator)	har bir belgi 1 bayt	1 dan 1 65535 belgigacha	Dim strS as String strS = "Basic"
Boolean (Mantiqiy)	2 bayt	True (Rost) yoki False (Yolg'on)	Dim binL as Boolean binL = True
Data (Sana)	8 bayt	January-1-100 (1.01.0100) dan December-31-9999 gacha	Dim dtmD as Data dtmD = #03-08- 2005#
Variant (Variant)	16 bayt (son uchun) 22 bayt + bitta belgi 1 bayt (qator uchun)	Barcha turlar uchun	Dim vntV vntV = 13.45 yoki vntV = "Text"

O'zgaruvchilarni e'lon qilish

Visual Basic dasturlash muhitida o'zgaruvchilarni e'lon qilish quyidagi sintaksis bo'yicha amalga oshiriladi:

Dim o'zgaruvchiNomi [As Oma'lumotlarTuri]

Quyidagicha misollarni keltirish mumkin:

Dim bInSuccess As Boolean

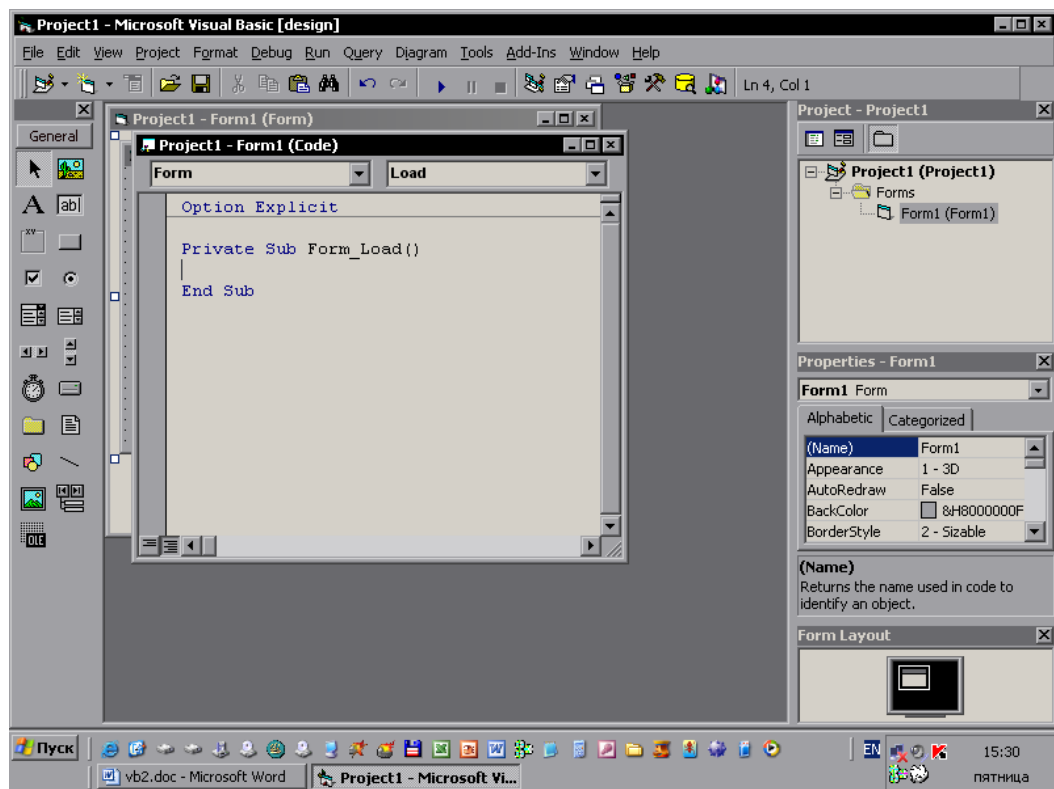
Dim strLastname As String, dblSum As Double

Fiksirlangan uzunlikdagi qatorlarni e'lon qilish uchun quyidagi sitaksis ishlatiladi:

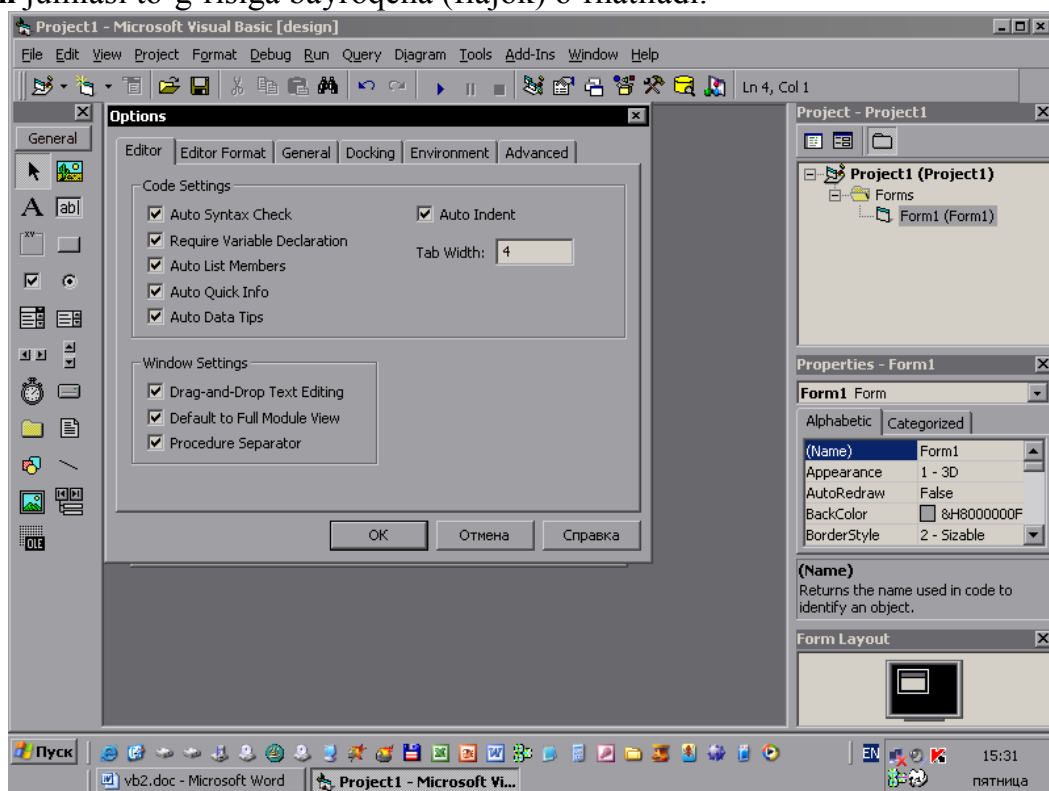
Dim O'zgaruvchinomi As String * O'zgaruvchiuzunligi

O'zgaruvchiUzunligi parametri belgilarning maksimal sonini ko'rsatadi. (*) belgisi esa o'zgaruvchi uzunligining fiksirlanganligini ko'rsatadi.

O'zgaruvchilarni aniq turda e'lon qilish orqali dastur translyatsiyasi rejimini o'rnatish tavsiya etiladi. Buning uchun modulning boshida **Option Explicit** (Aniq turda e'lon qilish) operatorini kiritish kerak.



Bu operatorni Visual Basicning dastur oynasidagi barcha modullarga avtomat ravishda qo'shish uchun **Tools** (Servis) menyusidagi **Options** (Parametrlar) buyrug'i bajariladi. **Options** muloqot oynasi ochiladi va undagi **Editor** ilovasining **Require Variable Declaration** jumlasiga bayroqcha (flajok) o'rnatiladi.



Agar Option Explicit operatorini modulga kiritmasangiz, unda siz o'zgaruvchini noaniq e'lonidan foydalanishingiz mumkin. Bunday holatda o'zgaruvchining turi birinchi o'zlastirish operatoridayoq aniqlanadi va shu vaqtning o'zida o'zgaruvchiga xotiradan joy ajratiladi.

O'zgaruvchiga qiymatlarni o'zlashtirish

Dasturda o'zgaruvchini ishlatishdan avval unga qiymat berish zarur. O'zlashtirishning eng sodda usuli “=” o'zlashtirish operatorini ishlatishdan iborat. U quyidagi ko'rinishga ega bo'ladi:

O'zgaruvchi = ifoda

O'zgaruvchi argumentiga o'zgaruvchi nomi beriladi va unga **ifoda** qiymati o'zlashtiriladi. Masalan,

sngFirst = 10

strLastname = "Ivanov"

Tenglik belgisidan o'ng tomonida faqat doimiylar joylashib qo'ymasdan birqancha murakkab ifodalar ham joylashishi mumkin.

sngResult = sngFirst + 255

strName = "ИВАНОВ" & ": " & strTeam

Variant turidagi o'zgaruvchialarni ishlatish avzalliklari

Variant turidagi o'zgaruvchialarni barcha turdagi ma'lumotlarni saqlash va amallarni bajarish uchun ishlatiladi. Ikkita holatni esdan chiqarmaslik kerak, ya'ni birinchidan, Variant turidagi o'zgaruvchilar ustida arifmetik amallar va funksiyalarni faqat u sonli qiymatga ega bo'lgandagina ishlatish mumkin, ikkinchidan, qatorlarni konkatenatsiya qilishda “+” operatorining o'rniga & operatorini ishlatish kerak.

Variant turidagi o'zgaruvchilar **Empty**, **Null** yoki **Error** kabi maxsus qiymatlarga ham egadur.

- **Empty qiymati.** Variant turidagi o'zgaruvchilarga 0 dan farqli qiymatlarni, bo'sh qatorni yoki **Null** qiymatini o'zlashtirish uchun **Empty** nomlanadi. **Empty** qiymatini aniqlash uchun **IsEmpty** funksiyasini ishlatish mumkin:
If IsEmpty(x) Then x = 0
- **Null qiymati.** Variant turidagi o'zgaruvchilar ma'lumotlar bazasi bilan ishlaydigan ilovalardagi bo'sh ma'lumotlarni ko'rsatish uchun **Null** qiymati ham ishlatish mumkin. **IsNull** funksiyasi yordamida Variant turidagi o'zgaruvchi **Null** qiymatiga egaligini tekshiradi. Variant turidagi o'zgaruvchiga **Null** qiymatini o'zlashtirish uchun **Null** kalt so'zini ishlatish mumkin: y = Null Variant turidagi ega bo'lmagan o'zgaruvchiga **Null** qiymatini o'zlashtirish xato bo'ladi.
- **Error qiymati.** Variant turidagi o'zgaruvchi proceduradagi hosil bo'lgan xatolar uchun Error qiymatini qabul qilishi mumkin. Lekin bu holatda ilova darajasidagi xatolar to'g'rilanmaydi. Ammo dasturchi uning qiymatiga qarab aniq bir tuzatishlar kiritishi mumkin.

Variant turidagi qiymatlarning ichki tasavvuri

Variant turidagi o'zgaruvchilar unga saqlanadigan ma'lumotlarning ichki tasavvurini qo'llab quvvatlaydi. Variant turidagi o'zgaruvchiga qiymatlar berishda Visual Basic muhiti bu qiymatlarning birqancha qulayliklarini qo'llaydi. Masalan, agar Variant turidagi o'zgaruvchiga kasr qismiga egamas katta bo'lmagan sonli qiymat o'zlashtirilgan bo'lsa, unda **Integer** tasavvuri ishlatiladi, agar kasr son o'zlashtirilgan bo'lsa, unda **Double** ichki tasavvuri ishlatiladi.

Doimiylar

Doimiy deganda dasturning bajarilishi jarayonida o'zgarmaydigan ifoda elementi tushiniladi. Bir nechta misollar keltiramiz:

- 142.34 – sonli doimiy;
- 5.6E+4 - sonli doimiy (равна 5 600);
- " Faylni o'qishda nosozlik" - simvolli doimiy;
- #09/01/2005# - sana turidagi doimiy;
- False – mantiqiy doimiy.

Doimiylarni e'lon qilish

Doimiylarni e'lon qilish uchun quyidagicha sintaksisga ega bo'lgan **Const** operatori ishlatiladi:

Const doimiyNomi [As ma'lumotTuri] = ifoda

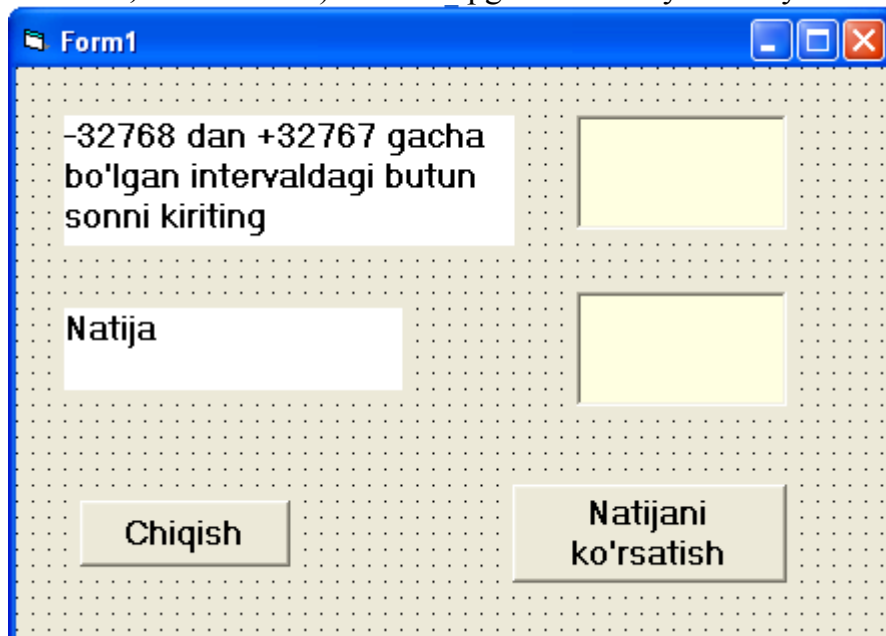
Masalan, **Const** strErrorFile As String = "Faylni o'qishda nosozlik"

O'zgaruvchilarning tavsifi

3-misol

IntX butun sonining tavsifi. Uning qiymati **Text2** matn maydoniga, natija esa **Text2** matn maydoniga kiritiladi. Dasturda quyida tavsif qilingan, qatorni songa va sonni qatorga almashtiradigan funksiya foydalaniladi.

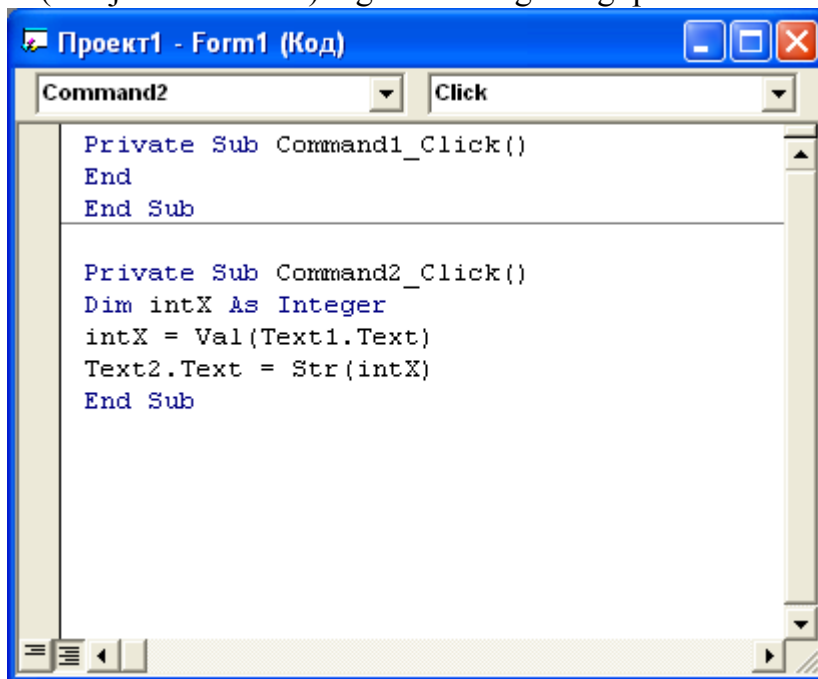
Ikkita belgidan (**Label1**, **Label2**), ikkita matn maydonindan (**Text1**, **Text2**) va ikkita tugmadan (**Command1**, **Command2**) tashkil topgan forma loyihasini yaratamiz.



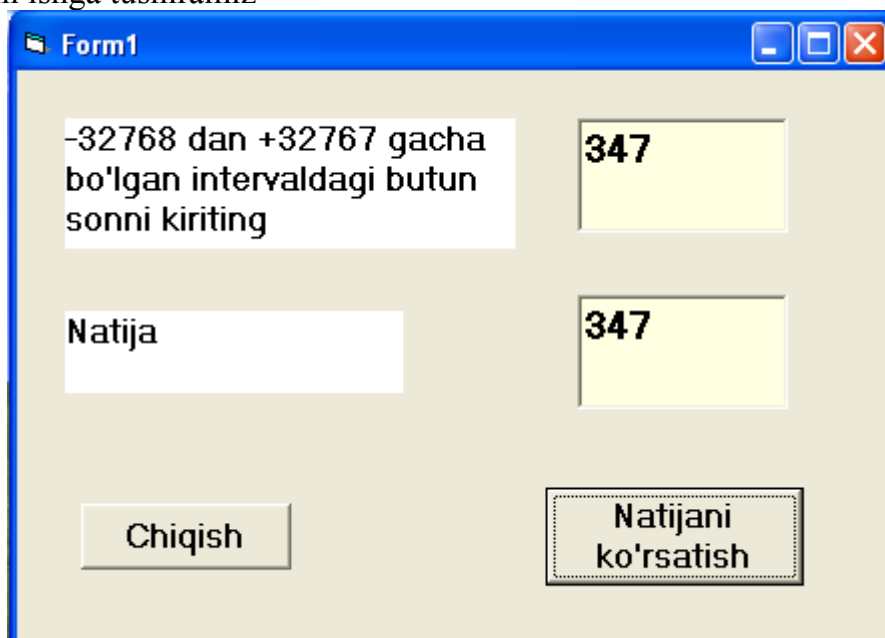
Ob'yekt	Xossa	O'rnatilgan qiymatlari
Label1	Caption Font BackColor	-32768 dan +32767 gacha bo'lgan intervaldagi butun sonni kiriting
Label2	Caption Font	Natija

	BackColor	
Command1	Caption	Chiqish
Command2	Caption	Natijani ko'rsatish
Text1 Text2	Text	Text xossasi maydonini tozalash

Command2 (Natijani ko'rsatish) tugmasi bosilgandagi protseduraning kodini yozamiz



Loyihani ishga tushiramiz



Shunga e'tibor qilish kerakki, ya'ni agar kiritilgan son qiymatlar diapazonidan tashqarida bo'lsa, **Overflow** to'lib ketish xatosi yuz beradi.

3. Kommentariyalar

Kommentariya dastur matniga tushuncha berish uchun ishlatiladi. Dastur matniga kommentariya kiritish uchun qatorning boshiga yoki kommentariya kiritilayotgan matn

boshiga (') belgisi kiritiladi. Bu belgidan keyin kelgan ixtiyoriy mant kommentariya deb qabul qilinadi, ya'ni Visual Basic ularni transliyatsiya qilmaydi. Masalan,
' Qator boshidan boshlanuvchi kommentariya
 $X = \sin(a) + \cos(b)$ ' Bu operatorndan keyin kommentariya

Operatorni birnechta qatorga joylashtirish

Agar operator katta uzunlikga ega bo'lsa, uni qatorni davom etish belgilaridan foydalanib, bir nechta qatorga bo'lish mumkin. Bu belgilar probel pastki chiziq () hisoblanadi.

Masalan, familiya, ism va sharflarni birlashtiruvchi qatorni bir nechta qatorga joylashtiramiz:

```
strName = strFamaly & strName & strFatherName
```

Buni quyidagicha yozish mumkin:

```
strName = strFamaly _  
& strName & strFatherName
```

Birnechta operatorlarni bir qatorga joylashtirish

Как правило, при написании программ операторы размещают на отдельной строке. Если операторы имеют небольшую длину, Visual Basic позволяет их поместить на одной строке, разделив двоеточием.

Например: $x = a + b : y = c - d : z = a * b$

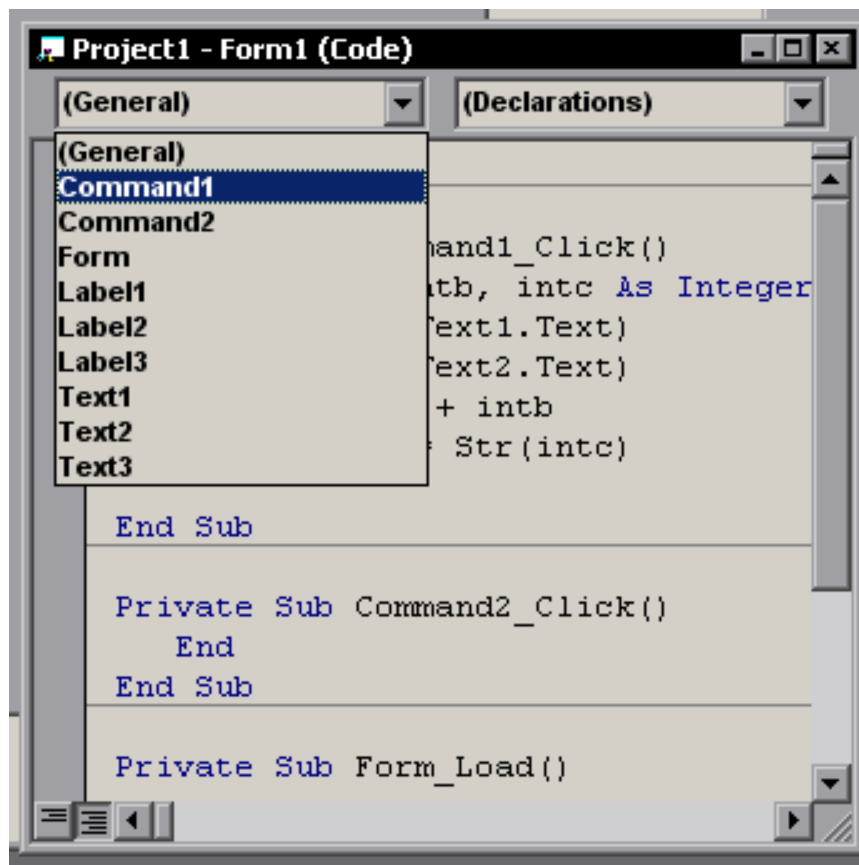
Kodni tagrirlash

Visual Basic muhitida dastur kodini kiritish uchun kod tahrirlagichidan foydalaniladi. Uni ishga tushirish uchun **Project Explorer** oynasidagi kodini kiritiladigan formaga yoki ob'yektga kursorni o'rnatish kerak va quyidagi amaldan biri bajariladi:

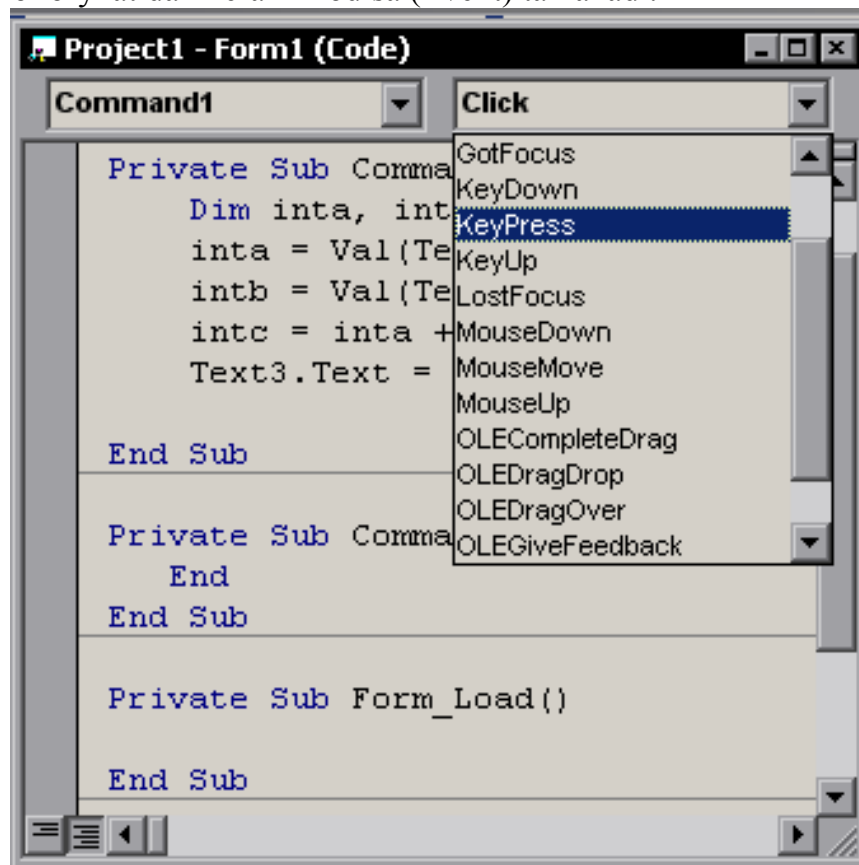
- **View** (Ko'rinish) menyusidagi **Code** (Kod) buyrug'ini tanlash;
- tanlab olingan ob'yekt ustida sichqonchani o'ng tugmasini bosganda hosil bo'lgan kontekstli menyudan **View Code** buyrug'ini tanlash;
- ob'yektning ustida sichqonchani chap tugmasini ikki marta bosish.

Bu amallarning bittasi bajarilganda dastur kodi kiritiladigan tahrirlagich oynasi ochiladi. Visual Basic ning har bir moduli uchun ichki seksiyalarga ajratilgan alohida kod oynasi yaratiladi. Kod tahrirlagich oynasidan selsiya tanlash **Object** ro'yxati yordamida amalgam oshiriladi.

Protsedura yaratish uchun ob'yekt tanlash: tahrirlovchi oynadagi **Object** (Ob'yekt) ro'yxatidan ob'yektlarning bittasi tanlanadi.



Procedure ro'yxatidan kerakli hodisa (Event) tanlanadi.



Aylananing radiusini ciqarish. Aylana uzunligini, doira yuzini, aylanaga ichki va tashqi chizilgan kvadratning yuzasini va perimetrini aniqlash.

Dastur kodi

Option Explicit

Private Sub Command1_Click()

Const Pi As Single = 3.1415

Dim sngR As Single

Dim sngD, sngAV, sngAO As Single

Dim sngC1, sngC2 As Single

Dim sngKV1, sngKV2 As Single

Dim sngKO1, sngKO2 As Single

sngR = Val(Text1.Text) ' Radiusni aniqlash

sngC1 = 2 * Pi * sngR ' Aylana uzunligi

sngC2 = Pi * sngR ^ 2 ' Doira maydoni

sngD = 2 * sngR ' Diametr

sngAV = sngD / Sqr(2) ' Ichki chizilgan kvadratning tomoni

sngKV1 = 4 * sngAV ' Ichki chizilgan kvadratning perimetri

sngKV2 = sngAV ^ 2 ' Ichki chizilgan kvadratning maydoni

sngAO = sngD * Sqr(2) ' Tashqi chizilgan kvadratning tomoni

sngKO1 = 4 * sngAO ' Tashqi chizilgan kvadratning perimetri

sngKO2 = sngAO ^ 2 ' Tashqi chizilgan kvadratning maydoni

Text2.Text = Str(sngC1)

Text3.Text = Str(sngC2)

Text4.Text = Str(sngKV1)

Text5.Text = Str(sngKV2)

Text6.Text = Str(sngKO1)

Text7.Text = Str(sngKO2)

```

End Sub
Private Sub Command2_Click()
End
End Sub

```

If, Go To, Select boshqaruv strukturalari

If va Select boshqaruv strukturalari **boshqaruv operatorlari** yoki **echim qabul qiluvchi konstruktsiyalar** deb ataladi. Shu bois ular dastur operatorlarining odatdagi ketma-ket bajarilishini o'zgartiradi. Bu konstruktsiyalar ishlatilmaganda dastur birinchi operatoridan oxirgi operatorgacha ketma-ket bajariladi. Echim qabul qiluvchi operatorlarini qo'llgash dasturda hosil bo'lgan shartlarga boi'lik holda aniqlangan amallarni bajarishga imkon beradi.

Misol: Tarmoqlanuvchi dastur

To'rtta A, B, C va D haqiqiy sonlarni kiriting. Maksimum va minimumlarini toping.

Algoritm:

1. A, B, C, D larni kiritish.
2. A va B qiymatlarining kattasini aniqlash va uni R1 o'zgaruvchisiga o'zlashtirish.
3. C va D qiymatlarining kattasini aniqlash va uni R2 o'zgaruvchisiga o'zlashtirish.
4. R1 va R2 qiymatlarining kattasini aniqlash va uni MAX o'zgaruvchisiga o'zlashtirish.
5. A va B qiymatlarining kichigini aniqlash va uni R1 o'zgaruvchisiga o'zlashtirish.
6. C va D qiymatlarining kichigini aniqlash va uni R2 o'zgaruvchisiga o'zlashtirish.
7. R1 va R2 qiymatlarining kichigini aniqlash va uni MIN o'zgaruvchisiga o'zlashtirish.
8. MAX va MIN o'zgaruvchilarining qiymatlarini chop etish

Dastur kodi

```
Option Explicit
Private Sub Command1_Click()
    Dim A, B, C, D As Single
    Dim R1, R2 As Single
    Dim MAX, MIN As Single
    A = Val(Text1.Text)
    B = Val(Text2.Text)
    C = Val(Text3.Text)
    D = Val(Text4.Text)

    If A > B Then
        R1 = A
    Else
        R1 = B
    End If
    If C > D Then
        R2 = C
    Else
        R2 = D
    End If
    If R1 > R2 Then
        MAX = R1
    Else
        MAX = R2
    End If
    If A < B Then
        R1 = A
    Else
        R1 = B
    End If
    If C < D Then
        R2 = C
    Else
        R2 = D
    End If
    If R1 < R2 Then
        MIN = R1
    Else
        MIN = R2
    End If

    Text5.Text = Str(MAX)
    Text6.Text = Str(MIN)
End Sub

Private Sub Command2_Click()
    End
```

End Sub

3.9. Select Case shartli strukturasi

Select Case shartli strukturasi bir nech shartlarni qayta ishlash imkonini beradi va **If.. .Then..** **.Else** konstruktsiya blokiga o'xshashdir. Bu struktura taxlil qilinuvchi ifodadan va **Case** operatorlari to'plamidan iborat.

Strukturasi:

Select Case O'zgaruvchi

CASE qiymat1

konstruktsiya1

CASE qiymat 2

konstruktsiya2

...

CASE qiymat K

konstruktsiyaK

CASE Else

alternativ konstruktsiya2

End Select

Misol: С помощью датчика случайных чисел имитировать бросок игральной кости. Вывести полученное значение текстом.

Dastur kodi

Option Explicit

Private Sub Command1_Click()

Dim R As Integer

Dim S As String

R = Int(6 * Rnd) + 1

Select Case R

Case 1

S = "BIR"

Case 2

S = "IKKI"

Case 3

S = "UCH"

Case 4

S = "TO'RT"

Case 5

S = "BESH"

Case Else

S = "OLTI"

End Select

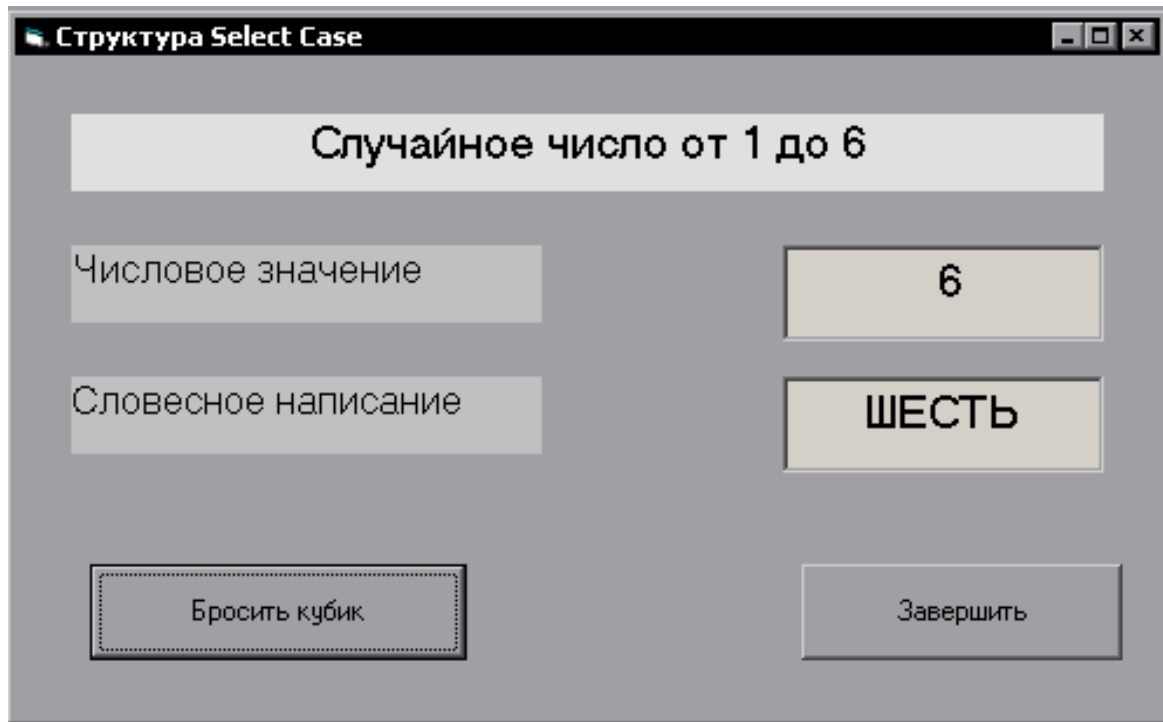
Text1.Text = Str(R)

Text2.Text = S

End Sub

```
Private Sub Command2_Click()
    End
End Sub
```

```
Private Sub Form_Load()
    Randomize
End Sub
```



Select Case strukturasida solishtirish amallarini ishlatish mumkin. Buning uchun **Is** yoki **To** xizmatchi soʻzini ifodaga qoʻshish kerak. **Is** xizmatchi soʻzi kompilyatorga tekshirilayotgan oʻzgaruvchining qiymatini ifodaning qiymati bilan solishtirishga koʻrsatmalar beradi. **To** xizmatchi soʻzi qiymatlar diapazonini beradi. Mayli yuqoridagi misolda tasodifiy sonlar 1 dan 10 gacha boʻlgan intervaldan olinsin. Dastur kodini oʻzgartiringiz va **Select Case** strukturasining ishlashini nazorat qiling.

Qayta ishlangan dastur kodi

Option Explicit

```
Private Sub Command1_Click()
    Dim R As Integer
    Dim S As String
    R = Int(10 * Rnd) + 1
    Select Case R
        Case Is < 4
            S = "KAM"
        Case 4 To 6
            S = "KO'P"
        Case Else
            S = "BO'LISHI MUMKIN EMAS"
    End Select
    Text1.Text = Str(R)
    Text2.Text = S
End Sub
```

End Sub

Private Sub Command2_Click()

End

End Sub

Private Sub Form_Load()

Randomize

End Sub

Siklli strukturalar

Dasturlash tillarida takrorlanuvchi jarayonlarni bajarish uchun siklli strukturalar ishlatiladi. Siklli strukturalarning uch xili mavjud:

1. Takrorlanishlar soni berilgan sikl (hisoblagichli sikllar)
2. Berilgan shart bajarilguncha bajariladigan sikl. Bunda shart sikl tanasi bajarilishidan oldin tekshiriladi (Yuqoridagi qator bilan boshqariladigan sikl)
3. Berilgan shart bajarilguncha bajariladigan sikl. Bunda shart sikl tanasi bajarilgandan so'ng tekshiriladi (Pastdagi qator bilan boshqariladigan sikl)

Sikl tanasi bo'yicha bir marta o'tish "iteratsiya" deyiladi.

Takrorlanishlar soni berilgan sikl (hisoblagichli sikl) For ... Next

For...Next strukturasini buyruqlar ketma-ketligini bir necha marta bajaradi. Bunaqangi konstrusiyani sikl deb ataladi va u yordamida bajariladigan dastur kodlarini – sikl tanasi deb ataladi..

For.. .Next strukturasining sintaksisi:

For Hisoblagich = Boshlang'ich_qiymat To Oxirg'i_qiymat Step Qadam

konstruksiya

Next Hisoblagich

Strukturaning birinchi argumenti - **Hisoblagich** siklning bajarilishlari sonini hisoblaydigan o'zgaruvchi nomini aniqlaydi. **Boshlang'ich_qiymat** parametri siklga birinchi marta kirishdan oldin **Hisoblagich** ga o'zlashtiriladigan boshlang'ich sonli qiymat bo'ladi. Sikl **Hisoblagich** ning qiymati **To** xizmatchi so'zidan keyingi **Oxirg'i_qiymat** ning qiymatidan oshib ketguncha bajariladi. Siklning har bir bajarilishidan so'ng **Hisoblagich** ning qiymati **Qadam** qiymatiga oshirilib boriladi. **Next** xizmatchi so'zi siklning oxirini ko'rsatadi. Siklning har bir bajarilishida **Hisoblagich** ning qiymati bilan **Oxirg'i_qiymat** argumentining qiymati solishtiriladi. Agar hisoblagichning qiymati **Oxirg'i_qiymat** ning o'rnatilgan qiymatidan oshib ketmasa, sikl tanasining konstruksiyasi bajariladi. Aks holda boshqarish **Next** dan keyingi operatorga o'tadi. **Hisoblagich** ning qiymatini o'zgartiruvchi **Qadam** ning qiymati manfiy ham bo'lishi mumkin. Bunday holatda sikl **Hisoblagich** ning qiymati **Oxirg'i_qiymat** ning qiymatidan kichik bo'lguncha bajariladi va hisoblagichning boshlang'ich qiymati oxirgi qiymatidan katta bo'lishi lozim. **Step** xizmatchi so'zini ishlatmasa ham bo'ladi. Bunday holatda esa qadamning qiymati avtomat ravishda 1 ga teng bo'ladi. **Hisoblagich**, **Boshlang'ich_qiymat**, **Oxirg'i_qiymat**, **Qadam** parametrlari – o'zgaruvchi, doimiy yoki butun turdagi ifoda bo'lishi mumkin.

Musbat qadamli

For ... Next strukturasini

Manfiy qadamli

For ... Next strukturasini

Agar **Bosh_qiym** ning qiymati **Oxir_qiym** qiymatidan katta (musbat **qadam** da) yoki **Bosh_qiym** ning qiymati **Oxir_qiym** qiymatidan kichik (manfiy **qadam** da) bo'lsa, sikl tanasi bir marta ham bajarilmaydi.

Agar **Hisoblagich** ning qiymati sikl ichidagi shart yakunlanish holati bajarilmaydigan qilib o'zgartirilsa (masalan, iteratsiya oxirida sikl **Hisoblagich**ga **Bosh_qiym** qiymati o'zlashtiriladi), unda "sikllashuv" hosil bo'ladi, ya'ni sikl cheksiz bajariladi.

12-misol: For ... Next strukturasini ishlatish

Arifmetik progressiyaning dastlabki 100 ta elementinining yig'indisini hisoblash.

Arifmetik progressiya o'zining a_0 nollik elementi va d ayirmasi bilan beriladi. Arifmetik progressiyaning har bir kelasi elementi formulasi $a_k = a_{k-1} + d$ bilan hisoblanadi.

Arifmetik progressiyaning dastlabki 100 ta elementinining yig'indisi quyidagicha:

$S = \sum a_k$, bu erda k 0 dan 100 gacha bo'lgan qiymatlarni ketma-ket qabul qiladi.

Algoritmi

1. a_0 va d larni kiritingiz
2. Arifmetik progressiyaning yig'indisi - S o'zgaruvchisiga a_0 ning qiymatini o'zlashtiringiz
3. For $K = 1$ To 100 ... Next siklida har bir iteratsiyada arifmetik progressiyaning har bir elementini va yig'indining qiymatini quyidagi formula yordamida hisoblanadi

$$a_k = a_{k-1} + d \quad S = S + a_k$$

4. S ni chiqarish

Dastur kodi

Option Explicit

```
Private Sub Command1_Click()  
    Dim A0, D, S As Double  
    Dim K As Integer  
    A0 = Val(Text1.Text)  
    D = Val(Text2.Text)
```

```
    S = A0
```

```
    For K = 1 To 100
```

```
        A0 = A0 + D
```

```
        S = S + A0
```

```
    Next
```

```
    Text3.Text = Str(S)
```

```
End Sub
```

```
Private Sub Command2_Click()
    End
End Sub
```

Sharti oldinda va oxirida berilgan sikllar

Bazi bir holatlarda sikl tanasining necha marta bajarilishini oldindan aniqlab bo'lmaydi. Bunday holatlarda sikldagi shartning qiymati True bo'lguncha bajariladi.

Bunaqangi sikllarning ikki holati mavjud:

1. Sikl sharti iteratsiyaga har bir kirishidan oldin tekshiriladi (sharti oldinda berilgan sikllar)
2. Sikl sharti har bir iteratsiyaning bajarilishidan so'ng tekshiriladi (sharti oxirida berilgan sikllar)

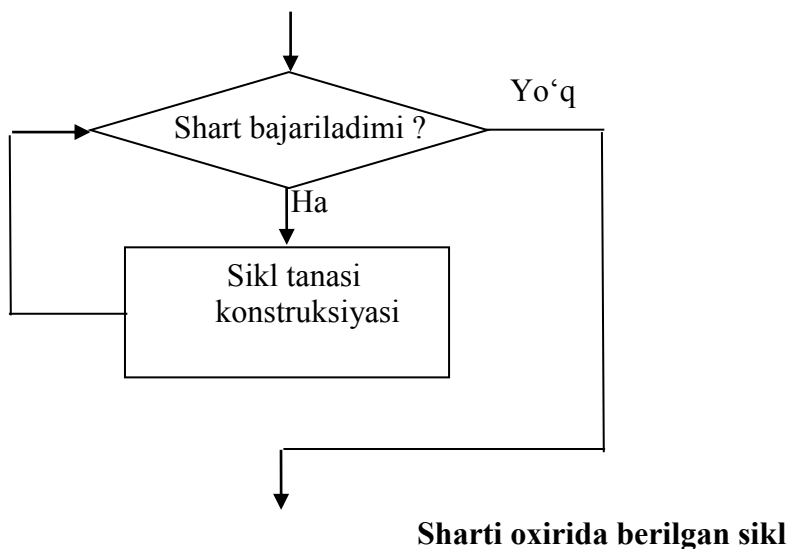
Sharti oldinda berilgan Do While siklining sintaksisi

Do While Shart

Konstruksiya

Loop

Do While ... Loop konstruksiyasida berilgan sikl operatori undagi berilgan shartning qiymati **True** bo'lguncha davom etadi. **Shart** argumenti mantiqiy ifoda bo'lib, siklga har bir kirishda tekshiriladi. Agar uning qiymati **True** bo'lsa, **Do While** va **Loop** xizmatchi so'zlari orasidagi buyruqlar ketma-ketligi bajariladi. Bu konstruksiya sikl tanasini tashkil qiladi. Agar siklga kirishda shartning qiymati **False** bo'lsa, sikldan chiqadi va boshqarish **Loop** dan keyingi konstruksiyaga beriladi. Shunaqangi holat mavjudki, ya'ni sikl ichidagi operator bir marta ham bajarilmasligi mumkin. Bu holat sikl sharti birinchi marta tekshirilganda uning qiymati **False** bo'lsa.

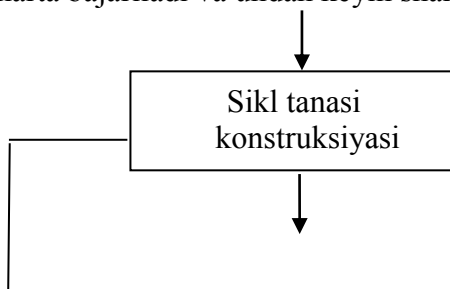


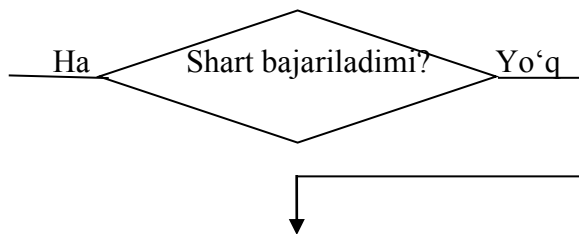
Do

Konstruksiyalar

Loop While Shart

Bu sikl strukturasini ishlatishda shart sikl oxirida tekshiriladi. Bu holatda sikl tanasi kamida bir marta bajariladi va undan keyin shart tekshiriladi.

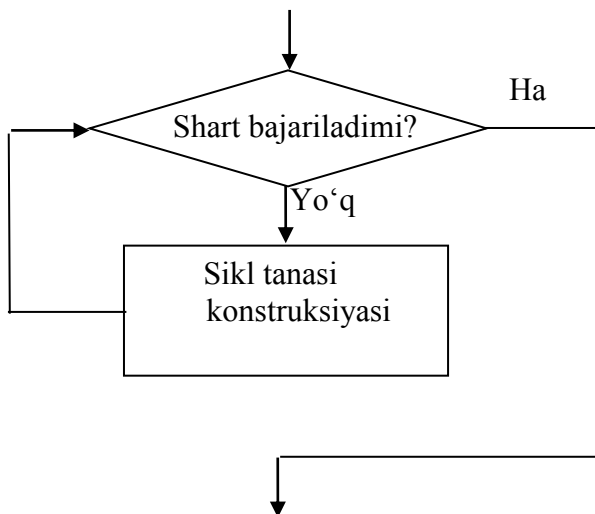




Do . . Loop sikl konstruksiyasining yana bir ko'rinishi bor. Ular avvalgi ko'rilgan sikllar bilan bir xil, lekin sikl shartining qiymati False bo'lguncha bajariladi.

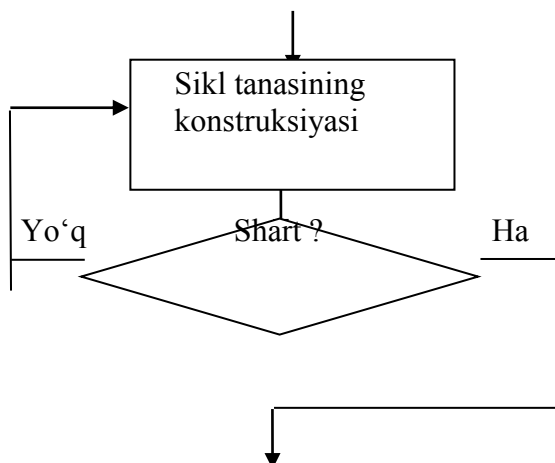
Sharti oldinda berilgan sikl

Do Until *Shart*
konstruksiya
Loop



Sharti oxirida berilgan sikl

Do
Konstruksiya
Loop Until *Shart*



Sikldan shartsiz chiqib ketish sharti oldinda va oxirida berilgan sikllarning strukturasi uchun **Exit Do** sintaksisiga ega. **Exit** buyrug'i siklning bajarilishini yakunlaydi va boshqarishni sikldan keyinga konstruksiyaga beradi.

14-misol: Shartli sikl strukturasini ishlatish (shart sikl tanasi bajarilishidan oldin tekshiriladi)

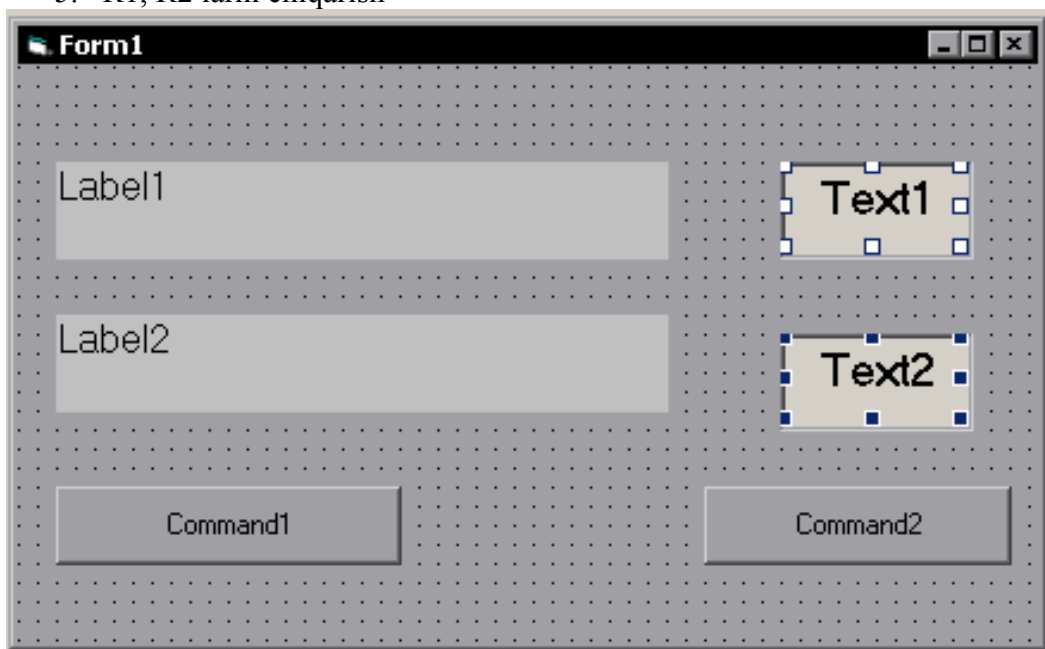
Kvadrat berilgan. Kvadrat tomoni 2 ga teng. Kvadrat markazi dekart koordinatalari sistemasining (0,0) nuqtasida joylasgan. Siklda kvadrat ichidan tasodifiy nuqta olinadi. Nuqta markazlari (0,0) nuqtada va radiuslari $R_1=0,5$; $R_2=1,0$ bo'lgan ikkita konsentrik aylanadan tashkil topgan halqaga tegishlilikini tekshirish kerak. Quyidagi shartlarning biri bajarilsa, siklni to'xtatiladi.

- Tasodifiy olingan nuqtalarning soni 300 ga teng bo'lsa.
- Halqaga tegishli bo'lgan nuqtalar soni 50 ga teng bo'lsa.

Nuqtalarning umumiy soni va halqaga tegishli bo'lgan nuqtalarning soni chop etiladi.

Algoritmi

1. K1 – aniqlanadigan nuqtalarning tartibi. Siklga kirishdan oldin unga nol qiymatini berish lozim.
2. K2 – halqaga tushuvchi nuqtalar soni. Siklga kirishdan oldin unga nol qiymatini berish lozim.
3. **Sikl sarlavqasi.** $K1 \leq 300$ va $K2 \leq 50$ shartlarini tekshirish.
4. Agar teksirish natijasi True bo'lsa, $K1 = K1 + 1$ iteratsiyasini amalga oshirish
(x,y) nuxtasi koordinatalarini topish
 $-1.0 \leq x \leq +1.0$, $-1.0 \leq y \leq +1.0$
Nuqtaning $0.25 \leq x^2 + y^2 \leq 1.0$ halqasiga tegishlilikini tekshirish
Agar $K2 = K2 + 1$ sharti bajarilsa, 3 punktga o'tish
5. K1, K2 larni chiqarish

**Dastur kodi**

```
Option Explicit
Private Sub Command1_Click()
Dim K1, K2 As Integer
Dim X, Y, D As Single
K1 = 0
K2 = 0
Do While (K1 <= 300) And (K2 <= 50)
X = 2# * Rnd - 1#
Y = 2# * Rnd - 1#
K1 = K1 + 1
D = X * X + Y * Y
```

```
If (D > 0.25) And (D < 1#) Then K2 = K2 + 1
```

```
Loop
```

```
Text1.Text = Str(K1)
```

```
Text2.Text = Str(K2)
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
End
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Randomize
```

```
End Sub
```

Belgi kattaliklar. Satrlar bilan ishlash. Tip o'zgartiruvchi operatorlar.

Avvalgi mavzularda ko'rib o'tilganidek, formaga o'rnatilayotgan ob'ektlarning xar biri aloxida o'ziga xos xususiyatlarga ega. Ushbu ob'ektlarning aksariyat xususiyatlari aynan belgi (satr) kattaliklar orqali aniqlanadi va unga qiymat beriladi. Masalan Text ob'ekti orqali dastur davomida biror o'zgaruvchiga qiymat berish uchun uning text xususiyatidan foydalaniladi. Bu xususiyat esa satr kattalikdagi qiymatga ega. Agar biz biror son kiritmoqchi bo'lsak, ushbu ob'ektga kiritilgan belgini songa aylantirib olishimizga to'g'ri keladi. YOki aksincha, dastur davomida olingan natijani ekranga chop etish uchun Label ob'ektidan foydalaniladi. Bu ob'ektda natijani chop etish Caption xususiyati bilan aniqlanadi va u belgi kattalikka ega. Demak xosil qilingan sonli qiymatni chop etishdan oldin uni belgi (satr) kattalikka aylantirib olishga to'g'ri keladi.

Ushbu muammolardan chiqish uchun barcha dasturlash tillari kabi Visual Basic da xam bir qator belgi kattaliklar bilan ishlash funksiyalari mavjud. Ushbu funksiyalar bilan quyida misollar yordamida tanishib chiqamiz.

Belgi kattalikni sonli kattalikka o'tkazish uchun Val() funksiyasidan foydalanamiz. Funksiyaning yozilish formati:

$\langle s_o'zgaruvchi \rangle := Val(\langle a_o'zgaruvchi \rangle)$

bu yerda $\langle a_o'zgaruvchi \rangle$ dagi belgi kattalikni Val funksiyasi sonli kattalikka o'giradi va natijani $\langle s_o'zgaruvchi \rangle$ ga uzatadi. Val funksiyasiga teskari funksiya xam mavjud bo'lib, u Str(). YOzilish formati quyidagicha:

$\langle a_o'zgaruvchi \rangle := Str(\langle s_o'zgaruvchi \rangle)$

bu yerda $\langle s_o'zgaruvchi \rangle$ dagi sonli qiymatni Str funksiyasi belgi kattalikka o'tkazadi va natijani $\langle a_o'zgaruvchi \rangle$ ga o'zlashtiradi.

Masalan, a va s o'zgaruvchilarga quyidagicha qiymat bersak: a= «10», s= «15». Agar va a va s o'zgaruvchilarning yig'indisini v=a+s ko'rinishda bersak, u xolda natija «1015» ga teng bo'ladi. CHunki bu yerda a va s lar belgi kattalikdagi qiymatlarga ega. Ularning sonli qiymatdagi yig'indisini olmoqchi bo'lsak, u xolda quyidagi amalni bajarishimizga to'g'ri keladi:

$v = val(a) + val(c)$

endi esa v o'zgaruvchining qiymati 25 ga teng bo'ladi. YA`ni Val funksiyasi yordamida a va s o'zgaruvchilar sonli kattalikka aylantirildi va ularning yig'indisi xisoblanib, v o'zgaruvchiga uzatildi.

Satr kattaliklar bilan ishlash uchun bir nechta funksiyalar mavjud. Ulardan biri satr boshida yoki oxirida mavjud bo'lgan probel (bo'sh joy)larni olib tashlash funksiyasi. Bu funksiyalar quyidagilar:

LTrim – satrning boshlanishida joylashgan probellarni olib tashlaydi;

RTrim – satr oxirida joylashgan probellarni olib tashlaydi;

Trim – satr boshida va oxirida joylashgan probellarni olib tashlaydi.

Misol uchun:

st=" Probellarni o'chirish "

print Ltrim(st) ' natija "Probellarni o'chirish "

print Rtrim(st) ' natija " Probellarni o'chirish"

print Trim(st) ' natija "Probellarni o'chirish"

Berilgan satr ichidan kerakli qismini qirqib olish uchun quyidagi funksiyalardan foydalanamiz:

Left (st,i) – st satrining chap tomonidan i ta belgini qirqib oladi;

Right (st,i) – st satrining o'ng tomonidan i ta belgini qirqib oladi;

Mid (st,i,j) – st satrining i-belgisidan boshlab j ta belgini qirqib oladi.

st="Salom mening do'stim"

Print Left(st,5) ' natija "Salom"

Print Right(st,6) ' natija "do'stim"

Print Mid(st,7,6) ' natija "mening"

Eslatma! Xar bir probel (bo'sh joy) xam bitta belgi sifatida qabul qilinadi.

Ba'zan klaviaturada mavjud bo'lmagan belgilarni xosil qilishga to'g'ri kelib qoladi. Bunday xolatlarda ushbu belgilarning kodlaridan foydalanishga to'g'ri keladi. Koddan belgiga aylantirishni Chr() funksiyasi amalga oshiradi. Masalan Enter tugmasining ASCII kodi 13 ga teng. Enter ning vazifasidan dasturlashda foydalanish uchun Chr(13) deb yozish kifoya.

Masalan, 1 dan 5 gacha bo'lgan sonlarni ma'lumot oynasida ustun shaklida xosil qilish uchun quyidagi dastur qismini yozish kifoya:

For i=1 to 5

s=s+ chr(13)+ str(i)

next i

Msg s

YUqoridagi operatorlardan tashqari yana qo'shimcha ravishda quyidagi funksiyalardan xam foydalanish mumkin:

Ucase(st) – st satridagi barcha kichik xarflarni bosh xarflarga o'tkazish;

Lcase(st) – st satridagi barcha bosh xarflarni kichik xarflarga o'tkazish.

Masalan,

st="salom": se="YANGI"

Print Ucase(st) ' natija "SALOM"

Print Lcase(se) ' natija "yangi"

StrConv(st) funksiyasi esa st satridagi xar bir so'zning birinchi xarfini bosh xarfga qolganlarini esa kichik xarflarga o'tkazish imkonini beradi:

st="aliev atxam solievich"

```
Print StrConv(st)    'natija "Aliev Atxam Solievich"
YOKi
st="ALIEV ATXAM SOLIEVICH"
Print StrConv(st)    'natija "Aliev Atxam Solievich"
```

InStr(st,s) funksiyasi s satrni st satr ichidan izlab topadi va s satr st satrining nechani belgisidan boshlanishini aniqlaydi:

```
st= "Salom olam"
```

```
s= "olam"
```

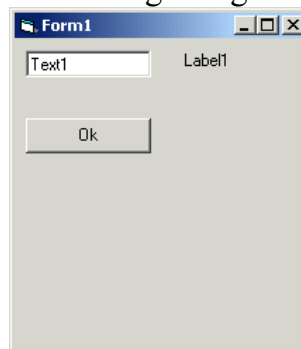
```
Print InStr(st,s)    'natija 7
```

CHunki «olam» so'zi «Salom olam» so'zining 7-belgisidan boshlanadi.

Satr kattaliklarga misol ko'rib o'taylik.

Berilgan s satrdagi barcha belgilarni ASCII kodini chop etuvchi dastur tuzish talab etilsin.

Echish: avval berilgan satrdagi xar bir belgi qirqib olinadi va ularning xar birini ASCII kodi chop etiladi. Dastur matnini yozishdan avval uning forma ko'rinishini yaratib olish kerak. Formaga matn kiritish uchun bitta Text, natija chiqishi uchun bitta Label va dasturdan natija olish uchun bitta Command ob'ektlarini o'rnatamiz (9-rasm). Properties oynasidan Label ning AutoSize xususiyatiga True, Caption xususiyatiga esa bo'sh joy o'rnatiladi. Command ob'ektining caption xususiyati esa Ok ga o'zgartiriladi.



9-rasm. Dastur formasining ko'rinishi.

Endi dastur matnini yozish uchun Ok tugmasi ustida sichqoncha ikki marta tez bosiladi va xosil bo'lgan kodlar oynasida quyidagi o'zgartirish kiritiladi:

```
Private Sub Command1_Click()
```

```
    S=Text1.Text    'matn s uzgaruvchiga olinadi
```

```
    l=Len(s)        'matn uzunligi aniklanadi
```

```
    For i=1 To l    'sikl matn uzunligigacha ochiladi
```

```
        s1=Mid(s, i, 1)    'bitta belgi kirkib olinadi
```

```
        Label1.Caption=Label1.Caption + Chr(13)+ s1 + " - " + Str(Asc(s1))
```

```
    'kirkib olingan belgining kodi chop etiladi
```

```
    Next i        'siklning navbatdagi kiymati
```

```
End Sub
```

Massivlarni elon qilish. Bir va ko'p o'lchovli massivlar.

Oddiy o'zgaruvchilardan boshqa miqdorlarni saqlash uchun massivlardan foydalaniladi. Massiv bir nomdagi, lekin turli indeksli o'zgaruvchilar to'plamidan iboratdir. Xar bir shunday o'zgaruvchilar massiv elementlari deyiladi. Massiv elementlarining miqdori

massiv o'lchami deyiladi. Massivning o'lchami EXMning operativ xotirasi va massiv elementlarining tipi bilan chegaralangan.

Massivning barcha elementlari bir xil tipga tegishli bo'ladi. Biroq, agar massiv Variant tipiga tegishli bo'lsa, u xolda massivning aloxida elementlari aloxida tipdagi ma'lumotlarni saqlashi mumkin. YA'ni, bir element sonli, boshqasi satrli bo'lishi mumkin.

Odatda massivning yozilish formati quyidagicha bo'ladi:

<massiv nomi>(<massiv o'lchami>) As <massiv elementlari tipi>

masalan, A(10) As Integer

bu yerda A-massiv nomi, 10-massiv o'lchami, ya'ni massiv elementlari A(0), A(1),...,A(10) lardan iborat, Integer – massivning xar bir elementining tipi.

YUqorida aytib o'tilganidek xar bir massivning elementi o'zida bir xil tipdagi aloxida-aloxida qiymatlarni saqlaydi.

Visual Basic da massivlar ikki turi mavjud:

1. Fiksirlangan
2. Dinamik

Fiksirlangan massiv – o'lchovi o'zgarmaydigan massiv, ya'ni uni e'lon qilishda o'lchami aniq qilib belgilanadi.

Dinamik massivning o'lchamini esa dastur ishlashi davomida o'zgartirish mumkin.

Dasturlashda massivdan foydalanish uchun avval uni e'lon qilib olish kerak. Fiksirlangan massivni e'lon qilish uning ko'rinish soxasiga bog'liq ravishda quyidagicha amalga oshirilishi mumkin:

- global massiv Public operatori yordamida modulning Declaration seksiyasida e'lon qilinadi;
- modul darajasidagi massiv Private operatori yordamida modulning Declaration seksiyasida e'lon qilinadi.
- lokal massiv Private operatori yordamida e'lon qilinadi.

Massiv e'lon qilinayotgan paytda uning nomidan keyin qavs ichida massivning yuqori chegarasi ko'rsatiladi. Odatda massivning quyi chegarasi 0 ga teng. Masalan, keltirilgan misolda, ya'ni modulning Declaration seksiyasida kiritiluvchi massiv 21 ta elementdan tashkil topgan. Massiv elementi indeksi 0 dan 20 gacha o'zgaradi:

Dim A(20) As Integer

Xuddi shunday o'lchamdagi massivni global e'lon qilish uchun quyidagi koddan foydalaniladi:

Public A(20) As Integer

Massivning quyi chegarasini xam aniq qilib belgilab qo'yish mumkin. Buning uchun To xizmatchi so'zidan foydalaniladi:

Dim A(1 To 20) As Integer

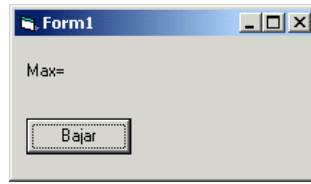
Bunday xolatda massiv 20 ta elementdan tashkil topadi va massiv elementlarining indeksi 1 dan 20 gacha o'zgaradi.

Ma'lumki, massivlar bir va ko'p o'lchovli bo'ladi. YUqorida ko'rib o'tilganlarning barchasi bir o'lchovli massivlarga misol bo'ladi. Bir o'lchovli massivlar ustida amallar bajarishga doir kichik bir misol ko'rib o'taylik:

Masala: Berilgan A(20) butun tipdagi massiv elementlari ichidan eng kattasini topish dasturini tuzing.

Echish: Berilgan massiv elementlari ichidan eng kattasini topish uchun, avval massiv elementlari kiritib olinadi. So'ng massivning birinchi elementini eng katta deb qabul qilinadi va shu elementga nisbatan qolgan elementlar birma-bir solishtirib chiqiladi.

Dasturning forma ko'rinishi sodda xolda yaratib olinadi, ya'ni formaga bitta Label (natijani chiqarish uchun) va bitta command (natija olish uchun) ob'ektlarini o'rnatamiz (10-rasm).



10-rasm. Dasturning forma ko'rinishi

Properties oynasini mos ravishda quyidagicha o'zgartiriladi:

Label	Caption	Max=
	Autosize	True
Command	Caption	Bajar

Dastur kodini yozish uchun «Bajar» tugmasi ustida sichqonchanning o'ng tugmasi ikki marta bosiladi va xosil bo'lgan kodlar oynasidagi prosedura ichiga quyidagilar kiritiladi:

```
Private Sub Command1_Click()
    Dim a(1 To 20) As Integer
    For i=1 To 20
        a(i)=Val(InputBox("A(" + Str(i) + ")="))
    Next i
    Max=a(1)
    For i=2 To 20
        If Max < a(i) Then Max=a(i)
    Next i
    Label1.Caption=Label1.Caption + Str(Max)
End Sub
```

Ikki o'lchovli massivlar xam bir o'lchovli massiv bilan o'xshash bo'ladi. Faqat ikki o'lchovli massivda qavs ichida ko'rsatiluvchi o'lcham ikkita, o'zaro vergul bilan ajratiluvchi miqdorlardan tashkil topadi:

A(10,15) As Real

Bu yerda A-massiv nomi, «10,15»-massivning 1- va 2- o'lchovlari miqdori, Real – massiv elementlari tipi.

Ikki o'lchovli massivni oddiy ustun va satrlardan iborat bo'lgan jadval deb olish mumkin. Jadvalning xar bir yacheykasida joylashgan qiymat massiv elementi qiymati xisoblanadi. Masalan, V(N,M) massiv quyidagicha ifodalanishi mumkin:

	1	2	3	...	M
1	B(1,1)	B(1,2)	B(1,3)	...	B(1,M)
2	B(2,1)	B(2,2)	B(2,3)	...	B(2,M)
...	...	...	...	...	...
N	B(N,1)	B(N,2)	B(N,3)	...	B(N,M)

YA'ni, V jadval M ta ustun va N ta satrdan iborat bo'lib, jami MxN ta yacheykadan iborat.

YUqoridagi masalani ikki o'lchovli massiv uchun o'zgartirsak, forma o'zgarishsiz saqlanadi, dastur kodida quyidagicha o'zgartirish qilinadi:

```
Private Sub Command1_Click()
    Dim a(1 To 20, 1 To 20) As Integer
    For i=1 To 20
        For j = 1 To 20
            a(i, j)=Val(InputBox("A(" + Str(i)+"," + Str(j)+ ")="))
        
```

```

Next j, i
Max=a(1,1)
For i=1 To 20
For j = 2 To 20
If Max < a(i,j) Then Max=a(i,j)
Next j, i
Label1.Caption=Label1.Caption + Str(Max)
End Sub

```

Ko'rinib turibdiki fiksirlangan massiv elementlarining miqdori avvaldan aniq chegaralab qo'yilyapti. Bu massiv elementlari miqdori avvaldan ma'lum bo'lgan masalalar uchun qo'l keladi. Lekin, massiv elementlari miqdori avvaldan ma'lum bo'lmasdan, dastur ishlashi davomida aniqlanuvchi xolatlarda anchagina muammolar yuzaga kelishi mumkin. Chunki keragidan ortiq miqdorni ko'rsatish operativ xotiradan ortiqcha joyni band qilib qo'yadi.

Visual Basic da dinamik massiv tushunchasi mavjud bo'lib, uning elementlari miqdori avvaldan chegaralanmaydi. Dinamik massivlarni e'lon qilish quyidagicha amalga oshiriladi:

1. Dinamik massiv xam fiksirlangan massivni e'lon qilish uchun foydalanilgan operatorlar yordamida amalga oshiriladi, global massiv Public yordamida, modul darajasidagi massiv Dim yordamida, massivni prosedurada e'lon qilish Dim yoki Static yordamida. Lekin massiv elementlari miqdori ko'rsatilmasdan, bo'sh qavs yoziladi:

```
Dim C() As Integer
```

2. ReDim bajaruvchi operator yordamida massiv o'lchami son yoki ifoda ko'rinishida beriladi. Massiv o'lchamlarini berish quyidagi usullardan biri ko'rinishida berilishi mumkin:

```
ReDim C(x)
```

```
ReDim C(20)
```

```
ReDim C(1 To 20)
```

ReDim operatori bilan massiv o'lchami berilganda yoki o'zgartirilganda joriy massivning oldingi qiymatlari yo'qotiladi. Agar massiv yangitdan e'lon qilinishi kerak bo'lganda bu usul qulay xisoblanadi. Agar massivning avvalgi qiymatlarini saqlagan xolda massiv o'lchami o'zgartirilishi kerak bo'lsa, u xolda ReDim operatori Preserve kalit so'zi bilan qo'llaniladi. Masalan, quyidagi kod massiv elementlari miqdorini avvalgi qiymatlarni saqlagan xolda bittaga orttiradi:

```
ReDim Preserve C(x+1)
```

Mavzu. Visual Basic da prosedura va funksiyalar.

Proseduralarning Parametrli va parametrsiz proseduralar. Proseduraga murojaat qilish va undan foydalanish.

Dastur tuzish jarayonida, uning turli joylarida ma'nosiga ko'ra bir xil, mustaqil xarakterga ega bo'lgan va yechilayotgan asosiy masalaning biror qismini hal qilishni o'z bo'yniga olgan murakkab algoritmdan bir necha marotaba foydalanishga to'g'ri keladi. Masalan, matrisalarni ko'paytirish, matrisani vektorga ko'paytirish, chiziqsiz tenglamani yechish, chiziqli algebraik tenglamalar sistemasini yechish, faktorial hisoblash, yig'indi hisoblash va hokazo kabi masalalarni hal qilish algoritmlari juda ham ko'p masalalarni yechishning bosh algoritmlarida qayta-qayta, turli boshlang'ich ma'lumotlar bilan qatnashishi mumkin. Bunday hollarda, malakali dasturchi dastur matnini ixchamlashtirish, dasturning ishonchlilik darajasini oshirish, dasturni tahrirlashni tezlashtirish va dasturning umumiylikligini

(universalligini) ta'minlash uchun prosedura va funksiyalardan kengroq foydalanib, mukammal dastur yaratishga harakat qiladi.

Prosedura va funksiyalar mustaqil dasturli ob'ektlar hisoblanadi. Bu mustaqil dasturli ob'ektni dasturchi o'z hoxishiga va undan olinadigan natijalariga ko'ra prosedura yoki funksiya ko'rinishida aniqlashi mumkin. Odatda olinadigan natija yagona qiymatli bo'lsa funksiyadan, olinadigan natijalar soni bir nechta bo'lsa proseduradan foydalanish maqsadga muvofiqdir.

Visual Basic da proseduralarning quyidagi ko'rinishlari mavjud:

- Sub
- Function
- Property

Sub prosedurasi.

Sub prosedurasi qiymat qaytarmaydi, bu prosedura ko'pincha unga bog'langan xodisalarni qayta ishlash uchun qo'llaniladi. Unga faqat standart modullar, forma va sinf modullarini o'rnatish mumkin. Sub prosedurasining yozilish formati quyidagicha:

```
[Private] [Public] [Static] Sub Prosedura_nomi (parametrlar)
    operatorlar
End Sub
```

Sub va End Sub kalit so'zlari oralig'ida ushbu proseduraga murojaat qilingan vaqtda bajarilishi lozim bo'lgan jarayonning dastur kodi yoziladi. Parametrlardan esa proseduraga uzatiluvchi o'zgaruvchilarni e'lon qilish uchun foydalaniladi.

Sub prosedurasi umumiy prosedura va xodisa proseduralariga bo'linadi.

Umumiy proseduralar xodisa proseduralar yoki boshqa bir prosedura tomonidan murojaat qilinganida takror-takror bajarilishi mumkin bo'lgan dasturning qismi deb qarashimiz mumkin. Uning yozilish formati quyidagicha:

```
[Private] [Public] [Static] Sub Prosedura_nomi (parametrlar)
    operatorlar
End Sub
```

Xodisa prosedurasi esa Visual Basic formasida joylashgan biror bir ob'ektga bog'liq bo'lib, shu ob'ektga nisbatan yuz beruvchi xodisa natijasida bajariladi. Xodisalar uchun Sub prosedurasining yozilish formati quyidagi ko'rinishga ega:

```
Private Sub Form_xodisaNomi (parametrlar)
    Operatorlar
End Sub
```

YOzilish formatidan ko'rinib turibdiki, ushbu xodisa prosedurasi formaga bog'langan. Masalan, prosedura forma ishga yuklanganida bajarilishi uchun Form_Load deb yozish kerak bo'ladi. Agar prosedura formaga joylashtirilgan boshqaruv elementiga nisbatan biror xodisa uchun yozilishi kerak bo'lsa, u xolda proseduraning yozilish formati quyidagicha o'zgaradi:

```
Private Sub boshqaruvElementiNomi_xodisaNomi (parametrlar)
    Operatorlar
End Sub
```

Xodisa proseduralari Visual Basic da avtomatik xosil qilinadi. YA'ni kodlar oynasining yuqori qismida ikkita ochiluvchi ro'yxatga o'xshash maydon mavjud bo'lib, chap tomonidagisi formaga o'rnatilgan boshqaruv elementlarining ro'yxatini o'zida saqlasa, o'ng tomonidagisi tanlangan boshqaruv elementiga xos bo'lgan xodisalar ro'yxatini o'zida saqlaydi (11-rasm).



11-rasm. Visual Basic da kodlar oynasi

Yuqorida aytib o'tganimizdek, prosedura hisoblab bergan natijalar uning faktik parametrlari orqali asosiy dasturga uzatiladi. Lekin, ayrim paytlarda prosedura parametrsiz ham bo'lishi mumkin. Bu holda asosiy dasturning barcha parametrlari prosedura parametrlari rolini bajaradi. Parametrsiz prosedurada ham proseduraning barcha bo'limlari saqlanib qoladi, faqat parametrlar ro'yhatigina qatnashmaydi.

Proseduralarni aniqlash va ulardan foydalanishni quyidagi misol yordamida ko'rib chiqaylik:

Misol: $u = \max(x + y, x * y)$, $v = \max(0.5, u)$ – berilgan x va y haqiqiy sonlardan foydalanib u va v qiymatlarni aniqlash.

bu yerda x, u - qiymatlari kiritiladigan haqiqiy tipli o'zgaruvchilar.

1. Masalani yechish dasturini proseduradan foydalanmay tuzilgan holi:

```
Dim x, y, u, v As Double
Dim a, b, s As Double
Private Sub Command1_Click()
  X=Val(Text1.Text)
  Y=Val(Text2.Text)
  A=x + y: b=x * y
  If a > b Then s=a Else s=b
  U=s
  A=0.5: b=u
  If a > b Then s=a Else s=b
  V=s
  MsgBox (Str(u) + " " + Str(v))
End Sub
```

Ahamiyat bersangiz, dasturdagi shartli operator ikki marta takrorlanib, bir xil ish bajardi.

2. Masalani yechish dasturini parametrsiz proseduradan foydalanib tuzilgan holi (endi yuqoridagi dasturda yo'l qo'yilgan kamchilikni proseduralar orqali tuzatishga harakat qilamiz):

```
Dim x, y, u, v As Double
Dim a, b, s As Double

Private Sub max1() ' max1 prosedurasi tashkil etildi
  If a > b Then s=a Else s=b
End Sub

Private Sub Command1_Click()
  X=Val(Text1.Text)
  y=Val(Text2.Text)
  a=x + y: b=x * y
  Call max1 ' max1 prosedurasiga murojaat
  uqs
```

```

a=0.5: b=u
Call max1 ' max1 prosedurasiga murojaat
vqs
MsgBox (Str(u)+ " " + Str(v))
End Sub

```

Asosiy dasturning operatorlar qismida ikki marta yozilgan max1 parametrsiz prosedurasiga murojaat, eʼlon qilingan prosedurani ikki marta asosiy dasturga olib kelib ishlatishni tashkil qiladi. Eʼtibor bergan boʻlsangiz proseduraga dasturning ixtiyoriy qismidan murojaat Call kalit soʻzi yoradmida amalgi oshirilmoqda. Ahamiyat berilsa, ikkinchi dastur birinchi prosedurasiz tuzilgan dasturga koʻra ixchamroq va soddaroqdir. Biz kiritgan prosedura hozircha faqat ikkita haqiqiy son ichidan kattasini aniqlab berdi holos, shuning uchun dastur matnining hajmini kamaytirishdan erishgan yutuq salmoqli boʻlmadi. Lekin, proseduralar asosan koʻp hajmli matndagi amallarni, vazifalarni bajarishga moʻljallanadi va bu holda erishilgan yutuq salmogʻi ancha yuqori boʻladi.

Parametrsiz proseduraning asosiy kamchiligi, uning asosiy dasturga va undagi maʼlum parametrlarga bogʻlanib qolganligidir.

Prosedura bilan asosiy dasturni bogʻlaydigan asosiy faktor bu – prosedura parametrlaridir. Parametrlarni ikkita tipga ajratiladi: qiymatli parametrlar (parametr-qiymat), oʻzgaruvchili parametrlar (parametr - oʻzgaruvchi).

Parametr - qiymat bu prosedurani ishlash jarayonini taʼminlovchi parametrlar hisoblanadi, yaʼni asosiy dastur qiymatlarini proseduraga uzatadigan parametrlardir.

Endi, yuqorida koʻrib chiqilgan sonlarni eng kattasini topish algoritmining dasturini qiymatli parametr bilan yozilgan proseduralar orqali amalga oshiraylik:

```

Dim x, y, u, v As Double
Dim s As Double

Sub max1(a, b As Double)
  If a > b Then s=a Else s=b
End Sub

Private Sub Command1_Click()
  X=Val(Text1.Text)
  y=Val(Text2.Text)
  Call max1(x + y, x * y)
  U=s
  Call max1(u, 0.5)
  V=s
  MsgBox (Str(u) + " " + Str(v))
End Sub

```

bu yerda a, b - proseduraning qiymatli formal parametrlari.

Proseduraga murojat qilishda formal va faktik parametrlarning tiplari oʻzaro mos kelishi kerak, aks holda dastur xato tuzilgan hisoblanadi. Yuqoridagi dasturdan koʻrinib turibdiki, a va b formal parametrlar oʻrniga natijaviy qiymatlari maʼlum ifodalar qoʻyildi. Demak, qiymatli faktik parametrlar oʻrniga, shu tipli natijaga erishuvchi ifoda yozilishi mumkin. Bundan tashqari, prosedurada kiritilgan a va b parametrlari faqat proseduraning ichidagina maʼnoga ega, tashqarida, misol uchun asosiy dasturda ular tushunarsiz, qiymatlari

aniqlanmagan miqdorlardir. SHuning uchun, qiymatli parametrlarga prosedura natijalarini o'zlashtirib, asosiy dasturga uzatib bo'lmaydi.

YUqorida tuzilgan dasturning asosiy kamchiligi, topilgan katta con doim s o'zgaruvchisiga o'zlashtiriladi. Misolimiz shartiga ko'ra esa, natijalar u va v o'zgaruvchilariga o'zlashtirilishi kerak edi. SHuning uchun, dasturda ikki marta qo'shimcha $u=s$ va $v=s$ o'zlashtirish operatorlari yozildi.

Bu kamchilikni tuzatish uchun proseduraga yana bir parametrni kiritamiz. Lekin, kiritilgan bu parametr proseduraga qiymat olib kirmaydi balki, prosedura natijasini asosiy dasturga olib chiqib ketadi. Bunday parametrni parametr - o'zgaruvchi deb ataladi.

Parametr-o'zgaruvchini parametr-qiymatdan farq qilish uchun prosedurani aniqlashdagi parametrlar ro'yhatida o'zgaruvchi oldidan **ByRef** xizmatchi so'zi yoziladi. Parametr - o'zgaruvchidan so'ng albatta, uning tipi ko'rsatib qo'yiladi. YUqorida aytganimizdek, formal parametr - qiymat o'rniga proseduraga murojat vaqtida shu tipli ifoda yozish mumkin bo'lsa, parametr - o'zgaruvchi uchun bu hol mutlaqo mumkin emas.

Prosedurani mukammallashtirib borish dinamikasini his etish uchun yana, yuqorida ko'rilgan maksimum topish misolining dasturini parametr - o'zgaruvchi ishlatgan holda ko'rib chiqamiz:

```
Dim x, y, u, v As Double
Dim s As Double
```

```
Sub max1(a, b As Double, ByRef s As Double)
  If a > b Then s=a Else s=b
End Sub
```

```
Private Sub Command1_Click()
  X=Val(Text1.Text)
  Y=Val(Text2.Text)
  Call max1(x + y, x * y, u)
  Call max1(u, 0.5, v)
  MsgBox (Str(u)+ " " + Str(v))
End Sub
```

SHunday qilib, bitta dasturni proseduraning uch xil varianti uchun tuzib chiqib, natijada ixcham va sodda dasturga ega bo'ldik.

Prosedura-funksiyani e'lon qilish. Funksiyaga murojaat qilish va undan foydalanish.

Hajmi katta va murakkab dasturlarni ishlab chiqishda, tabiiyki katta qiyinchiliklarga duch kelinadi. Katta, kompleks dasturlarni zarur muddatda yaratishga bitta dasturchining vaqti yetmaydi. Bunday hollarda, ya'ni muhim ahamiyatga ega bo'lgan va qisqa muddatlarda yaratilish kerak bo'lgan dasturlarni ishlab chiqish uchun dasturchilarning katta guruhini jalb etishga to'g'ri keladi. Bunday, yagona dasturni yaratishdagi parallel ish olib borishda prosedura va funksiyalarning roli juda katta bo'ladi. Bajarilishi kerak bo'lgan ishni mustaqil bo'limlarga ajratilib, har bir mustaqil ish alohida dasturlanib, keyinchalik ular yagona - asosiy dasturga birlashtiriladi.

Asosiy dasturda ishlatiluvchi o'zgaruvchilar va prosedura parametrlarini qanday tanlab olish kerak degan muammo, bajariladigan ishning eng og'ir qismlaridan biri bo'lib qoladi. Agar ularni bir-birlariga bog'lab yuborilsa, u holda asosiy dasturdagi biror o'zgaruvchiga

kiritilgan o'zgartirish, prosedurada ishlatilgan va shu o'zgaruvchiga bog'liq barcha ishlarni qaytadan tahlil qilib, tekshirib chiqishga olib keladi. Bunday chalkash va og'ir ishni bajarishning qiyinligi dastur tuzishda parallel, bir nechta dasturchining ish olib borishiga halaqit beradi.

SHuning uchun, prosedura va funksiyalarni yozishda har bir dasturga o'zi yechayotgan masalaga muvofiq holda, turli xil ichki o'zgaruvchilar, dasturli ob'ektlar o'zgaruvchilarining turli qiymatlarini tanlab olish huquqi beriladi. Xattoki, bitta o'zgaruvchini turli xil vazifalarda ishlatsa ham bo'ladi. Visual Basic tilida bunday masalani hal qilish uchun lokallashtirish prinsipi ishlab chiqilgan, ya'ni prosedura yoki funksiyada ishlatilgan o'zgaruvchi shu prosedura yoki funksiyaning ta'sir doirasida (ichida) gina o'z qiymatini saqlab qoladi. Prosedura va funksiyalarning ichida aniqlanib, qiymatlangan o'zgaruvchilarni lokal (ichki) o'zgaruvchilar deb ataladi. Tashqarida, ya'ni asosiy dasturda kiritilgan o'zgaruvchilar esa umuman olganda dasturning ixtiyoriy joyida o'z qiymatini saqlab qola oladi. Bu o'zgaruvchilarni global (tashqi) o'zgaruvchilar deb ataladi.

Quyidagi misolda lokallashtirish prinsipi yaqqol ko'zga tashlanadi:

```
Const n=1
Dim t, x As String
Private Sub p(x, y As Double)
Dim n As Double
N=x + t
T=y
MsgBox (Str(n) + " " + Str(t)+ " "+ Str(x))
End Sub
Private Sub Command1_Click()
T=n / 2
X="+"
Call p(n, 0.8)
MsgBox (Str(n) + " " + Str(t) + " "+ x)
End Sub
```

bu yerda t – asosiy dasturning global o'zgaruvchisi;
x, y – R prosedurasining formal parametrlar;
n – P proseduradagi lokal o'zgaruvchi.

Matematika kursidan funksiya tushunchasi bizga yaxshi tanish bo'lib, uning yordamida funksiya va argument o'rtasidagi bog'liqlik aniqlanadi. Visual Basic tilida ham funksiya tushunchasi kiritilgan bo'lib, uni shartli ravishda ikki turga ajratsak bo'ladi: standart funksiyalar va dasturchi tomonidan aniqlangan prosedura - funksiyalar. Standart funksiyalar har bir algoritmik til uchun aniqlanib, amalda ko'p uchrab turuvchi funksiyalarning qiymatlarini hisoblab berishga mo'ljallangan. Masalan, Sin(x), Cos(x), exp(x), Abs(x), sqr(x) va h.k.

Xuddi standart funksiyalar kabi dasturchi ham o'zi uchun zarur, mustaqil dastur ob'ektlarini funksiyalar ko'rinishida aniqlab, undan kerakli paytda foydalanishi mumkin.

Funksiya Visual Basic tilida quyidagi struktura bo'yicha aniqlanadi:

```
[Private][Public][Static] Function prosNomi (Parametrlar) [as type]
    operatorlar
End Function
```

YUqorida eslatib o'tganimizdek, funksiyalar ham proseduralar kabi mustaqil dasturlar hisoblanib, asosiy dastur orqali boshqariladi va xuddi asosiy dastur va proseduraga o'xshash ctukturada yoziladi.

Prosedura uchun aytilgan gaplarning deyarli barchasi funksiya uchun ham o'rinlidir. Funksiyaning proseduradan asosiy farqi quyidagilardir:

- funksiya carlavhasi boshqacha aniqlanadi;
- funksiyaning ishi davomida olinadigan natija funksiyaning ismiga o'zlashtiriladi, ya'ni funksiyaning tana qismida albatta, funksiya ismiga mos tipli qiymat o'zlashtirilgan bo'lishi kerak;
- funksiyaadan asosiy dasturga uning ismi orqali bittagina qiymat beriladi.

Funksiyaga murojat ham xuddi proseduradagi kabi amalga oshiriladi, lekin funksiyaning mos tipli ifodada qatnashishi kabi qo'shimcha imkoniyati mavjud.

Endi funksiyanani aniqlash va unga murojat qilishni to'liqroq o'rganish uchun quyidagi misolni ko'rib chiqamiz:

Misol: $f(n) = n!$ ($n! = 1 * 2 * 3 * \dots * n$ - faktorial) funksiyaadan foydalanib,

$$Y = \frac{20! + 3!}{5! + 31!} * \frac{(k + 1)!}{m!} - \text{ifodani hisoblashni tashkil qiling:}$$

Dim k, i As Integer

Dim m As Integer

Dim y As Double

Function Fact(n As Integer) As Integer

Dim p As Integer

Dim j As Integer

J=1

For p=1 To n

J=j * p

Next p

Fact=j

End Function

Private Sub Command1_Click()

K=Val(Text1.Text)

M=Val(Text2.Text)

Y=(Fact(2) + Fact(3)) / (Fact(5) + Fact(3)) * Fact(k + 1) / Fact(m)

MsgBox (Str(y))

End Sub

Funksiyalarni aniqlashda doim shunday harakat qilish lozimki, uning tana qismida formal parametrlar va funksiyanani aniqlash uchun zarur bo'lgan lokal o'zgaruvchilargina qatnashsin. Dasturning global o'zgaruvchisiga iloji boricha prosedura yoki funksiya ichidan turib qiymat bermaslik kerak, aks holda dastur xato natija berishi mumkin.

Rekursiv funksiyalar

Visual Basic tilida prosedura – funksiyalar bilan ishlashda, funksiyalarning rekursivlik hossasidan foydalanish imkoniyati yaratilgan.

Rekursiya tushunchasiga misol qilib oddiy faktorial hisoblashni keltirish mumkin:

$$n! = \begin{cases} 1 & \text{agar } n = 0 \\ n \cdot (n-1)! & \text{agar } n > 0 \end{cases}$$

bu yerda ko'rinib turibdiki $n!$ qiymati $(n-1)!$ orqali aniqlanayapti, ya'ni rekursiya degani o'zi orqali o'zini aniqlash ma'nosini anglatadi.

Visual Basic tili ham funksiyalarni rekursiv aniqlash imkoniyatini beradi. Funksiyani rekursiv aniqlash uning tana qismida o'ziga - o'zi murojat qilish orqali amalga oshiriladi.

YUqoridagi faktorial hisoblashni rekursiv funksiyalar orqali amalga oshiraylik:

```
Dim n As Integer
```

```
Dim y As Integer
```

```
Function Fact(m As Integer) As Integer
```

```
Dim k As Integer
```

```
If m=0 Then Fact=1 else Fact=Fact(m - 1) * m
```

```
End Function
```

```
Private Sub Command1_Click()
```

```
    N=Val(Text1.Text)
```

```
    Y=Fact(n)
```

```
    MsgBox (Str(y))
```

```
End Sub
```

Funksiyalarni rekursiv aniqlash qisqa va tushunarli tilda bo'ladi, rekursiv emas aniqlash esa uzoq va funksiyani ko'rinish effektini buzadi, lekin birinchi holda sarflangan EHM vaqti va xotira nisbatan ancha yuqoridir.

Mavzu. Fayllar bilan ishlash. Faylni yozish va o'qish uchun ochish. Fayl ishini yakunlash. Diskdagi fayl qanday o'qiladi?

YAratiladigan kompyuter dasturlarining ko'pchiligida diskdan o'qish yoki diskka yozish zarur. Ko'p hollarda bu jarayon foydalanuvchiga ko'rinmaydi, ya'ni siz Visual Basic dan foydalanayotgan bo'lsangiz, Visual Basic diskdagi ma'lumotlar bazasi bilan ishlashda ma'lumotlar bazasini boshqaruvchi yadrodan foydalanadi. Bu holda siz fayllarni ochish yoki yopish haqida tashvishlanmasangiz ham bo'ladi. Biroq axborotni o'qish va diskka yozishni bilib qo'yish kerak bo'lib qolishi mumkin.

Masalan, ish jarayonida dastur tashkiliy fayllardan birortasini kelgusida axborotni chiqarish uchun diskka ko'chirib olish zaruriyatini yuzaga kelishi mumkin.

Fayllar bilan ishlash uchun turli operator va funksiyalar mavjud bo'lib, ular quyidagi jadvalda keltirilgan:

Operator, funksiya	Vazifasi
Open	Faylni ochadi
Close	Barcha ochilgan fallarni yopadi
Close #	Ko'rsatilgan tartib raqamidagi faylni yopadi
Reset	Bufer(xotira)da saqlanayotgan barcha fayllarni yopadi
Print #	Ma'lumotlarni faylga yozadi
FileCopy	Fayldan nusxa oladi
EOF	Fayl oxirini aniqlaydi
FileAttr	Ochiq fayldan foydalanish turini aniqlaydi
FileDateTime	Faylning yaratilgan vaqtini va sanasini anig'laydi

FileLen	Fayl xajmini baytda aniqlaydi
FreeFile	Ochilgan fayllarning bo'sh qolgan tartib raqamini aniqlaydi
GetAttr	Fayl atributini aniqlaydi
SetAttr	Faylga atribut o'rnatadi
Loc	Ochilgan fayldagi aktiv turgan joyini aniqlaydi
LOF	Ochilgan faylning xajmini baytlarda aniqlaydi
Seek	Ochilgan faylning aktiv turgan joyini o'zgartiradi
Dir	Joriy turgan papka nomini aniqlaydi
Kill	Faylni o'chiradi
Lock	Ko'p foydalanuvchili muxitda faylni ximoyalaydi
UnLock	Ko'p foydalanuvchili muxitda fayl ximoyasini olib tashlaydi
Name	Fayl nomini o'zgartiradi
Get #	Fayl tarkibidagi ma'lumotni o'qiydi
Input #	Fayl tarkibidagi ma'lumotni o'qiydi
Line Input #	Fayl tarkibidagi satrni o'qiydi
Put #	Ma'lumotlarni faylga yozadi
Write #	Ma'lumotlarni faylga yozadi

Faylni ochish uchun Open buyrug'idan foydalaniladi. U quyidagicha ko'rinishga ega:

Open pathName For mode [access] [lock] As [#] fileNumber [LenqrecLength]

bu yerda:

- pathName – faylning to'la nomi;
- mode – fayl bilan ishlashga ruxsat olish turi. Ular Append, Binary, Input, Random turlaridir;
- access – fayl bilan ishlash usuli. Fayl o'qish(Read) uchun, yozish(Write) uchun yoki o'qish va yozish(ReadG'Write) uchun ochilishi mumkin;
- lock – Tarmoqdada ko'pchilik foydalanganida ishlatiladi. Qiymatlar va ularning ta'rifi quyidagicha: Shared (Umumiy), Lock Read (O'qishni ximoyalash), Lock Write (YOzishni ximoyalash), Lock Read Write (o'qish va yozishni ximoyalash);
- fileNumber – Faylni qayta ishlash uchun uning tartib raqami. Tartib raqam 1 dan 511 gacha bo'lishi mumkin (511 ta faylni bir vaqtda ochish mumkin);
- recLength – Faylga ma'lumotlarni yozish yoki o'qishda chegaralangan bufer (xotira xajmi).

Quyidagi misol matn faylini oynaga qanday yuklashni ko'rsatadi:

1. Yangi dastur yarating, unga matn oynasi va buyruq tugmasini joylashtiring. Matn oynasining Scrollbars xususiyatini Vertical holatiga, Multiline xususiyatini esa True holatiga o'tkazing.
2. Tugmaning Click(Bosish) hodisasiga quyidagi kodni yozing:

```

On Error GoTo ErrorHandler
Dim FileNum As Integer
FileNum=FreeFile
Open "C:\AUTOEXEC.BAT" For Input As #FileNum
Text1.Text=Input(LOF(FileNum), #FileNum)
Close #FileNum
Exit Sub
ErrorHandler:

```

MsgBox "Xatolik:" & Err.Description

3. Dasturni ishga tushiring va tugmani bosing. Matn oynasida AUTOEXEC.BAT faylining mazmuni chiqadi.

Bu misoldagi kodning ishlashi haqida bir necha fikrlar. Visual Basic ochiq fayllarni nomeri bo'yicha identifikasiyalaydi. FreeFile funksiyasi bo'sh nomerni qaytaradi, Open ... For Input operatori faylni ochadi, keyin esa Input funksiyasi faylni o'qiydi. LOF (Length of File – fayl uzunligi) funksiyasi Input funksiyasi o'qishi uchun fayl uzunligini aniqlaydi. O'qish operatsiyasi so'ngida Close# funksiyasi faylni yopadi.

Faylga ma'lumot yozish

Faylga ma'lumot yozish kodi faylni o'qish kodiga o'xshash. Aslida, fayl o'qishG' yozish rejimida ochilganda siz faylni o'qishingiz va zarur bo'lsa, unga ma'lumot yozishingiz mumkin. Faylga ma'lumot yozish quyidagicha ko'rinishga ega buladi:

YUqoridagi kabi oyna xosil qiling va Command1 tugmachasini Click xodisasiga keltiring hamda quyidagi kodni yozing:

```
On Error GoTo ErrorHandler
Open "C:\MyFile.tmp" For Output As #1
Write #1, Text1.Text
MsgBox "Faylga Text1 maydonidagi yozuvlar saqlandi"
Close #FileNum
Exit Sub
ErrorHandler:
MsgBox "Xatolik:" & Err.Description
```

Text1 maydoniga o'zingizning tarjimai xolingizni kiritib uni Bloknnot yordamida "C:\MyFile.tmp" dan ko'rishingiz mumkin.

Ro'yxatga oluvchi faylni yaratish

Bu misolda dasturni ishga tushirilgan hamda yopilgan sanasi va vaqtini o'ziga oladigan ro'yxatga olish faylini yaratish namunasi keltirilgan.

Eng muhimi faylni to'g'ri rejimda ochish zarur. Agar siz fayldan ikkilik ma'lumot o'qimoqchi bo'lsangiz, faylni (ikkilik ma'lumot) o'qish (Open for Binary) rejimida oching. SHu bilan birga siz faylni qaysi rejimda ochilishidan qa'tiy nazar ma'lumotlar bazasiga murojat qilish rejimini ham belgilashingiz mumkin.

1. YAngi oyna (Form) yarating va uning Load(YUklash) hodisasiga quyidagi kodni yozing:

```
On Error GoTo ErrorHandler
Open "C:\EASYSSTEP.LOG" For Append As #1
Write #1, "Dastur ishga tushdi: " & Str$(Now)
Close #1
Exit Sub
ErrorHandler:
MsgBox "Xatolik: " & Err.Description
```

2. YAratilgan oynaning Unload hodisasi uchun ham yuqoridagi kodni yozing va Write operatori bilan boshlanuvchi satrni quyidagi satr bilan almashtiring:

Write #1, «Dastur yopiladi:» & Str\$(Now)

Bu misol dastur birinchi marta ishga tushirilganida ro'yxatga oluvchi fayl hosil qiladi. Keyin ishga tushganda esa axborotni mavjud faylga yozib boradi.

YUqoridagi dasturni bir necha bor (kamida ikki marta) ishga tushiring hamda bloknot yordamida "C:\EASYSSTEP.LOG" faylini ochib ko'ring. U yerda mavjud dastur qachon ishga tushirilganligi va qachon yopilganligi yozib qoldirilgan.

Mavzu. Visual Basic ning grafik imkoniyatlari. Geometrik figuralarni xosil qilish. Xarakatli tasvirlarni xosil qilish.

Visual Basic ning grafik imkoniyatlari yetralicha katta bo'lib, uni bir yoki ikkita mavzu yordamida qamrab olib bo'lmaydi. SHuning uchun Visual Basic ning grafik imkoniyatlarining asosiylari xaqida to'xtalib o'tishga xarakat qilamiz.

Visual Basic ning grafik imkoniyatlari yordamida quyidagi asosiy amallarni bajarish mumkin:

- formadagi axborotni chiziq, ramkalar va freymlar yordamida guruxlash va belgilash;
- grafik tasvirli axborotlarni qo'shish
- unga animasiyalarni qo'shish bilan interfeysni kuchaytirish.

Visual Basic da grafika va tasvirlar bilan ishlash uchun grafik ob'ektlar va uslublardan foydalaniladi:

- formaga tasvirlarni o'rnatish va ishlash;
- Picture boshqaruv elementini qo'llash;
- Image boshqaruv elementini qo'llash.

Grafika bilan ishlash uchun oddiy boshqaruv elementlari.

Boshqaruv elementlari panelida oddiy grafik elementlarni yaratish uchun tugmalar mavjud.

Tugma	Nomlanishi	Vazifasi
	Line (to'g'ri chiziq)	Formada chiziq xosil qiladi
	Shape (kontur)	Formada konturni xosil qiladi
	Frame (ramka)	Formada ob'ektlarni guruxga qo'shish imkonini beruvchi ramka xosil qiladi

Line boshqaruv elementi - qalinligi odatda 1 ga teng bo'lgan to'g'ri chiziq. CHiziq ko'rinishini o'zgartirish uchun Line ob'ektining BorderStyle xususiyatidan foydalaniladi. Bu xususiyat quyidagi qiymatlarni qabul qiladi:

- 0 – Transpared ko'rinmas chiziq;
- 1 – Solid oddiy chiziq;
- 2 – Dash shtrixli chiziq;
- 3 – Dot punktirli chiziq;
- 4 – Dash-Dotshtrix-punktirli chiziq;
- 5 – Dash-Dot-Dot ikki punktirli shtrixpunktir chiziq;

CHiziq qalinligi `BorderWidth` xususiyati yordamida, uning rangi esa `BorderColor` xususiyati yordamida beriladi.

`Shape` boshqaruv elementi turli ko'rinishdagi ramkalarni xosil qilish uchun qo'llaniladi. `Shape` ob'ekting `Shape` xususiyati quyidagilarni qabul qiladi:

0 – Rectangle	to'g'ri to'rtburchak;
1 – Square	kvadrat;
2 – Oval	ellips;
3 – Circle	aylana;
4 – Rounded Rectangle	silliqlik burchakli to'g'ri to'rtburchak
5 – Rounded Square	silliqlik burchakli to'g'ri kvadrat

Ramkaning chiziq tipi, qalinligi va rangini `Line` ob'ekti singari o'rnatiladi.

`Frame` boshqaruv elementi sarlavxali ramkaga o'xshab ketadi. Ramka chizig'i o'stidagi sarlavxa ob'ektning `Caption` xususiyati orqali beriladi. `Frame` ob'ekti o'ziga joylashuvchi ob'ektlarning ususiylarini boshqaradi. Masalan, `Frame` ning `Visible` xususiyatiga `False` qiymat berilsa, ya'ni ob'ekt ko'rinmas xolatga qo'yilsa, u xolda unda joylashgan barcha ob'ektlar ham ko'rinmas xolatga o'tadi.

Grafik tasvirlarni formaga o'rnatish uchun `Image` va `Picture` ob'ektlaridan foydalaniladi. Buning uchun boshqaruv elementlari panelidan quyidagi tugmalar qo'llaniladi:

Tugma	Nomi	YAratiluvchi ob'ekt
	Image	Image
	PictureBox	Picture

Ushbu ob'ektlardan joylashuvchi tasvir sifatida faqat aniqlangan tipdagi fayllardan foydalaniladi. Aniqlangan tiplar quyida berilgan: `bmp`, `dib`, `ico`, `cur`, `wmf`, `emf`, `gif`, `jpg`, `jpeg`.

Tasvirlar `Image` ob'ektiga uning `Picture` xususiyati yordamida o'rnatiladi. `Image` ob'ektida joylashgan tasvirning o'lchamlari ob'ektga nisbatan proporsional ravishda o'zgaradi. O'lchamlarni o'zgartirishni `Stretch` xususiyati bilan boshqariladi. Agar bu xususiyat `False` qiymatga ega bo'lsa, u xolda `Image` ob'ekti tasvir o'lchamlariga bog'liq bo'ladi. Agar qiymat `True` ga teng bo'lsa, u xolda tasvirning o'lchami ob'ekt o'lchamiga bog'liq ravishda o'zgaradi.

`Picture` boshqaruv elementi `Image`dan farqli ravishda to'liq konteyner sifatida ishlay oladi. U boshqa bir boshqaruv elementlarini qabul qilishi mumkin. YA'ni grafik uslub (`Line`, `Circle`, `Point`, `Pset`) yordamida chizish yoki `Print` uslubi yordamida matnni xosil qilish mumkin.

`Picture` boshqaruv elementi `Visible` xususiyatiga ega bo'lib, uning qiymatiga bog'liq xolda ob'ektga o'rnatilgan tasvirlarni ko'rinishi yoki ko'rinmas bo'lib qolishi mumkin. `Picture` ob'ekti yordamida `PaintPicture` uslubini qo'llash bilan oddiy animasiyalarni qurish mumkin.

Formani loyixalash vaqtida grafik tasvirni o'rnatish uchun ob'ektning `Picture` xususiyatidan foydalaniladi:

1. Ob'ektning `Properties` oynasi ochiladi;
2. `Picture` xususiyati tanlanadi;
3. Xususiyat oynasining o'ng ustunidagi uch nuqtali tugma bosiladi;
4. Xosil bo'lgan `LoadPicture` muloqot oynasida kerakli fayl tanlanadi;
5. Otkro't tugmasi bosiladi.

Grafik uslublar

Visual Basic grafik boshqaruv elementlariga qo'shimcha ravishda grafik uslublar to'plamiga xam egadir:

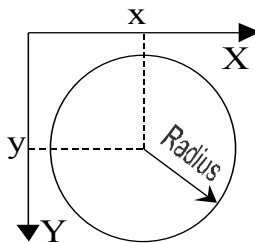
№	Uslub	Aniqlanishi
1.	Circle	Aylana, yoy yoki ellips chizish
2.	Cls	Chiqarish soxasini tozalash
3.	Line	To'g'ri chiziq chizish
4.	PaintPicture	Tasvirni chizish (yuklash)
5.	Paint	Nuqta rangini aniqlash
6.	Print	Matnni xosil qilish
7.	Pset	Nuqta xosil qilish

Grafik uslublarda ranglarni baxolash RGB va QBColor funksiyalari bilan amalga oshiriladi.

Circle uslubi

Circle ning yozilish formati quyidagicha ifodalanadi:

Object.Circle[Step](X,Y),Radius,[color,start,end,aspect]



17-rasm. Aylananing ob'ektga nisbatan joylashishi.

Bu yerda,

- Object – uslub qo'llanilayotgan ob'ekt nomi. Agar ob'ekt ko'rsatilmasa, u xolda ob'ekt deb forma ko'rsatiladi;
- X,Y – chiziluvchi aylana, yoy yoki ellipsning markazi koordinatasi. Koordinataning o'lchov birligi ob'ektning ScaleMode xususiyatida ko'rsatiladi.
- Radius - chiziluvchi aylana, yoy yoki ellipsning radiusi. Radiusning o'lchov birligi ob'ektning ScaleMode xususiyatida ko'rsatiladi.
- Color – Long tipidagi qiymat bo'lib, chiziq rangini belgilaydi. Agar ushbu parametr ko'rsatilmasa, u xolda ForeColor xususiyatida ko'rsatilgan rang qabul qilinadi. Rangni berish uchun RGB() yoki QBColor funksiyalaridan xam foydalansa bo'ladi.
- Start, End – yoy yoki ellipsning bir qismi chizilayotganda yoyning radiandagi boshlang'ich va oxirgi nuqtalari: 2π dan -2π gacha.
- Aspect – aylananing elliptiklik koeffitsienti. Aylana uchun ushbu parametr 1.0 ga teng.

CLS uslubi yordamida tasvir chizilayotgan ob'ekt tozalanadi:

Object.Cls

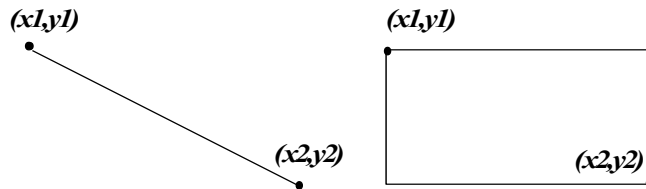
Bu yerda Object – tasvir chizilayotgan ob'ekt nomi.

Line uslubi.

Line uslubi to'g'ri chiziq chizish uchun qo'llaniladi (18-rasm) va quyidagicha yozilish formatiga ega:

Object.Line(x1,y1)-(x2,y2),[color],[B][F]

- Object – uslubi qo'llanilayotgan ob'ekt. Agar ob'ekt ko'rsatilmasa, u xolda ob'ekt sifatida forma ko'rsatiladi;
- x1, u1 – chiziluvchi chiziqning boshlang'ich nuqtasining koordinatalari;
- x2, u2 – chiziluvchi chiziqning oxirgi nuqtasining koordinatalari;
- color – chiziqning RGB palitradagi rangi. Agar rang ko'rsatilmasa, u xolda ob'ektning ForeColor xususiyatida ko'rsatilgan rang qabul qilinadi;
- V – to'g'ri to'rtburchak chizishni belgilaydi. Bunda x1,u1 lar to'g'ri to'rtburchakning yuqori chap va x2,u2 lar esa pastki o'ng burchak koordinatalarini belgilaydi.
- F – to'g'ri to'rtburchak soxasini chiziq rangida bo'yash.



18-rasm. Line uslubida koordinatalarni belgilash

Point uslubi.

Point uslubi nuqtaning RGB palitrasidagi rangini aniqlaydi. Uslubning yozilish formati quyidagicha:

Object.Point(X,Y)

- Object – uslub qo'llaniladigan ob'ekt. Agar ob'ekt ko'rsatilmasa, u xolda ob'ekt sifatida forma ko'rsatiladi;
- X, Y – ob'ektda joylashgan nuqtaning koordinatasi.

Print uslubi.

Print uslubi ko'rsatilgan ob'ektda matn xosil qilish uchun qo'llaniladi. Uslubning yozilish formati quyidagicha:

Object.Print[OutputList]

- Object - uslub qo'llaniladigan ob'ekt. Agar ob'ekt ko'rsatilmasa, u xolda ob'ekt sifatida forma ko'rsatiladi;
- OutputList – ob'ektda xosil qilinuvchi satrli ifoda yoki ifodalar ro'yxati.

Pset uslubi.

Pset uslubi ob'ektning ko'rsatilgan nuqtasini rangini belgilaydi. Ushbu uslubni Point ning teskarisi deyish xam mumkin. YOzilish formati quyidagicha:

Object.Pset(x,y),[color]

- Object - uslub qo'llaniladigan ob'ekt. Agar ob'ekt ko'rsatilmasa, u xolda ob'ekt sifatida forma ko'rsatiladi;
- x,u – ob'ektdagi nuqta koordinatasi;
- color – nuqtaning RGB palitradagi rangi.

YUqorida ko'rib o'tilgan uslublar yordamida kichik dasturni misol sifatida ko'rib o'taylik.

Formaga bitta command va Picture ob'ektlarini o'rnatamiz va ularning xususiyatlarini quyidagicha o'zgartiramiz:

Form1

Caption	Gragik
---------	--------

Picture1

ScaleMode	3-Pixel
-----------	---------

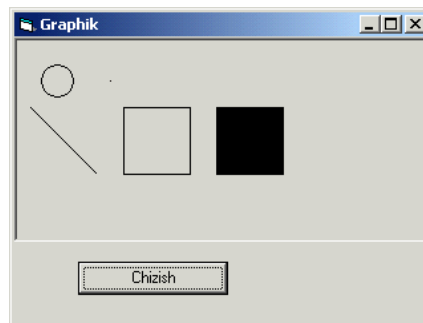
Command1

Caption	Chizish
---------	---------

Command1 tugmasi ustida sichqoncha tugmasi ikki marta tez bosish bilan kodlar oynasida tegishli prosedura xosil bo'ladi. Xosil bo'lgan prosedurada quyidagicha o'zgartirishlarni amalga oshiramiz:

```
Private Sub Command1_Click()  
    Picture1.Circle (30, 30), 12 'aylana  
    Picture1.Line (10, 50)-(60, 100) 'to'g'ri chiziq  
    Picture1.Line (80, 50)-(130, 100), B 'to'g'ri to'rtburchak  
    Picture1.Line (150, 50)-(200, 100), BF 'bo'yalgan to'g'ri to'rtburchak  
    Picture1.PSet (70, 30), 15 'nuqta  
End Sub
```

SHundan so'ng G'5 tugmasini bosamiz, dastur ishga tushadi. Chizish tugmasini bosganimizdan keyin, ekranda dastur natijasini ko'rishimiz mumkin (19-rasm).



19-rasm. Dastur oynasi.

Xarakatli tasvirlarni yaratish.

Xarakatlanuvchi tasvir deganda biror ob'ekt yoki rasm (chiziq, aylana va x.k.)larni ekran bo'ylab siljishi tushuniladi. Bunday jarayonni qanday amalga oshiriladi? Umuman olganda, bu juda xam oddiy deyish mumkin. YA'ni oddiy bir chiziq chiziladi, bir oz vaqt o'tgandan so'ng ushbu chiziq o'chirib tashlanadi va boshqa bir koordinatada xosil qilinadi. Bunday o'chirib chizishlar bir necha marotaba takrorlanganda chiziq ma'lum bir yo'nalish bo'yicha xarakatlanayotganga o'xshab ketadi.

Misol sifatida quyidagi masalaga dastur tuzaylik: aylana ekranda o'ngdan chapga va chapdan o'ngga borib-kelib turishini tashkil eting.

Buning uchun quyidagi ishlarni amalga oshirish lozim:

- Form1 forma tashkil qilamiz; Caption xususiyatini «Harakatdagi aylana» deb o'zgartiramiz;
- Formaga General oynasidan PictureBox komponentini formaga o'rnatamiz (20-rasm). Bu ob'ekt aylanani xosil qiluvchi ob'ekt sifatida foydalaniladi. Aylananing xarakatlanishi xam aynan shu ob'ekt ichida amalga oshiriladi;
- Formaga Command komponentidan xam bitta o'rnatamiz. Aylananing xarakati shu tugma bosilishi bilan boshlanadi yoki to'xtatiladi. Caption xususiyatini «Start» o'zgartiramiz.
- Ma'lumki xarakat xar doim vaqt bilan uzviy bog'liq xisoblanadi. SHuning uchun formaga

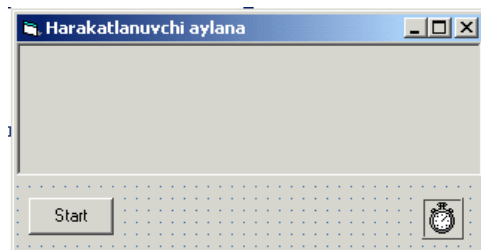
Timer  komponentini xam o'rnatamiz. Interval xususiyatini 200, Enabled xususiyatini esa False ga o'zgartiramiz. Bu yerda Interval vaqtning o'zgarish tezligi bo'lib, qanchalik

katta qiymat berilsa, vaqtning o'zgarishi shunchalik sekinlashadi. Enabled xususiyati esa vaqt xisoblanishini boshlash yoki boshlamaslikni belgilaydi.

SHunday qilib formaning umumiy ko'rinishi 21-rasmda ko'rsatilgan shaklni oladi.



20-rasm. PictureBox komponenti



21-rasm. Dasturning forma ko'rinishi

Endi dastur kodini yozish uchun Start tugmasi ustida sichqon tugmasi ikki marta tez bosiladi. Xosil bo'lgan kodlar oynasida quyidagi o'zgartirishlarni amalga oshiramiz:

```
Private Sub Command1_Click()
If Command1.Caption="Stop" Then
Command1.Caption="Start"
Timer1.Enabled=False
Else
Command1.Caption="Stop"
Timer1.Enabled=True
End If
D=5
End Sub
```

Bu prosedurada Command tugmasining Caption xususiyatini xolatga ko'ra o'zgartirish amalga oshirilgan. Ya'ni agar Start bosilgan bo'lsa tugma ustidagi yozuv Stop ga o'zgaradi yoki aksincha. SHunga ko'ra aylana xarakati xam davom etadi va to'xtaydi. d o'zgaruvchi esa xarakat qadami bo'lib, uni ixtiyoriy berish mumkin. Agar katta bo'lib ketsa, xarakat tekis chiqmasligi mumkin.

Aylanani xosil qilish va uning navbatdagi koordinatada chizilishi quyidagi prosedurada tashkil etilgan. Ushbu prosedura Timer komponenti ustida sichqoncha tugmasi ikki marta tez bosish bilan xosil qilinadi va quyidagi o'zgartirishlar qilinadi:

```
Private Sub Timer1_Timer()
X=x + d
Picture1.ForeColor=&H8000000F
Picture1.Circle (x - d, 30), 10
Picture1.ForeColor=QBColor(0)
Picture1.Circle (x, 30), 10
If x > 300 Or x < 5 Then d=d * (-1)
```

End Sub

E'tibor qiladigan bo'lsak, bu yerda d o'zgaruvchiga boshlang'ich qiymat avvalgi prosedurada berilgan, ya'ni ushbu o'zgaruvchi barcha proseduralarda ishlashi kerak. Buning uchun o'zgaruvchi global qilib e'lon qilinishi kerak:

Dim d As Integer

X o'zgaruvchi esa aylananing x o'qi bo'yicha koordinatasi bo'lib, u d qadam bilan o'zgarib turadi. Aylana avval ko'rinuvchi chiziqda chiziladi va uni o'chirish uchun uning ustidan fon rangida ustidan chiziladi. Yangisi esa keyingi koordinatada xosil qilinadi. SHunday qilib aylana ekran bo'ylab o'ngdan-chapga va aksincha chapdan-o'ngga xarakatlanib turadi. Dasturning umumiy ko'rinishi quyidagicha bo'ladi:

Dim x , d As Integer

Private Sub Command1_Click()

If Command1.Caption="Stop" Then

Command1.Caption="Start"

Timer1.Enabled=False

Else

Command1.Caption="Stop"

Timer1.Enabled=True

End If

$D=5$

End Sub

Private Sub Timer1_Timer()

$X=x + d$

Picture1.ForeColor=&H8000000F

Picture1.Circle ($x - d$, 30), 10

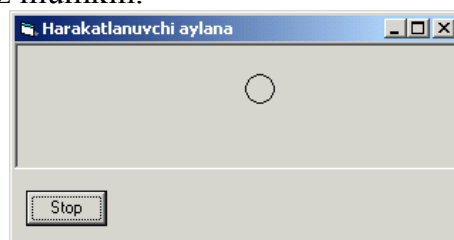
Picture1.ForeColor=QBColor(0)

Picture1.Circle (x , 30), 10

If $x > 300$ Or $x < 5$ Then $d=d * (-1)$

End Sub

Dasturni ishga tushirib, Start tugmasi bosilsa aylana xarakatlanayotgan 22-rasmda ko'rsatilgan formani ko'rishimiz mumkin.



22-rasm.

Mavzu. Foydalanuvchi interfeysini qayta ishlash. Interfeys tiplari, elementlari (menyu, qurilmalar paneli, muloqot oynalari), CommonDialog

Interfeys – foydalanuvchi ishini osonlashtiruvchi ichki qobiq dasturi hisoblanib, foydalanuvchiga ko'rinmas mexanizmlarni boshqarish hamda kompyuterdagi xujjatlar va boshqa ma'lumotlar ustida ishlash imkoniyatini beradi. Xar qanday dasturning asosiy

maqsadi – ma'lumotlarni qayta ishlashda eng yuqori qulaylik va effektli ishlashdan iborat (Xujjatlar, ma'lumotlar ombori, grafika yoki tasvir va boshqalar). Bunday dasturlarni INTERFEYS dan foydalanmay turib tuzish imkoni mavjud emas. SHuning uchun interfeys xar qanday dastur tuzishning eng asosiy bo'limi bo'lib qoladi.

Interfeyslarni qayta ishlashda quyidagi prinsplarni qo'llash kerak:

- Standartlashtirish. Interfeyslardan foydalanishda, standart, ko'p dasturchi va foydalanuvchilar tomonidan sinovdan o'tganlari ishlatilishi lozim;
- Foydalanuvchi uchun qulaylik. Interfeys barcha uchun tushunarli bo'lishi lozim. Barcha amallar tez esda qolishi va ko'p amallar bajarilmasligi kerak. Masalan tugmachalarning ortiqcha bosilishi, shart bo'lmagan so'rovlarning chiqishi va boshqalar;
- Ichki dizayn. Interfeys ko'rinishi ko'zni tez toliqtirmasligi balki foydalanuvchi uni ishlatishda butun kun bo'yi ishlashi lozimligini hisobga olish lozim;
- Formani jixozlash. Forma boshqaruv elementlari bilan to'ldirilishi talab etiladi;
- Gruppalash. Boshqaruv elementlarini vazifasiga qarab guruxlarga ajratiladi. Bu esa foydalanuvchi kerakli amallarni tez bajarishda qo'l keladi.
- Formaga ob'ektlarni joylashtirish. Ob'ektlar formada bir biri bilan zich joylashmay erkin joylashishi lozim bo'ladi. CHunki foydalanuvchi erkni ishlashi kerak.

Interfeys turlari.

Windows operasion sistemasi uchun Visual Basic 6 yordamida dasturlashda uch xil turdagi interfeys mavjud va ular quyidagilar: yakka dokumentli SDI (Sindle-Document Interface), ko'p dokumentli MDI(Multiple-Document Interface) va provodnik turidagi interfeys (Explorer).

Eslatma: YUqorida keltirilgan va quyida keladigan dokument so'zlari qandaydir xujjat emas balki, ma'lumotlar ustida turli amallar bajaruvchi forma haqida boradi.

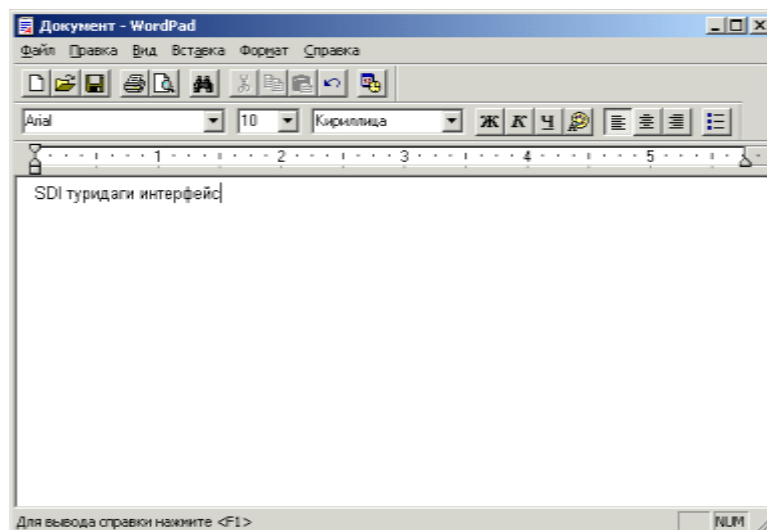
YAkka xujjatli interfeys – bu tipdagi interfeys faqatgini bir oynada yakka xujjat bilan ishlay oladi. Bunga misol qilib Microfoft WordPad dasturini aytish mumkin (11-rasm). Ko'p xujjatlar bilan bir vaqtda ishlash uchun bu dasturni bir necha bor ishga yuklash zarur bo'ladi. U xar gal ishga yuklanganida bir xil dastur xotiradan joy olaveradi natijada kompyuter xotirasi keraksiz bir xildagi ma'lumotlar bilan to'lib qoladi va uning ishlashi sezilarli darajada sekinlashadi.

YUqoridagilardan kelib chiqib bu turdagi interfeysdan foydalanishni butunlay yo'q qilmaslik lozim chunki ularning ham o'z o'zni bor. Masalan nusxa ko'chiruvchi yoki kichik imkonyatli interfeyslar.

SDI – interfeys.

SDI turidagi interfeys quyidagi boshqaruv elementlaridan tashkil topgan:

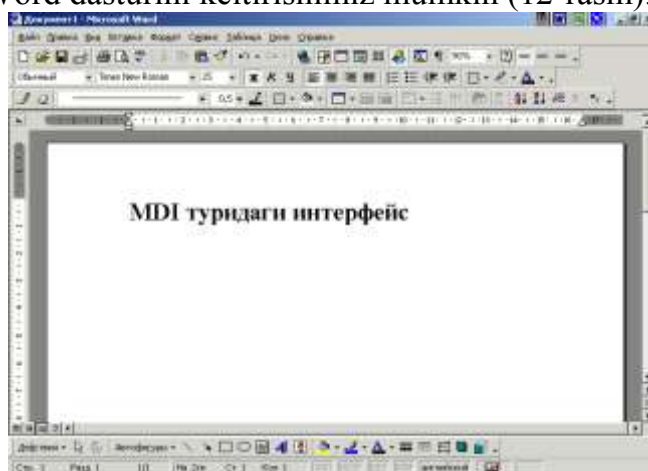
- bosh menyu;
- boshqaruv elementlaridan tashkil topgan panel;
- taxrirlanadagan ma'lumotlarni ko'rsatuvchi oyna;
- ma'lumotlar bilan ishlash uchun boshqaruv elementlari;
- ish xolatini ko'rsatuvchi ostki satr.



11-rasm. WordPad interfeysi

MDI – turidagi interfeys.

MDI – turidagi interfeysning asosiy vazifasi xar xil ma`lumotlardan iborat bo`lgan bir necha xujjatlarni o`z ichiga olish va ularni qayta ishlashdan iborat. MDI – turidagi interfeysga misol qilib Microsoft Word dasturini keltirishimiz mumkin (12-rasm).



12-rasm. MS Word interfeysi

Bunday turli interfeys uchun avlod qoldiruvchi oyna zarur. U bir nechta yaratilgan oynalarni boshqarish imkoniyatiga ega bo`ladi. Yangi yaratilgan avlod oynalarining son chegarasi kompyuter imkoniyatiga bog`liq.

MDI turidagi interfeys tarkibiga quyidagi elementlar kiradi:

- bosh menyu;
- boshqaruv elementlaridan tashkil topgan panel;
- dasturning asosiy oynasi;
- taxrirlanadagan ma`lumotlarni ko`rsatuvchi avlod oyna(lari)si;
- avlod oyna(lari)sidagi ma`lumotlar bilan ishlash uchun boshqaruv elementlari;
- ish xolatini ko`rsatuvchi ostki satr.

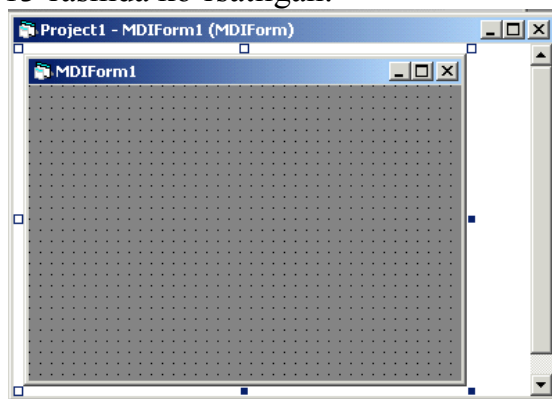
Endi esa quyida MDI turidagi interfeysni yaratish texnologiyasini ko`rib chiqishimiz mumkni.

MDI – interfeysning avlod qoldiruvchi oynasi.

Avlod qoldiruvchi forma yaratish uchun quyidagi amallar ketma-ketligi bajariladi:

- VB6 ishga yuklanganidan so`ng Standard EXE satri tanlanadi va OK tugmasi bosiladi;
- Project menyusidan Add MDI Form buyrug`i tanlanadi.

MDI formaning ko'rinishi 13-rasmda ko'rsatilgan:



13-rasm. MDI avlod qoldirish oynasi

Avlod qoldiruvchi oyna va avlod oynalarining asosiy xususiyatlari quyidagi jadvalda keltirilgan: Bu xodislarni taxrirlash uchun Properties oynasidan foydalaniladi.

Xususiyat	Vazifasi
BackColor	Forma fonining rangini o'zgartiradi.
BorderStyle	Forma turini tanlash. (Xajm, diologli, yakka ramkali va h.)
Caption	Forma sarlavhasini o'zgartiradi. (MDI Form1 ← MDI formasi)
ControlBox	Oyna sarlavhasidagi tugmachalarni ko'rsatishni boshqaradi
Font	Formadagi shrift turini o'zgartirish
ForeColor	Formada joylashgan matn rangini o'zgartirish
Height	Forma balandligini o'zgartirish
Icon	Formaga belgi qo'yish
Left	Formani ekranning chap qismiga nisbatan joylashishi
MDIChild	Formani MDI – avlodiga aylantiradi (True)
Name	Forma nomini o'zgartirish
ScaleMode	Formaga tegishli o'lchov birligini o'zgartirish (mm, sm, dyum, piksel...)
ShowInTaskbar	Topshiriqlar panelida ko'rsatish yoki ko'rsatmaslikni ta'minlaydi
Width	Forma enini o'zgartiradi
WindowState	Forma xolatini belgilaydi: 0-Normal – Oddiy ko'rinish; 1-Minimized – Kichiklashtirilgan ko'rinish; 2-Maximized – Ekranni to'la qoplaydigan ko'rinish.

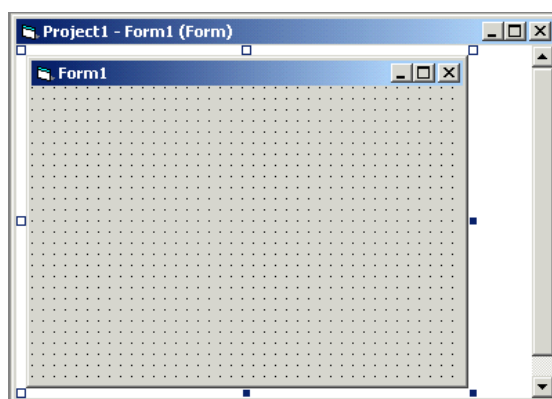
Barcha ob'ektlar ustida turli xodisalar(Sobo'tiya-Events)ni bajartirish mumkin. Masalan Ob'ekt sirtida sichqonchaning chap tugmasi bosilishi yoki Enter tugmasining bosilishi. Forma uchun ham turli xodisalar mavjud bo'lib ularning asosiylari quyidagi jadvalda keltirilgan:

Xodisa	Vazifasi
Activate	Forma aktivlashganida (joriy ya'ni ko'ringanida) ro'y beradigan xodisa
Click	Forsa sirtida sichqonchaning chap tugmasi bosilganida yoki Enter, Space tugmalari bosilganida ro'y beradi

DdlClick	Forma sirtida sichqonchaning chap tugmasi ikki marta bosilganida ro'y beradi
Deactivate	Forma aktivlik xolatidan chiqqanida ro'y beradigan xodisa
GotFocus	Formaning sirti aktiv bo'lganida ro'y beradigan xodisa
Initialize	Forma instalyasiya bo'lganida ro'y beradi
KeyUp	Tugmacha bosib qo'yib yuborilgandida ro'y beradi
Load	Forma ishga yuklanayotganida ro'y beradi
MouseUp	Forma sirtida sichqonchaning chap tugmasi bosilganida ro'y beradi
Resize	Forma xajmi o'zgarganida ro'y beradi
UnLoad	Forma ekrandan yo'qolib xotiradan o'chirilayotganida ro'y beradi

MDI – interfeysining yaratiluvchi avlod oynasi.

Yaratiluvchi avlod oynasi quyidagi rasmda ko'rsatilgan:



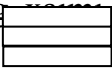
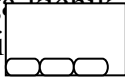
14-rasm. Yaratiluvchi avlod oynasi

Avlod oynasi avlod qoldiruvchi oyna tarkibida yaratilib, uni asosiy oyna ichida xaraktlantirish mumkin.

Avlod oynasining asosiy oynada joylashish turini Arrange uslubi yordamida o'zgartirish mumkin, u quyidagi jadvalda keltirilgan:

Xodisa	Kiy-mati	Vazifasi
vbCascade	0	Kaskadli joylashish. Yangi yaratilgan oyna avvalgi oynaga nisbatan o'ng yon tomonda pasroqda xosil bo'ladi.
vbTileVertical	1	Asosiy oynadagi barcha



		oynalar vertikal xolatda joylashadi va ularning xajmi bir xil bo'ladi.	
cbTileHorizontal	2	Asosiy oynadagi barcha oynalar vertikal xolatda joylashadi va ularning bir xil bo'ladi.	
vbArrangeIcon	3	Barcha avlod oynalari asosiy oynaning quyi qismiga ko'rinishda joylashadi	

Biz yuqorida asosiy va avlod oynalarining xususiyat, xodisa va turlari bilan tanishib chiqdik. Endi esa ularni yaratish dasturini ko'rib chiqamiz. Quyidagi amallar ketma-ketligini bajaring:

1. Yangi proekt yarating. Buning uchun File menyusida New Project (Proekt yaratish) buyrug'i sirtida sichqonchaning chap tugmasini ikki marta bosing va xosil bo'lgan oynadan Standart EXE belgisini tanlab Enter tugmasini bosing
2. Project menyusidan Project Properties buyrug'ini tanlang
3. Xosil bo'lgan oynada joylashgan Project Name maydoniga MyMDIApp so'zini kiriting va OK tugmasini bosing
4. Properties oynasidan MDIChild xususiyatini topamiz va unga True qiymatini beramiz
5. Formaning Name xususiyatiga frmChildMDI nomini beramiz
6. Caption xususiyatiga esa «MDI ning avlod oynasi» degan sarlavxa kiritamiz
7. Proektga MDI formani qo'shishi uchun Project menyusidan Add MDI Form buyrug'ini tanlaymiz
8. Uning Name xususiyatiga mdiParentMDI nomini beramiz
9. Uning Caption xususiyatiga «MDI ning avlod qoldiruvchi oynasi» degan sarlavxa kiritamiz
10. Turli amallarni bajarish uchun yuqori menyu xosil qilamiz. U quyidagicha yaratiladi:
 - Avlod qoldiruvchi oynani tanlaymiz.
 - Tools menyusidan Menu Editor buyrug'ini tanlaymiz.
 - Name maydoniga mnuFile so'zini kiritamiz.
 - Caption maydoniga Fayl so'zini kiritamiz va Next tugmasini bosamiz.
 - Ko'rsatkichi o'ng tarafga qaragan tugmachani bosamiz.
 - Caption maydoniga Yangi forma so'zini va Name maydoniga mnuFileNewForm so'zini kiritamiz hamda Next tugmasini bosamiz.
 - Yana Caption maydoniga Oyna so'zini va Name maydoniga mnuWindow so'zini kiritamiz hamda OK tugmasini bosamiz.

Endi esa uni Run menyusidagi Start buyrug'i yordamida ishlatib ko'rishimiz mumkin. Dastur ishini to'xtatish uchun End buyrug'ini tanlaymiz.

11. Yangi oyna yaratish uchun dastur tuzishimiz lozim bo'ladi. Buning uchun avlod qoldiruvchi oynani tanlaymiz va uning sirtida sichqonchaning chap tugmasini ikki marta bosamiz.

12. Natijada kodlar oynasi xosil bo'ladi. Oynaning chap yuqori qismidagi ro'yxatdan mnuFileNewForm va o'ng tarafdagi ro'yxatdan Click punktlari tanlanadi. Xosil bo'lgan proseduraga quyidagi kod kiritiladi:

```
Dim frmNewForm As New frmChildMDI  
frmNewForm.Show
```

YUqoridagi kodning vazifasi. New frmChildMDI - oynasidan yangi frmNewForm avlod oynasi yaratilmoqda va u Show uslubi yordamida vizuallashmoqda (ko'rinmoqda). Bu kod, dastur ishga yuklanib «Fayl» menyusidan «YAngi forma» buyrug'i tanlanganida ishlaydi.

13. Oynaning chap yuqori qismidagi ro'yxatdan mnuWindow va o'ng tarafdagi ro'yxatdan Click punktlari tanlanadi. Xosil bo'lgan proseduraga quyidagi kod kiritiladi:

```
mdiParentMDI.Arrange vbTileVertical
```

14. Kodlar oynasining eng yuqori qismiga chiqiladi va Enter tugmasi bosiladi. Xosil bo'lgan bo'sh satrga quyidagi kod kiritiladi:

```
Dim frmCount As Integer
```

15. Oynaning chap yuqori qismidagi ro'yxatdan MDIForm va o'ng tarafdagi ro'yxatdan Load punktlari tanlanadi. Xosil bo'lgan proseduraga quyidagi kod kiritiladi (ko'pchilik xollarda Load punkti avtomatik chiqadi):

```
frmChildMDI.Caption="1 -rakamli avlod formasi"  
frmCount=1
```

16. mnuFileNewForm_Click() prosedurasining kodi quyidagi kod bilan almashtiriladi:

```
Dim frmNewForm As New frmChildMDI  
frmCount=frmCount + 1  
frmNewForm.Caption=Str(frmCount) + " - rakamli avlod formasi"  
frmNewForm.Show
```

YUqoridagi amallar to'la bajarilganidan so'ng dasturni ishga tushirib ko'rishimiz mumkin. Dasturning to'la kodi quyidagicha:

```
Dim frmCount As Integer
```

```
Private Sub MDIForm_Load()  
    frmChildMDI.Caption="1 - rakamli avlod formasi"  
    frmCount=1  
End Sub
```

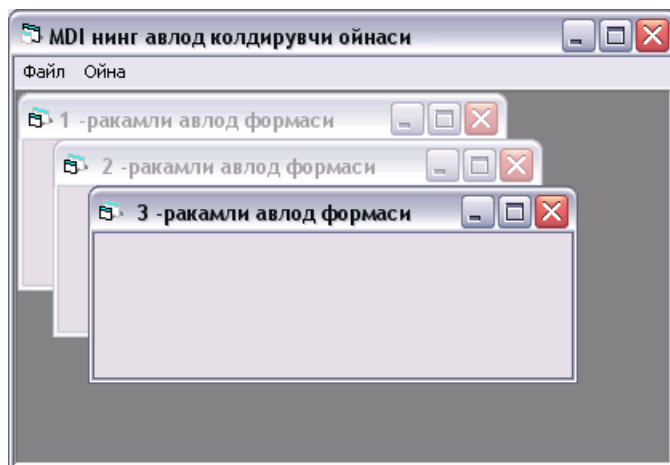
```
Private Sub mnuWindow_Click()  
    mdiParentMDI.Arrange vbTileVertical  
End Sub
```

```

Private Sub nmuFileNewForm_Click()
    Dim frmNewForm As New frmChildMDI
    frmCount=frmCount + 1
    frmNewForm.Caption=Str(frmCount) + " - rakamli avlod formasi"
    frmNewForm.Show
End Sub

```

Dastur ishlagandan keyin Asosiy oyna avlod oynalari bilan birgalikda quyidagi ko'rinishga ega bo'ladi:



Menyu

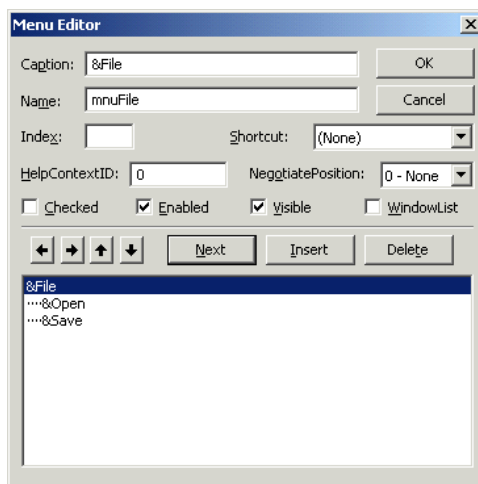
Bugungi kunda yaratilayotgan barcha dasturlar menyular yordamida boshqariladi. Sababi undan foydalanish qulay va osondir. Menyu ikki xil bo'lib, ular yuqori menyu va kontekst menyulardir.

Yuqori menyuning ham o'ziga tegishli xususiyatlari mavjud bo'lib ular quyidagi jadvalda keltirilgan:

Xususiyat	Vazifasi
Name	Menyuning nomi
Caption	Menyu sarlavxasi
Checked	Agar shu xususiyat ga TRUE qiymati berilsa menyuning sarlavxasi oldiga belgi qo'yiladi.
Enabled	Menyuning ishlashini ta'minlaydi. True qiymat qabul qilsa ishlaydi aks xolda ishlamaydi.
HelpContextID	Menyuga tegishli yordam konteksti saqlanadi.
Index	Menyu indeksi
NegotialePosition	Menyuning ekranda joylashishi
Shortcut	Menyu vazifasini tez bajarish uchun tugmachalardan foydalanish
Visible	Menyuni ekranda ko'rsatish yoki uning aksi
WindowList	Ochilgan oynalar sonini

Menyu yaratish

Barcha yuqori menyularni yaratishda Menu Editor muxarriridan foydalaniladi (15-rasm).



15-rasm. Menu Editor muloqot oynasi

Menyuni xosil qiluvchi muxarrirni quyidagi amallar yordamida chaqirish mumkin:

- Tools menyusidan Menu Editor buyrug'ini tanlash orqali;
- Menu Editor tugmasini bosish orqali ;
- Ctrl+E tugmachalarini teng bosish orqali.

Menu Editor oynasida yana bir gurux boshqaruv oynalari mavjud bo'lib, ular quyidagi imkoniyatlarga ega:

- ko'rsatkichi chapga qaragan  tugma ichki menyuni asosiy menyuga aylantiradi;
- ko'rsatkichi o'ngga qaragan  tugma asosiy menyuni ichki menyuga aylantiradi;
- ko'rsatkichi pastga va yuqoriga qaragan   tugmalar menyu punktlari o'rnini almashtiradi;
- Next(Keyingi) – Maxsus ko'rsatkichni keyingi punktga o'tkazadi;
- Insert(Joylash) – Menyu punktlari orasiga yangi punkt qo'shadi;
- Delete (O'chirish) – Tanlangan punktni o'chiradi.

Muloqot oynalari

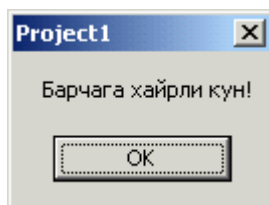
Visual Basic 6 da dasturchilar uchun tayyor xoldagi diolog oynalari mavjud bo'lib, ulardan vazifasiga qarab foydalanish mumkin. U oynalar quyidagilardir:

- Open (Ochish) – kerakli faylni izlab topish uchun;
- Save As (O'zgartirib saqlash) – saqlanadigan fayl joyini izlab topish uchun;
- Font (SHrift) – siftni tanlash uchun;
- Color (Rang) – rang tanlash uchun;
- Print (CHop etish) – chop etish turini o'rnatish uchun;
- Help (YOrdam) – Yordam sistemasini ishlatish uchun.

YUqoridagi oynalar bilan chuqurroq tanishamiz.

MsgBox xabarlar oynasi

MsgBox xabarlar oynasidan turli oraliq so'rovlari vaqtida hamda xar xil vaziyatlar xaqida xabar berishda foydalaniladi (16-rasm).



16-rasm. MsgBox xabarlar oynasi yordamida xosil qilingan matn.

MsgBox xabarlar oynasining to'la ta'rifi qo'yidagicha:

MsgBox (prompt[, buttons] [, title] [, helpfile, context])

bu yerda –

- prompt – Xabar matni. Matn ko'pi bilan 1024 belgini o'z ichiga oladi. Matinni ikkinchi bo'lagini pastga tushirish uchun Chr(13) qatori qo'shib yoziladi(...+Chr(13)+...);
- buttons – Xabarlar oynasida xosil bo'luvchi tugmacha turlarini va xabar belgilarini o'zida saqlaydi. Biz ular bilan quyida tanishamiz;
- title – Xabarlar oynasining sarlavxasi;
- helpfile – Yordam fayliga yo'lni o'zida saqlaydi;
- context - Yordam fayli tarkibidagi matnga yo'lni o'zida saqlaydi.

Immediate oynasiga quyidagi kodni kiriting va <Enter> tugmasini bosing:

MsgBox "Barchaga hayrli kun!", vbYesNo + vbExclamation, "Xabarlar oynasi"

Immediate oynasiga o'tish uchun View menyusidan Immediate Window buyrug'i tanlanadi yoki Ctrl+G tugmachalari teng bosiladi.

Visual Basic ning multimediali imkoniyatlari.

Multimedia – bu xar qanday axborot tashuvchilardagi video-audio axborotlar bilan kompyuter yordamida ishlashni ta'minlovchi texnologiyadir. Visual Basic da multimedia vositalarini boshqarish maxsus MCI (Multimedia Control Interface) interfeys yordamida amalga oshiriladi.

Visual Basic ko'p sonli multimedia qurilmalarini to'liq nazorat qila oladi. Multimediani boshqaruvchi muxitdan foydalanish uchun qo'llaniluvchi faylning formatini bilish zarur. Qurilmani boshqarish uchun fayl strukturasi bog'liq ravishda qurilma tipini va uning drayveri mavjud yoki mavjud emasligini bilish kerak.

Visual Basic dagi MCI multimediani boshqaruvchi interfeys deyarli barcha tipdagi multimedia formatli fayllarni boshqara oladi:

AVI	AVI (Audio Visual Interleaved) formatidagi videofayl
MPEG	Siqilgan formatdagi videotasvir
MID, RMI	MIDI (Musical Instrument Digital Interface) formatdagi ketma-ketlik
WAV	Ovozli fayllar

Visual Basic da multimediani boshqaruv muxitidan foydalanish uchun ayni paytda foydalanayotgan faylingizni tipini bilishingiz zarur. Multimediadagi axborot fayl

strukturasiga bog'liq ravishda treklar ko'rinishida strukturalanadi. Masalan, musiqali kompakt diskda trek bu aloxida musiqa bo'lib xisoblanadi va tartib bo'yicha raqamlanadi.

Visual Basic da multimedia imkoniyatlaridan foydalanish uchun MCI interfeysi qo'llaniladi. Bu interfeysning asosiy komponenti – MMControl boshqaruv elementi bo'lib, u multimediani boshqaruvchi va uni to'liq nazorat qiluvchi xususiyatlar va buyruqlar to'plamidan iboratdir.

MCI dan dasturda foydalanish uchun proektga Microsoft Multimedia Control 6.0 kutubxonasini ulash kerak. Buning uchun Project (Proekt) menyusidan Components (Komponentlar) buyrug'i tanlanadi va xosil bo'lgan Components muloqot oynasidan Microsoft Multimedia Control 6.0 kutubxonasiga belgi qo'yilishi va Ok tugmasi bosilishi yetarlidir. Ushbu kutubxona ulanganidan keyin boshqaruv elementlari panelida MMControl tugmasi xosil bo'lganini ko'ramiz. MMControl multimedia boshqaruv jlementi to'qqizta tugmadan iborat bo'lgan panel bo'lib, ular quyidagi jadvalda keltirilgan:

Tugma	Nomlanishi	Vazifasi
	Prev (avvalgi)	YOzuv (fayl)ning boshiga o'tish
	Next (navbatdagi)	YOzuv (fayl)ning oxiriga o'tish
	Play (ijro etish)	YOzuvni ijro ettirish
	Pause (tanaffus)	Ijro yoki yozuvni to'xtatib turish
	Back (orqaga qadam)	Bir qadam (kadr) orqaga o'tish
	Step (oldinga qadam)	Bir qadam (kadr) oldinga o'tish
	Stop (stop)	Ijro yoki yozuvni to'xtatish
	Record (yozish)	Ma'lumot yozish
	Eject (bo'shatish)	Ma'lumot tashuvchini qurilmadan olish

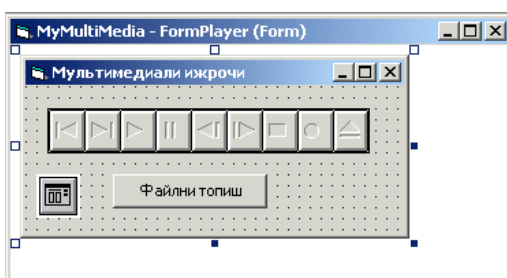
MMControl boshqaruv elementi multimediani boshqarish imkoniyatini beruvchi bir nechta xususiyatlarga ega:

Xususiyat	vazifasi
AutoEnabled	Qurilmani ishga tayyorgarligi vaqtida undan barcha tugmalarni avtomatik tarzdi ishchi xolatga qo'yadi
CanEject	Agar ma'lumot tashuvchi qurilmada bo'lmasa True qiymatga ega, aks xolda False
CanPlay	Agar qurilma ijro etishga tayyor bo'lsa True qiymatga ega, aks xolda False
CanRecord	Agar qurilma yozishga tayyor bo'lsa True qiymatga ega, aks xolda False
DeviceType	Qurilmani o'rnatish
Error	MCI ning oxirgi xatoligini aniqlash
ErrorMessage	Oxirgi xatolik xaqida xabar berish
FileName	Ijro uchun faylni o'rnatish
Frames	Step buyrug'i yordamida o'tkazib yuboriluvchi qadamlar soni beriladi
HWndDisplay	CHizish oynasining deskriptorini belgilaydi
Length	Ma'lumotlar uzunligini aniqlash

Mode	Qurilmaning joriy ish rejimini aniqlash
Position	TimeFormat xususiyatiga bog'liq ravishda qurilmadagi joriy pozitsiyani aniqlaydi
RecordMode	Qurilmaning joriy yozuv rejimini aniqlash
Silent	Ovozni uzish
Start	Ma'lumot tashuvchining boshlang'ich pozitsiyasiga o'rnatish
TimeFormat	Vaqt formatini aniqlash
To	Ma'lumot tashuvchining oxirgi pozitsiyasiga o'rnatish

Mci boshqaruv elementi yordamida audio fayllar ijrochisi dasturini yaratamiz. Bunday ijrochini yaratish uchun quyidagi ketma-ketlikni bajarish kerak bo'ladi:

1. YAngi standart forma yaratiladi.
2. Proektni MyMultiMedia deb nomlaymiz. Buning uchun Project menyusidan Project1 Properties tanlanadi. Xosil bo'lgan muloqot oynasidan Project Name ga MyMultiMedia deb yoziladi.
3. Formaning nomini esa Properties xususiyat oynasidan Name xususiyatiga FormPlayer, Caption xususiyatiga esa Multimediali ijrochi deb yozamiz.
4. Components muloqot oynasi yordamida formaga Microsoft Multimedia Control 6.0 komponentini formaga o'rnatamiz. Bu muloqot oyna Project menyusidan Components buyrug'ini tanlash bilan xosil qilinadi. Boshqaruv panelida xosil bo'lgan MMControl komponenti ustida sichqon tugmasi ikki marta tez bosilishi bilan, komponent formada xosil qilinadi. Ushbu komponentning nomini MMCntrolCDPlayer deb o'zgartiramiz.
5. Ushbu dastur yordamida kerakli biror ovozli faylni ochish imkoniyati xam bo'lishi talab etiladi. SHuning uchun bizga qo'shimcha komponent, ya'ni faylni ochish uchun muloqot oyna zarur. Components muloqot oynasi yordamida Microsoft Common Dialog Control 6.0 komponenti tanlanadi va boshqaruv panelidan CommonDialog komponenti ustida sichqon tugmasi ikki marta tez bosilishi bilan komponent formada xosil qilinadi. Bu komponentning nomini cdPlayer deb o'zgartiramiz.
6. Formaga bitta CommandButton tugma o'rnatiladi. Tugmaning nomini cdFindFile, Caption xususiyatini esa Faylni topish deb o'zgartiramiz. Dasturimizning forma ko'rinishi 23-rasmda shaklini oladi.



22-rasm. Dasturning forma ko'rinishi

7. Kodlar oynasini ochamiz va quyidagi kodlarni kiritamiz:

```
Private Sub Form_Load()
    MMControlCDPlayer.Notify=False
    MMControlCDPlayer.Wait=True
    MMControlCDPlayer.Shareable=False
    MMControlCDPlayer.DeviceType="WaveAudio"
End Sub
Private Sub cdFindFile_Click()
    cdPlayer.ShowOpen
    MMControlCDPlayer.FileName=cdPlayer.FileName
```

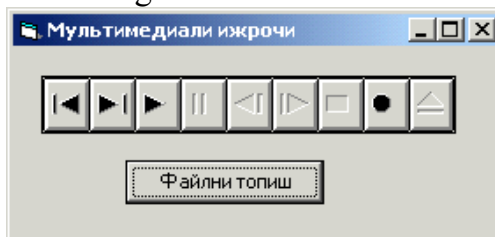


```

MMControlCDPlayer.Command="open"
End Sub
Private Sub Form_Unload(Cancel As Integer)
    MMControlCDPlayer.Command="Close"
End Sub

```

Dasturni ishga tushiramiz. Faylni topish tugmasini bosib, kerak faylni tanlaymiz. Tanlangan faylni ijro ettirish MMControlCDPlayer elementi yordamida amalga oshiriladi. 24-rasmda dastur ishchi xolatdagi aks ettirilgan.



24-rasm. Dasturning ishchi xolati

SHuningdek, videofayllar ijrochisi xam shunga o'xshash yaratiladi. Faqat qurilma tipining boshqa biri tanlanishi lozim. YA'ni, yuqoridagi kodga quyidagicha o'zgartirish kiritamiz:

```

Private Sub Form_Load()
    MMControlCDPlayer.Notify=False
    MMControlCDPlayer.Wait=True
    MMControlCDPlayer.Shareable=False
    MMControlCDPlayer.DeviceType="AVIVideo"
End Sub
Private Sub cdFindFile_Click()
    cdPlayer.ShowOpen
    MMControlCDPlayer.FileName=cdPlayer.FileName
    MMControlCDPlayer.Command="open"
End Sub
Private Sub Form_Unload(Cancel As Integer)
    MMControlCDPlayer.Command="Close"
End Sub

```

Ushbu dastur yordamida AVI formatidagi videofayllarni ijro ettirish mumkin. Bunday xolatda tasvir ko'rish oynasida xosil bo'ladi. Albatta, tasvirni ko'rish oynasida chiqishi noqulayliklarni yuzaga keltiradi. Tasvirni formaning belgilangan soxasida xam xosil qlishi mumkin. Buning uchun quyidagi ketma-ketliklarni amalga oshirish kerak:

1. Formaga Picture boshqaruv elementini o'rnatamiz va uning nomini picView deb o'zgartiramiz.

2. YUqoridagi kodni quyidagicha o'zgartiramiz:

```

Private Sub Form_Load()
    MMControlCDPlayer.Notify=False
    MMControlCDPlayer.Wait=True
    MMControlCDPlayer.Shareable=False
    MMControlCDPlayer.DeviceType="WaveAudio"
    MMControlCDPlayer.DeviceType="AVIVideo"
End Sub
Private Sub cdFindFile_Click()

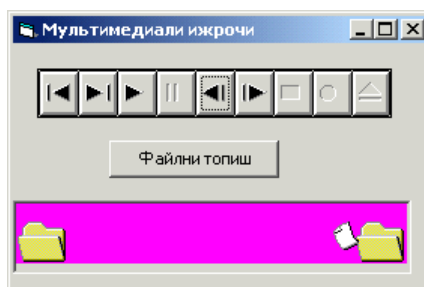
```

```

cdPlayer.ShowOpen
MMControlCDPlayer.FileName=cdPlayer.FileName
MMControlCDPlayer.Command="open"
' Tasvir chiquvchi oynani belgilash
MMControlCDPlayer.hWndDisplay=picView.hwnd
FormPlayer.picView.SetFocus
End Sub
Private Sub Form_Unload(Cancel As Integer)
    MMControlCDPlayer.Command="Close"
End Sub

```

Dasturning ishchi holatidagi formasining ko'rinishi 25-rasmda ko'rsatilgan.



25-rasm. Dastur formasining ishchi xolatidagi ko'rinishi

Foydalanilgan adabiyotlar ro'yxati

1. А. Ануньев, А. Федоров. «Самоучитель Visual Basic 6.0». «ВНВ-Санкт-Петербург» 2001 г. Уч.пос.
2. Тим Андерсон. Visual Basic. Қадам ба қадам. Тошкент “Ўзбекистон”-2002.
3. Марков Б. Визуальная программирования, Москва, 2003, Уч.пос.
4. Очков и др 128 советов начинающему программисту, Москва, 1991, Уч.пос.
5. VB NET Complete Publication, NewDehli, 2002, Уч.пос.
6. Гринзоу Лу. Философия программирования для WINDOWS 95-NT Символ-Плюс, 2002., Уч.пос.

Mundarija

Mavzu. Kirish. Visual Basic ni o'rnatish va ishlatish Visual Basic muhiti	3
Visual Basic 6.0 ni ishga tushirish	3
Integrallashgan ishlab-chiqarish muhiti	4
Asosiy menyu.....	5
Mavzu. Visual Basicda boshlang'ich amallar va proektlar. Dastur matnining umumiy ko'rinishi.	8
Standart asboblarni paneli	8
Mavzu. Vizual Basic dasturlash tili elementlari. O'zgaruvchilar	13
O'zgaruvchilarni e'lon qilish	15
O'zgaruvchiga qiymatlarni o'zlashtirish.....	17
Variant turidagi qiymatlarning ichki tasavvuri	17
Doimiylar	18
Doimiylarni e'lon qilish	18
O'zgaruvchilarning tavsifi	18
3. Kommentariyalar.....	19
Operatorni bir nechta qatarga joylashtirish.....	20
Bir nechta operatorlarni bir qatarga joylashtirish	20
Kodni tag'rirlash.....	20
If, Go To, Select boshqaruv strukturalari	23
3.9. Select Case shartli strukturalari	25
Siklli strukturalar.....	27
Takrorlanishlar soni berilgan sikl (hisoblagichli sikl) For ... Next	27
Sharti oldinda va oxirida berilgan sikllar	29
Sharti oldinda berilgan Do While siklining sintaksisi.....	29
Sharti oxirida berilgan sikl	29
Sharti oldinda berilgan sikl	30
Sharti oxirida berilgan sikl.....	30
Loop Until <i>Shart</i>	30
Belgi kattaliklar. Satrlar bilan ishlash. Tip o'zgartiruvchi operatorlar.	32
Massivlarni elon qilish. Bir va ko'p o'lchovli massivlar.	34
Mavzu. Visual Basic da prosedura va funksiyalar.	37
Prosedura-funksiyani e'lon qilish. Funksiyaga murojaat qilish va undan foydalanish.....	41
Mavzu. Fayllar bilan ishlash. Faylni yozish va o'qish uchun ochish. Fayl ishini yakunlash. Diskdagi fayl qanday o'qiladi?.....	44
Mavzu. Visual Basic ning grafik imkoniyatlari. Geometrik figuralarni xosil qilish. Xarakatli tasvirlarni xosil qilish.....	47
Mavzu. Foydalanuvchi interfeysini qayta ishlash. Interfeys tiplari, elementlari (menyu, qurilmalar paneli, muloqot oynalari), CommonDialog	53
Foydalanilgan adabiyotlar ro'yxati	66

