O’ZBEKISTON RESPUBLIKASI OLIY VA O’RTA MAXSUS TA’LIM VAZIRLIGI

BERDAQ NOMIDAGI QORAQALPOQ DAVLAT UNIVERSITETI

FIZIKA – MATEMATIKA FAKULTETI

«Amaliy matematika» kafedrasi

«Amaliy matematika va informatika » bakalavr ta’lim yo’nalishi IV – kurs bitiruvchisi Tursunboyev Azamatning

BITIRUV MALAKAVIY ISHI

Mavzu: C# tilida test sinovlari yordamida talabalar bilimini baholashning dasturiy ta’minotini yaratish

Ilmiy rahbar: **dots. Eshmuratov Sh.A**

Kafedra mudiri: **dots. Berdimuratov M**

Nukus-2019

MUNDARIJA

KIRISH

Hozirgi kunda maktablarga, kasb-hunar kollejlarda, akademik litseylarda, oliy o’quv yurtlarida o’quvchilarga va talabalarning bilimini baholash eng ahamiyatli masalalarning biri bo’lib keladi. Sababi har bir o’quvchi, talaba yoshlar o’z bilimlarini haqqoniy baholanishini, o’z bilimiga loyiq baho olishini hohlaydi.

Bugungi kunga kelib ularning bilimlarini baholashning bir necha usullari bor. Ularga yozma turda baholash, og’zaki turda, “Insert” usuli yordamida, “BBB sxemasi”, “T sxemasi”, test usuli va h.k kiradi. Har bir baholash turining o’ziga xos kriteryalari bor. Eng ko’p tarqalgan baholash turi test usuli bo’lib hisoblanadi. Bu usulning boshqa usullarga solishtirganda bir qancha qulayli tomonlari bor. Eng asosiy qulayli tomonlaridan biri quyidagilardan iborat:

- ✓ O’qituvchi bir vaqtning o’zida bir qancha bolalarning bilimlarini tekshiradi;
- ✓ O’qituvchi o’quvchidan o’tilgan mavzularning deyarli barchasi bo’yicha bilimini tekshirishi mumkin;
- ✓ O’qituvchining savollariga to’liq javob bergani yoki javob bermagani noaniq bo’lmaydi, to’g’ri yoki xato ekani aniq bo’ladi;
- ✓ Bularning barchasini belgili bir qisqa vaqt davomida amalga oshiriladi;

Demak ko’rinib turibdi bu test usulidan foydalanganda eng asosiysi bu vaqtdan yutishga bolar ekan. Shundan, yuqori o’quv o’rinlarida kirish imtihonlarni shu test usulidan foydalaniladi.

Lekin test usulining yuqorida aytilgan qulayli tamonlaridan boshqa kamchiliklari ham bor. Masalana, bir nechta o’quvchilardan test usulidan foydalanib bilimini baholash kerak bo’lsin. Birinchi navbatda test savollari har bir o’quvchi uchun varaqga bosmadan chiqarish kerak. Bu o’qituvchidan sarf xarajatni talab qiladi. Ya’ni test savollari bir nechta variontlarni ishlash kerak. Bu o’qituvchi uchun qo’shimcha ishni talab qiladi. Eng oxirgisi ularning belgilagan javoblarini tekshirish kerak. Bu qiyin emas agarda o’quvchilar soni oz bo’lsa, agarda ko’p bo’lsa o’qituvchining ko’p vaqtini olishi mumkin.

Albatta, hozirgi davr kompyuterlar, zamonaviy texnologiyalar asri. Qo’lda ishlanadigan ishlar, hisoblashlar, EHM larda amalga oshirish imkoniyati bor. Ular odamlar ishini avtomatlashtiradi. Bu bitiruv malakaviy ishining maqsadi, yuqoridagi test usulining kamchiliklaridan qutilish, ishlanadigan ish avtomatlashtirish, vaqtni ya’nada unimlash uchun “test dastur” ni yaratish bolib topiladi. Bu test dastur quyidagicha afzalliklarga ega bolishi kerak:

- ✓ Test savollarin qag’ozga chiqarish shart bo’lmasligi kerak;
- ✓ O’quvchilar o’zlarining baholarini shu vaqtda bilib olishi kerak;
- ✓ Test savollarini har bir o’quvchi uchun variant ishlashi va shart bo’lmasligi kerak;
- ✓ Vaqt unumli bo’lishi kerak;
- ✓ O’qituvchi ishini avtomatlashtirishi kerak;

Faqat talab qilingan narsa bu o’qish joylarining kompyuterlar bilan ta’minlanganligi. Albatta, hozirgi kunga kelib o’quv joylari kompyuterlar zamonaviy texnologiyalar bilan ta’minlanayabdi.

Shu bitiruv malakaviy ishi test dasturni yaratish, uning tuzilishi haqida to’xtalib o’tamiz. Bitiruv malakaviy ishi kirish, to’rt paragraf, xulosa, foydalanilgan adabiyotlar ro’yxati va ilovalardan iborat.

Bitiruv malakaviy ishi birinchi paragrafida, C# tilining paydo bo’lishi, rivojlanish tarixi, uning Dot.Net bilan bog’liqligi haqida keltirib o’tiladi. Ikkinchi paragrafida, Visual C# boshqarish elementlari, ularning bajaradigan vazifalari, xossalari va jarayonlari aytilib o’tiladi. Uchinchi paragrafida, TEST_RTF dasturiy ta’minotining asosiy algoritmining strukturasi, modeli haqida aytiladi. To’rtinchi paragrifida dasturiy ta’minotidan qanday foydalanish kerakligi, foydalanish ko’rsatmalari keltriladi. Xulosada bo’lsa test dasturning yutuqlari haqida aytiladi.

1-§. C# tili haqida tushunchalar

Har bir dasturni yaratishda dastlab algoritm tuziladi va shu algoritm bo'yicha dasturlash tillarining birida dastur yaratiladi. Men shu bitiruv malakaviy ishida test dastur tuzishda C# tilidan foydalandim.

Shundan, malakaviy ishini chuqur tushunish uchun C# tili haqida uning yaratilish tarixi haqida to'xtab o'tamiz.

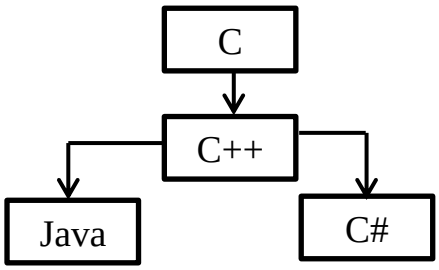
Dasturchilar ko'pincha dasturning effektivligi va qo'llanishda uning tez ishlashini orttirish uchun har xil usullarni izlaydi. Ayirim holatlarda biz dastur ishlashda ko'proq instrumentlarni talab qilamiz, ya'ni dasturlashtirish tillar bilan ishlagan vaqtida. Tilining effektivligi uning tezligi va shuning bilan bir vaqtida o'rinliligini (qulayliligini) bog'liqli bo'ladi. Tilning sintaksisi qisqa tushunarli bo'lishi kerak. U korrekt kodini yarata olishi bilan keng imkoniyatlarga ega bo'lishi kerak. Shunday tillarning biri C# tildir.

Dot.Net muhiti qo'llab quvatlaydigan Microsoft kompaniyasida yaratilgan C# tili dastur boy merosga tayanadi. Uning bosh arxitektori shu sohaning yetakchi mutaxasisi – Anders Xeylsberg [3] dir.

C# - dunyodagi ikki eng yirik C va C++ kompyuter tillarining avlodi bo'lib hisoblanadi. U C tilining sintaksis, kalit so'zlarni va operatorlarini me'ros qilib olgan. U C++ tilinda aniqlangan obekt modelini ko'rish va takomillashtirish imkoniyatiga ega. C# boshqa Java tili bilan ham yaqindan bog'liqlikda. C# va Java umumiy kelib chiqishiga ega bo'lganligi bilan, ularning ko'pchilik ahmiyatli tomonlari bir-biridan farq qilib, ularni "qarindosh – ukalar" deb atashga bo'ladi.

C# 1990 – yil oxirlarida yaratilgan bo'lib, Microsoft Net – strategiyasining bir bo'lagi bo'ldi. Eng birinchi bo'lib 2000 – yilni o'rtalarinda a-versiya sifatida chiqdi. C# ning bosh arxitektori dasturlash sohasini butun dunyo tan olgan Anders Xeylsberg (Anders Hejlsberg) hisoblanadi.[3]

C# tilining genealogik manbasin quyidagi rasmdan ko'rishingizga bo'ladi:



C# ning Dot.Net qobig'i bilan bog'liqligi. C# o'ziga yetarli darajada kompyuter tili bo'lgani bilan, u Dot.Net muhiti bilan o'zaro aloqada bo'ladi. Buning ikki sababi bor. Birinchidan, C# dastlab Microsoft kompaniyasi tomonidan Dot.Net muhitida bajariluvchi kodni tuzish uchun ishlab chiqarilgan. Ikkinchidan, bu muhiti C# tilida foydalaniladigan kutubxona aniqlangan. Shundan, C# tilida tuzilgan dastur Windows operatsion tizimida to'liq ishlashi uchun shu kompyuterda Dot.Net o'rnatilgan bo'lishi kerak. Hozirgi kunda C# ning ham Dot.Netning ham bir necha versiyalari bor. Ular quyidagi jadvalda ko'rsatilgan:

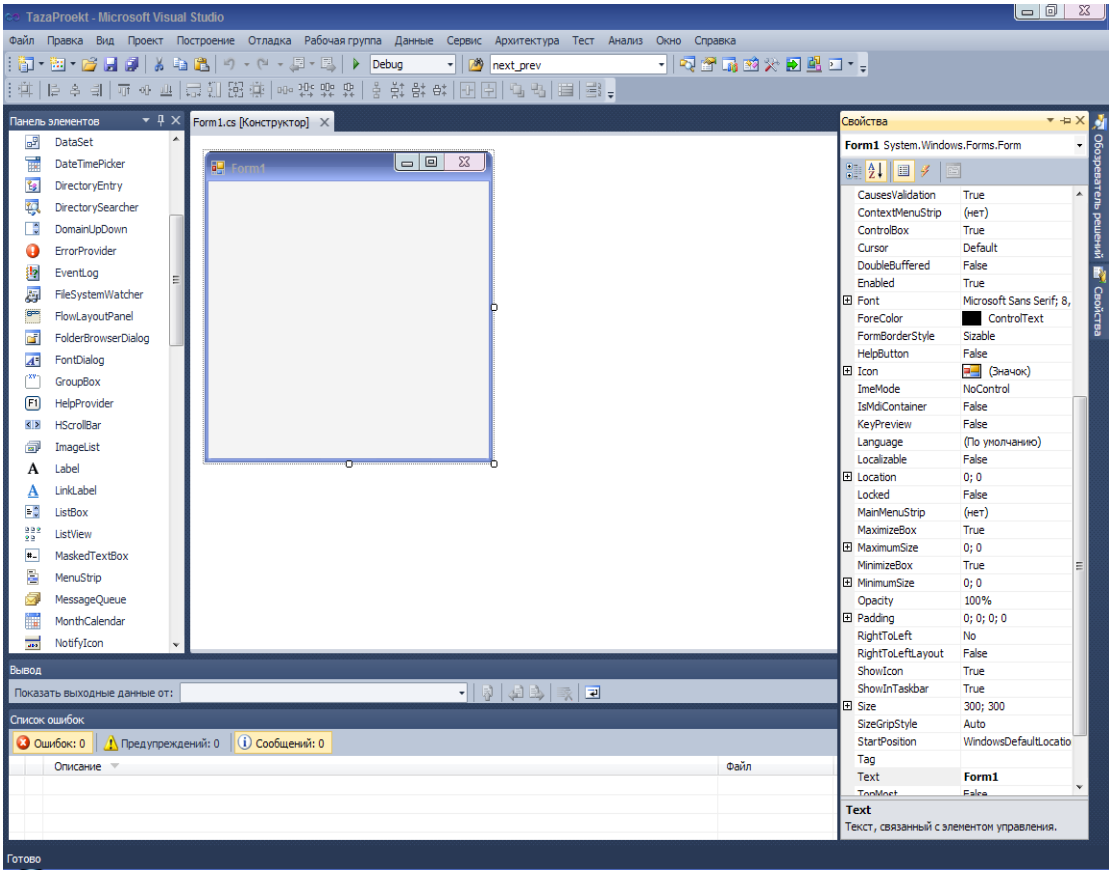
Versiya	**Spetsifikatsiya**			**Vaqt**	**Dot.Net**	**Visual Studio**
	ECMA	ISO/IEC	Microsoft			
C# 1.0	Dekabr 2002	Aprel 2003	Yanvar 2002	Yanvar 2002	Dot.Net 1.0	Visual Studio .NET 2002
C# 1.2			oktyabr 2003	Aprel 2003	Dot.Net 1.1	Visual Studio .NET 2003
C# 2.0	Iyun 2006	sentyabr 2006	sentyabr 2005	Noyabr 2005	Dot.Net 2.0	Visual Studio 2005
C# 3.0	Ko'rsatilmagan		Avgust 2007	Noyabr 2007	Dot.Net 3.5	Visual Studio 2008
C# 4.0			Aprel 2010	Aprel 2010	Dot.Net 4	Visual Studio 2010
C# 5.0				Avgust 2012	Dot.Net 4.5	Visual Studio 2012

Jadvalda Visual Studiolarning versiyalarining ro'yxati ko'rsatilgan. Visual Studio bir necha tillarni o'zida jamlagan dastur bo'lib topiladi. [10] uning ichida C++, C#, Visual C#, Visual Basic kabi tillar mavjud. Men shu bitiruv malakaviy ishni

yozishda Visual Studio 2010 ning ichidagi Visual C# dan foydalandim. Sababi, C# dan ko'pincha konsol dasturlarni yaratishda foydalanladi, Visual C# dan bo'lsa interfeysli dasturlar yaratishda foydalanadi. Test dasturda bo'lsa forma ya'ni foydalanuvchilar uchun qulayli interfeys yaratish kerak. Keyingi paragrafda Visual C# bilan qanaqa qilib ishlash kerak ekanligi, uning komponentalari, ularning xossalari va hodisalari bilan tanishtirib o'tamiz.

2-§. Visual C# komponentalari va ularning xossalari

Visual C# da eng dastlab dastur tuzishni boshlaganda eng dastlab Visual Studioni ochamiz hamda "**Создать проект…**" ni bosamiz va proektga nom berib "**OK**" tugmasini bosganda Visual C# ish oynasi ochiladi(2.1-rasm).



2.1-rasm

Undan so'ng F5 klavishasini bosish orqali bajaramiz. Shunda **.exe** fayli ya'ni dastur yaratiladi. Bu dastur hali hech qanday ishni bajarmaydi. Keyingi bosqichda formaning xossalarini o'rnatamiz. Formaning xossalari quyidagi jadvalda keltirilgan.

Xossasi	Formaning xossalari

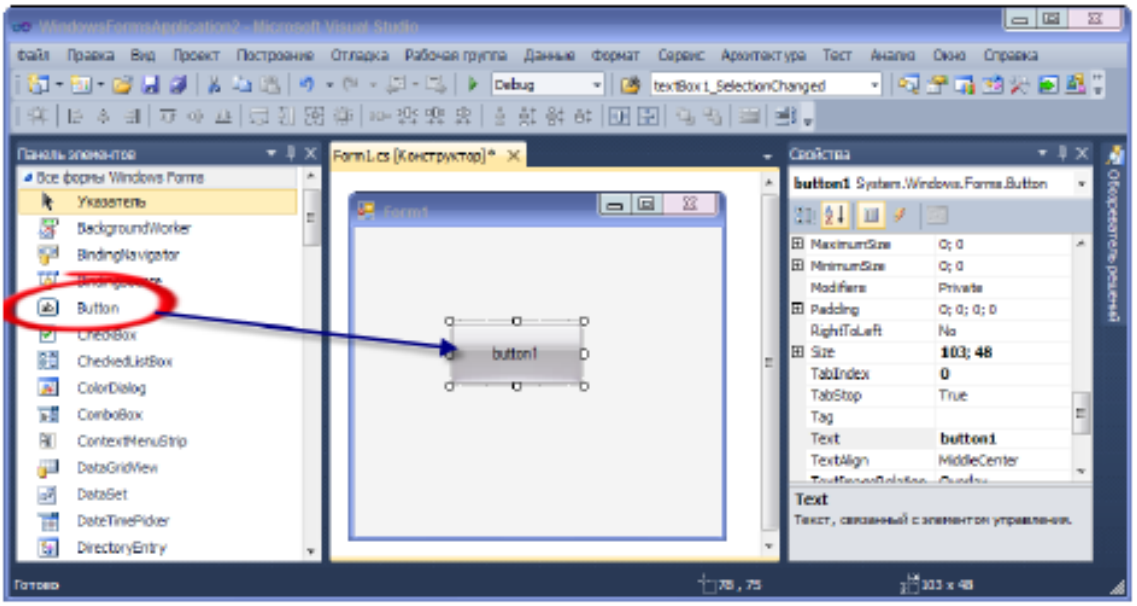
Name	Formaning nomi.
Text	Sarlavhalar satridagi matn.
Size	Forma o'lchamlari. Ya'ni **width** formaning eni, **height** bo'lsa balandligini belgilab beradi.
StartPosition	Forma birinchi marta bajarilganda uning ekrandagi o'rni. Forma ekran o'rtasida chiqishi mumkin (**CenterScreen**). Agar bu xossa **Manual** qiymat qabul qilgan bo'lsa, unda formaning joylashadigan o'rnini **Location** xossasi orqali beriladi.
Location	Formaning ekrandagi joylashgan o'rni. Formaning yuqori tarafidan ekranning yuqori tarafigacha bo'lgan masofa Y, formaning chap tarafidan ekranning chap tarafigacha bo'lgan masofa X qiymatlarida beriladi.
FormBorderStyle	Forma turi(chegaralari). Forma oddiy oyna(**Sizable**). Fikserlangan o'lchamdagi oyna(**FixedSingle, Fixed3D**), dialog oynasi(**FixedDialog**) hech qanday tugmalarsiz (**SizableToolWindow, FixedToolWindow**). Agar bu xossa **None** qiymatini qabul qilgan bo'lsa unda formada sarlavhalar satri ham chegaralari bo'lmaydi.
ControlBox	Bu xossa sistemali menyu tugmalarning ko'rinishini boshqaradi. Agar bu xossa **False** qiymatini qabul qilgan bo'lsa, unda sarlavhalar satridagi yig'ish, ekranni kichiklashtirish, chiqish tugmalari ko'rinmaydi.
MaximazeBox	Bu formani kattalashtirish va kichiklashtirish tugmasi. Agar bu xossaning qiymati **False** bo'lsa unda shu tugma bosilmaydi.
MinimazeBox	Bu formani yig'ish tugmasi. Agar bu xosiyat qiymati **False** bo'lsa shu tugma bosilmaydi.
Icon	Sarlavhalar satridagi nishon.
Font	Shift u formadagi barcha komponentlarning shriftini belgilab beradi, agar ularning har biri shrift o'zgartirilmagan bo'lsa.
ForeColor	Rang, u ham shu formadagi barcha komponentalarning matinning rangini avtomat turda o'zgartiradi agarda ularning har birini **ForeColor** xossasi aniq turda ko'rsatilmagan bo'lsa.
BackColor	Fon rangi.
Opacity	Formaning ko'rinish darajasi. Agar bu xossaning qiymati 100% bo'lsa unda forma to'lig'i bilan ko'rinib turadi, kamayib borgan sari forma pastida ko'rina boshlaydi, 0%

	bo’lsa formaning o’zi ko’rinmay qoladi. [6]

Button. “**Button**” bu buyuruq bajaruvchi tugma hisoblanadi. Foydalanuvchi dasturdan foydalanayotganda button ustida ishlaydigan amal faqat uni bosish bo’lib topiladi. Dastur yaratish vaqtida biz “**Button**” dan foydalanadigan bo’lsak uni kerakli o’ringa olib borib qo’yamiz va asosiy bajaradigan vazifasini kiritamiz. “**Button**” ning asosiy bajaradigan vazifasi “**Click**” vazifasi ichida yoziladi. Dastur kodi holatida u quyidagicha bo’ladi.

Private void button1_Click(object sender, EventArgs e) {}

“**Button**” ni formaga joylashtirish 2.2- rasmda ko’rsatilgan.



2.2-rasm

Bizning dasturmizda “**Button**”ni ko’p foydalanamiz. Umumiy hohlagan dastur holatida “**Button**”dan ko’p va juda keng foydalanadi. Endi “**Button**” boshqarish elementlarining xossalari bilan tanishib chiqamiz.

Xossasi	Bajaradigan vazifalari
Name	Boshqarish elementi nomi.
Text	Tugmadagi matn.
TextAlign	Matnning tugmada joylashishi. Matn tugma o’rtasida joylashishi mumkin (**MiddleCenter**), chap tarafida (**MiddleLeft**), o’ng tarafida. (**MiddleRight**)
FlatStyle	Stiyl. Tugma standart bo’lishi mumkin (standart), tekis (**Flat**) yoki Popup.
Loacation	Tugmaning boshqarish elementi ustida joylashgan o’rni. X

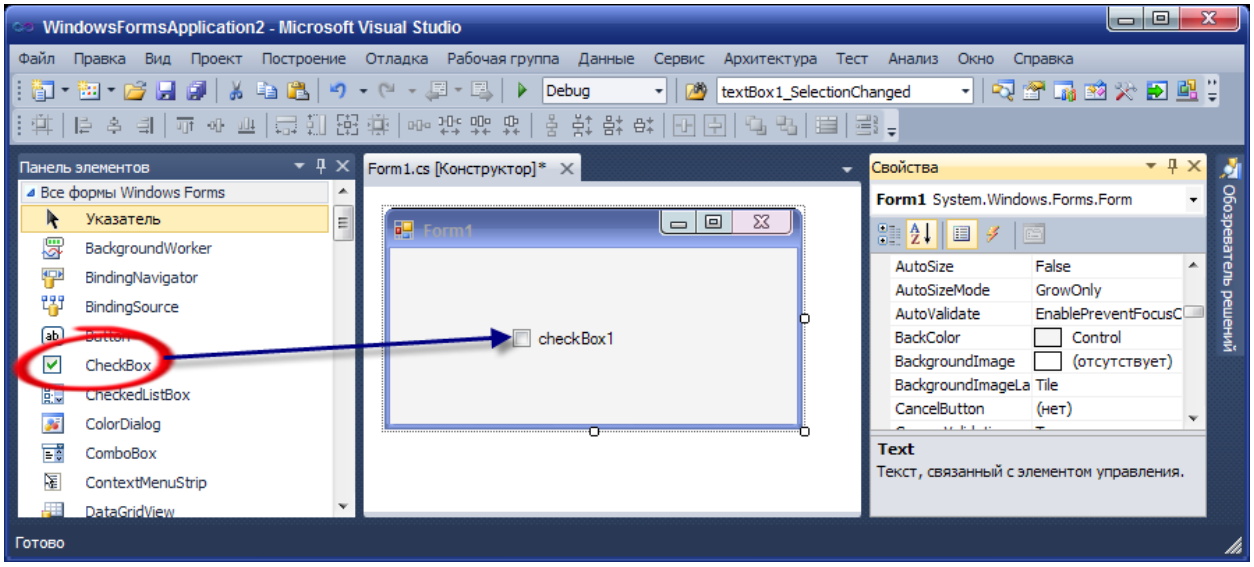
	tugmaning chap tarafi chegarasidan formaning chegarasigacha bo'lgan masofa, Y bo'lsa tugmaning yuqorigi chegarasidan formaning yuqori chegarasigacha bo'lgan masofa.
Size	Tugma o'lchami.
Font	Shrift, tugmadagi yozilgan matn shrift.
ForeColor	Tugmadagi matn rangi.
Enabled	Tugmaga ruxsat. Tugma bosiladi agar bu xossa qiymati **True** bo'lsa, bosilmaydi agar **False** bo'lsa.
Visible	Tugmani yashirish imkoniyatini beradi. Agar **True** bo'lsa tugma ko'rinada, **False** bo'lsa yashiradi.
Cursor	Sichqoncha ko'rsatkichining tugmaga borgandagi turini aniqlab beradi.
Image	Bu tugma ustida rasm qo'yadi.
ImageAlign	Rasmning tugmadagi joylashgan o'rnini aniqlab beradi. Rasm tugma o'rtasida joylashishi mumkin (**MiddleCenter**), chap tarafida (**MiddleLeft**), o'ng tarafida (**MiddleRight**). Bundan boshqa rasmni tugmada joylashtirish mumkin (**TopLeft**, **TopCenter, TopRight, BottomLeft, BottomCenter, BottomRight**) .
ImageList	Bu rasmlar yig'indisi bo'lib shularning ichidan rasm qo'yishga bo'ladi. Uning uchun formaga **ImageList** komponentasi qo'shilgan bo'lishi kerak.
ImageIndex	Tugmada chiqadigan **ImageList** rasmlar yig'indisidan olingan rasm nomeri (indeksi)
ToolTip	Tugmaga borganda sichqoncha ko'rsatkichi yonida hosil bo'ladigan yordamchi. Bu xossasi qo'llanish uchun formaga **ToolTip** komponentasi qo'shilgan bo'lishi kerak. [6]

CheckBox. CheckBox boshqarish elementi ko'pincha qoldirilgan vazifani bajaradi ya'ni uni bosishdan birdan qandayda bir buyuruq bajarilishi shart emas. Bu ko'pincha foydalanuvchi dasturni o'ziga qulayli qilib sozlayotgan (nastroyka), yoki kerakli instrumentlarni olganda qo'llaniladi. Ya'ni o'ziga kerak bo'lganlariga bayroqcha qo'yib olishi mumkin. **CheckBox** ning "**CheckState**" xossasida 3 holat bo'lishi mumkin: belgilanmagan (**Unchecked**), belgilangan (**Checked**) va aniq emas (**Indeterminate**). Bular "**ThreeState**" xossasi "**True**" qiymatini qabul qilganda bo'ladi, agar "**False**" qiymatini qabul qilsa unda faqat 2 holat (belgilangan va

belgilanmagan) bo’ladi. “**CheckState**” xossasi belgilangan (**Checked**) va aniq emas (**Indeterminate**) qiymatlari qabul qilsa unda “**Checked**” xossasi avtomat “**True**” qiymatini qabul qiladi, belgilanmagan (**Unchecked**) bo’lsa, unda “**Checked**” xossasi “**False**” qabul qiladi. Buni dastur holatinda ko’rishimizga bo’ladi:

```
checkBox1.CheckState = System.Windows.Forms.CheckState.Indeterminate;
checkBox1.CheckState =System.Windows.Forms.CheckState.Checked;
checkBox1.Checked = true;
checkBox1.CheckState = System.Windows.Forms.CheckState.Unchecked;
checkBox1.Checked = false;
```

2.3-rasmda formaga joylashtirilgan “**CheckBox**” va uning xossalarini ko’rishingizga bo’ladi.



2.3-rasm

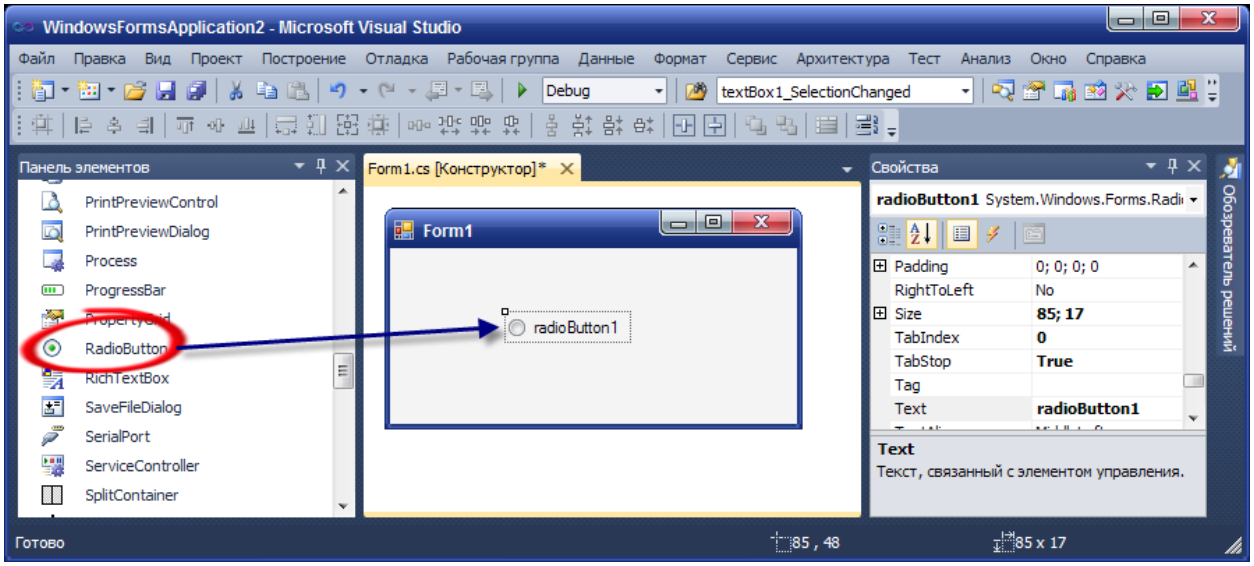
CheckBox boshqarish elementini dasturning qanday joylarinida foydalanadiganligi haqida aytib o’tdik. Bizning dasturimizga ham u sozlash (nastroyka) bo’limi uchun kerak. Ya’ni matn topshirish vaqtinda to’g’ri, noto’g’ri javoblar sonini, savollarni o’tkazuvchi tugmani qo’shish yoki olib tashlash va shunga o’xshagan bir nechta joylarni foydalanamiz. **CheckBox** boshqarish elementining xossalari quyidagi jadvalda ko’rsatilgan.

Xossasi	Bajaradigan vazifasi
Text	Tugma yonida yoziladigan matn
Checked	Bu uning “**CheckBox**” ning holati. Agar u “**True**” bo’lsa unda

	bayroqcha belgisi qo'yiladi, agar "**False**" bo'lsa hech narsa qo'yilmaydi
TextAllign	Unda matn holati. Matn o'rtada joylashishi mumkin (**MiddleCenter**), chap tarafda (**MiddleLeft**), o'ng tarafda (**MiddleRight**). Bundan boshqada bir nechta turlari bor (**TopLeft, TopCenter, TopRight, BottomLeft, BottomCenter, BottomRight**)
CheckAlign	Bu komponent maydonidagi tugma holati. Agar u yuqori chap tarafdagi chegaraga yaqin bo'lsa unda qiymati "**TopLeft**"ga teng bo'ladi. Bundan boshqa ham bir nechta holatlari bor
Enabled	Bu komponentni ishlatish yoki ishlatmaslik imkontini bildiradi. Agar "**False**" bo'lsa unda ishlata olmaymiz
Visible	Bu komponentni ko'rsatish yoki yashirishini boshqaradi. Agar uning qiymati "**False**" bo'lsa unda komponent ko'rinmaydi
AutoCheck	Bu xossasi "**CheckBox**" dagi rasmni bosganda uning holati o'zgarishi yoki o'zgarmasligini aniqlab beradi. Uni formaga qoyganimizda qiymati "**True**" bo'lib beriladi.
FlatStyle	Bu "**CheckBox**" stili. "**CheckBox**" standart bo'lishi mumkin (**Standart**), tekis(**Flat**) yoki "**Popup**" bo'lishi mumkin
Appearance	U "**CheckBox**" ko'rinishini aniqlab beradi. Agar uning qiymati "**Normal**" bo'lsa odatdagidek ko'rinishda bo'ladi, agar "**Button**" bo'lsa tugma ko'rinishida bo'ladi
Image	Komponent maydonidagi rasm
ImageList	Bu rasmlar yig'indisi bo'lib shularning ichidan rasm qo'yishga bo'ladi. Uning uchun formaga "**ImageList**" komponentasi qo'shilgan bo'lishi kerak
ImageIndex	Tugmada chiqadigan "**ImageList**" rasmlar yig'indisidan olingan rasm nomeri (indeksi) [6]

RadioButton. "**RadioButton**" boshqarish elementi "**ChechBox**" boshqarish elementiga o'xshaydi. Ular orasida ikki ahamiyatli farq bor: Agar bir nechta "**RadioButton**"ni bir "**GroupBox**"ga joylashtirsak uning ichidan faqat bir

“**RadioButton**”ni belgilanadi, qolganlari belgilanmay qoladi, “**CheckBox**”da bo’lsa hoxlagancha belgilsh mumkin. “**RadioButton**”da “**ThreeState**” xossasi yo’q, ya’ni u “**CheckBox**”ga o’xshab “**Indeterminate**” qiymatini qabul qilmaydi. “**RadioButton**” boshqarish elementi 2.4-rasmda ko’rsatilgan.



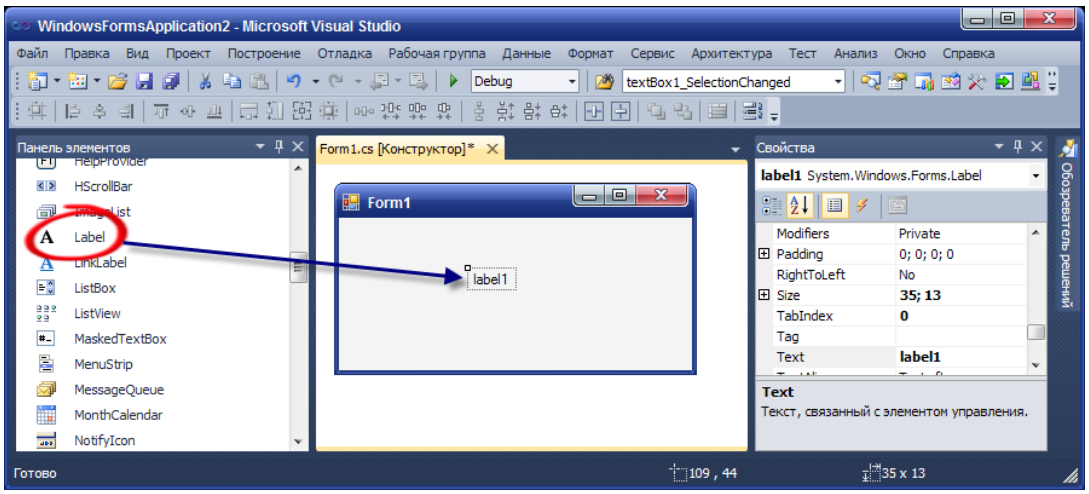
2.4-rasm

Bu boshqarish elementini ko’pchilik dasturlarda faqat bir qiymatni olish kerak bo’lgan holatda foydalanadi. Bizning dasturda ham shu maqsadda foydalaniladi. Masalan, foydalanuvchi o’zi haqidagi ma’lumotlarni kiritayotganda o’qish joyini kiritadigan bo’lsa, yuqori o’quv o’rni bilan kasb-hunar kolleji ikkisini bir vaqtning o’zida kirita olmasligi kerak. Faqat bittasini tanlashi kerak. Shunday holatlarda foydalaniladi. “**RadioButton**” boshqarish elementi xossalari quyidagi jadvalda ko’rsatilgan.

Xossasi	Bajaradigan vazifasi
Text	Tugma yonida yoziladigan matn
TextAllign	Komponentdagi matn holati. Matn o’rtada joylashishi mumkin (**MiddleCenter**), chap tarafda (**MiddleLeft**), o’ng tarafda (**MiddleRight**). Bundan boshqada bir nechta turlari bor (**TopLeft, TopCenter, TopRight, BottomLeft, BottomCenter, BottomRight**)
CheckAlign	Bu komponent maydonidagi tugma holati. Agar u yuqori chap tarafdagi chegaraga yaqin bo’lsa unda qiymati “**TopLeft**”ga teng bo’ladi. Bundan boshqa ham bir nechta holatlari bor
Enabled	Bu komponentning ishlatish yoki ishlatmaslik imkoniyatini

	bildiradi. Agar “**False**” bo’lsa unda ishlata olmaymiz
Visible	Komponentning ko’rsatish yoki yashirishini boshqaradi. Agar uning qiymati “**False**” bo’lsa unda komponent ko’rinmaydi
AutoCheck	Bu xossasi “**RadioButton**” dagi rasmni bosganda uning holati o’zgarishi yoki o’zgarmasligini aniqlab beradi. Uni formaga qoyganimizda qiymati “**True**” bo’lib beriladi.
FlatStyle	Bu “**RadioButton**” stili. “**RadioButton**” standart bo’lishi mumkin (**Standart**), tekis(**Flat**) yoki “**Popup**” bo’lishi mumkin
Appearance	U “**RadioButton**” ko’rinishini aniqlab beradi. Agar uning qiymati “**Normal**” bo’lsa odatdagidek ko’rinishda bo’ladi, agar “**Button**” bo’lsa tugma ko’rinishida bo’ladi
Image	Komponent maydonidagi rasm
ImageAlign	Rasmning komponentdagi joylashgan o’rnini aniqlab beradi. Rasm komponenta o’rtasida joylashishi mumkin (**MiddleCenter**), chap tarafda (**MiddleLeft**), o’ng tarafda (**MiddleRight**). Bundan boshqada bir nechta turlari bor (**TopLeft, TopCenter, TopRight, BottomLeft, BottomCenter, BottomRight**)
ImageList	Bu rasmlar yig’indisi bo’lib shularning ichidan rasm qo’yishga bo’ladi. Uning uchun formaga “**ImageList**” komponentasi qo’shilgan bo’lishi kerak
ImageIndex	Tugmada chiqadigan “**ImageList**” rasmlar yig’indisidan olingan rasm nomeri (indeksi) [6]

Label. Label boshqarish elementi matn ko’rinishida bo’lib u boshqa boshqarish elementi haqida yoki foydalanuvchi bilishi kerak bo’lgan ma’lumotlarni berib turadi. U ko’pincha “**TextBox**”, “**ComboBox**” boshqarish elementlari uchun qo’llaniladi ya’ni foydalanuvchi qanday ma’lumot kiritishi kerakligi matn turida beriladi. Bu boshqarish elementidan foydalanish uchun “**Text**” xossasiga kerakli ma’lumotni yozib uni kerakli joyga olib borishimiz yetarli bo’ladi. **Label** boshqarish elementi 2.5-rasmda ko’rsatilgan.



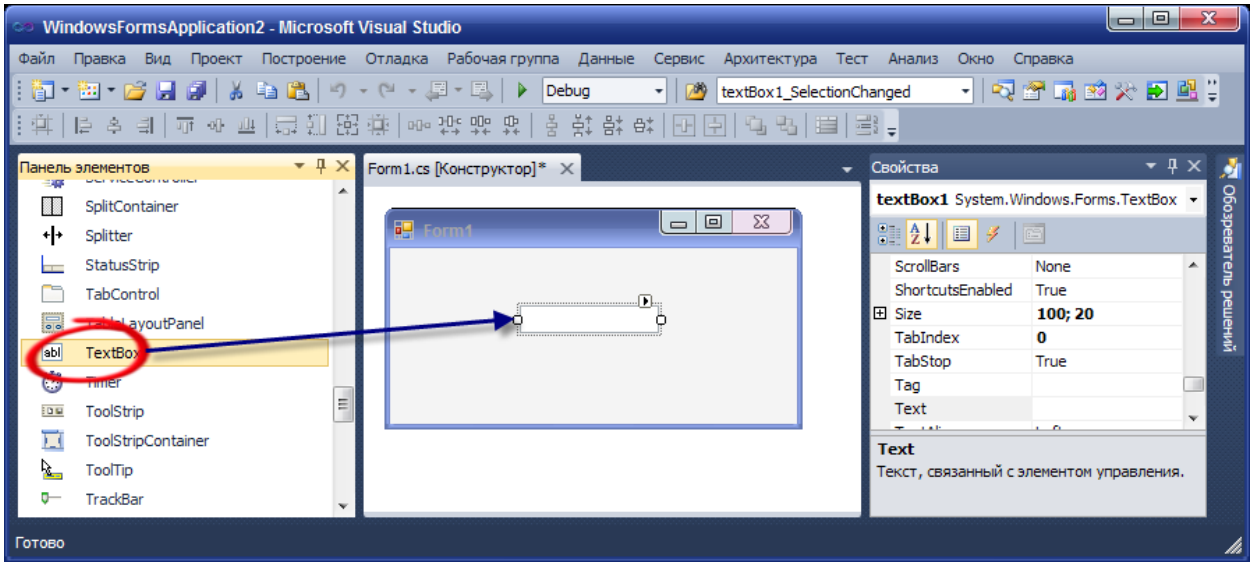
2.5-rasm

Label boshqarish elementi yuqorida aytganimizdek kerakli ma'lumotni ko'rsatish uchun foydalaniladi. Masalan, anketa to'ldiruvchi dastur bo'lsin. Agar ma'lumotlarni kiritish maydonchasi yonida u yerda qaysi ma'lumotni kiritish ko'rsatilmagan bo'lsa foydalanuvchi tushinmay qoladi. Shuning uchun "**Label**" foydalaniladi. Ya'ni familiyasini kiritadigan joyiga "**Familiyangizni kiriting**", yoshini kiritadigan joyiga "**Yoshingizni kiriting**" deb "**Label**" orqali yozib qo'ysak, u foydalanuvchi uchun tushunarli bo'ladi. Bizning dasturmizda matn topshiruvchi o'zi haqinda ma'lumotlarni kiritadigan joyida, to'g'ri va noto'g'ri belgilangan javoblarni ko'rsatishda, vaqtni ko'rsatishda, sozlash bo'liminda va h.k. ko'p joylarida foydalanamiz. Bu boshqarish elementi xossalarini quyidagi jadvalda joylashtirdik.

Xossasi	Bajaradiga vazifasi
Name	Komponent nomi bo'lib, u dastur holatida uning boshqa xossalaridan foydalanishda qo'llaniladi
Text	Ko'rinib turuvchi matn
Location	Forma ustida joylashgan o'rni
AutoSize	Komponentning o'lchamlari avtomat turda o'zgarishini aniqlab beradi. Agar uning qiymati **True** bo'lsa unda **Text** xossasidagi matni o'zgartirilsa o'lchamlar ham avtomat o'zgaradi
Size	Komponent o'lchamlari. Agar **AutoSize** xossasi **False** qiymatini qabul qilsa o'lchamlarini o'zgartirishga bo'ladi
MaximumSize	Kattalashtirish mumkin bo'lgan o'lchamni belgilab beradi. Bunda **MaximumSize.Width**-eni, **MaximumSize.Heigth**-balandligi
Font	Komponentdagi matnning shrift

ForeColor	Komponentdagi matn rangi
BackColor	Komponent foni rangi
TextAlign	Komponentdagi matn holati. Matn o'rtada joylashishi mumkin (**MiddleCenter**), chap tarafda (**MiddleLeft**), o'ng tarafda (**MiddleRight**). Bundan boshqada bir nechta turlari bor (**TopLeft, TopCenter, TopRight, BottomLeft, BottomCenter, BottomRight**)
BorderStyle	Komponent ramkasi ko'rinishi. Komponentni formaga joylashtirganimizda ramka bo'lmaydi (qiymati **None** bo'ladi). Ramkaning **Fixed3D** va **FixedSingle** degan turlari bor. [6]

TextBox. "**TextBox**" bu asosiy boshqarish elementlarining biri bo'lib, foydalanuvchi tarafidan matnli ma'lumotlarni kiritishga moslashgan. "**TextBox**" ni foydalanishda bir satrli va ko'p satrli qilish mumkin. Bu uning "**MultiLine**" xossasidagi qiymatni "**True**" yoki "**False**" qilib qo'yishingizga bog'liq. Ya'na "**MaxLength**" xossasi orqali kiritilishi mumkin bo'lgan simvollar sonini cheklashga bo'ladi. O'zi "**TextBox**" ga 64 kBt hajmdagi matnni kiritish mumkin. Bundan katta hajmdagi ma'lumotlar bilan ishlash kerak bo'lsa unda "**RichTextBox"** dan foydalangan ma'qul. "**TextBox**" 2.6-rasmda ko'rsatilgan.



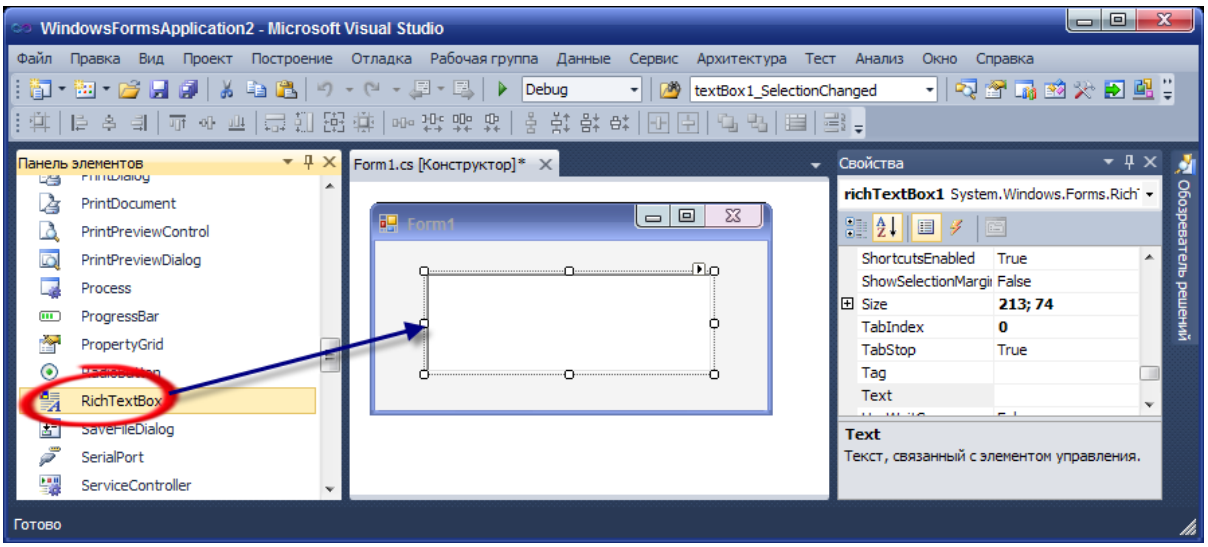
2.6-rasm

"**TextBox**" boshqarish elementi ko'pchilik dasturlarda qisqa matnli ma'lumotlarni kiritishda foydalaniladi. Bizning dastur holatida ham shunday vazifani

bajaradi. Asosan foydalanuvchi o’zi haqida ma’lumotlarni kiritayotganda foydalaniladi. “**TextBox**” xossalarini quyidagi jadvaldan ko’rishingizga bo’ladi.

Xossasi	Bajaradiga vazifasi
Name	Komponent nomi bo’lib, u dastur holatinda uning boshqa xossalaridan foydalanishda qo’llaniladi
Text	Ma’lumotlar kiritish maydonida turuvchi matn
Location	Forma ustida joylashgan o’rni
MaximumSize	Kattalashtirish mumkin bo’lgan o’lchamni belgilab beradi. Bunda **MaximumSize**.**Width**-eni, **MaximumSize**.**Heigth**-balandligi
Font	Komponentdagi matnning shrift
ForeColor	Komponentdagi matn rangi
BackColor	Komponent foni rangi
TextAlign	Komponentdagi matn holati. Matn o’rtada joylashishi mumkin (**MiddleCenter**), chap tarafda (**MiddleLeft**), o’ng tarafda (**MiddleRight**). Bundan boshqada bir nechta turlari bor (**TopLeft, TopCenter, TopRight, BottomLeft, BottomCenter, BottomRight**)
MaxLength	**TextBox** ga kiritish mumkin bo’lgan simvollarning maksimal soni
PasswordChar	Foydalanuvchi tarafidan matn kiritilsa u faqat qo’yilgan bir simvolni ko’rsatib turadi. Masalan * simvoli qo’yilgan bo’lsa, matn deb kiritsak unda faqatgina **** simvollari ko’rinib turadi (bu parol kiritiladigan holatlarda foydalaniladi)
MultiLiue	Uning qiymati “**True**” bo’lsa “**TextBox**” ga bir nechta satrga matnlar kiritish mumkin, “**False**” bo’lsa faqat bir satrda matn kiritish mumkin.
ReadOnly	Uning qiymati “**True**” bo’lsa “**TextBox**”dagi matnni tahrirlash mumkin. “**False**” bo’lsa tahrirlash mumkin emas [6].

RichTextBox. “**RichTextBox**” boshqarish elementida katta hajmdagi ma’lumotlarni kiritishga va qayta ishlashga bo’ladi. Bundan boshqa ham “**RichTextBox**”da matn rangini tahrirlash, shrifrini o’zgartirish, rasm qo’yishga bo’ladi. Ya’na “**RichTextBox**”da “**Microsoft Word**” matn tahrirlash imkoniyatlarini o’zida jamlagan. Lekin u to’g’ridan-to’g’ri faqatgina **.rtf** formatdagi hujjatlarni o’qiydi. “**RichTextBox**” boshqarish elementi 2.7-rasmda ko’rsatilgan



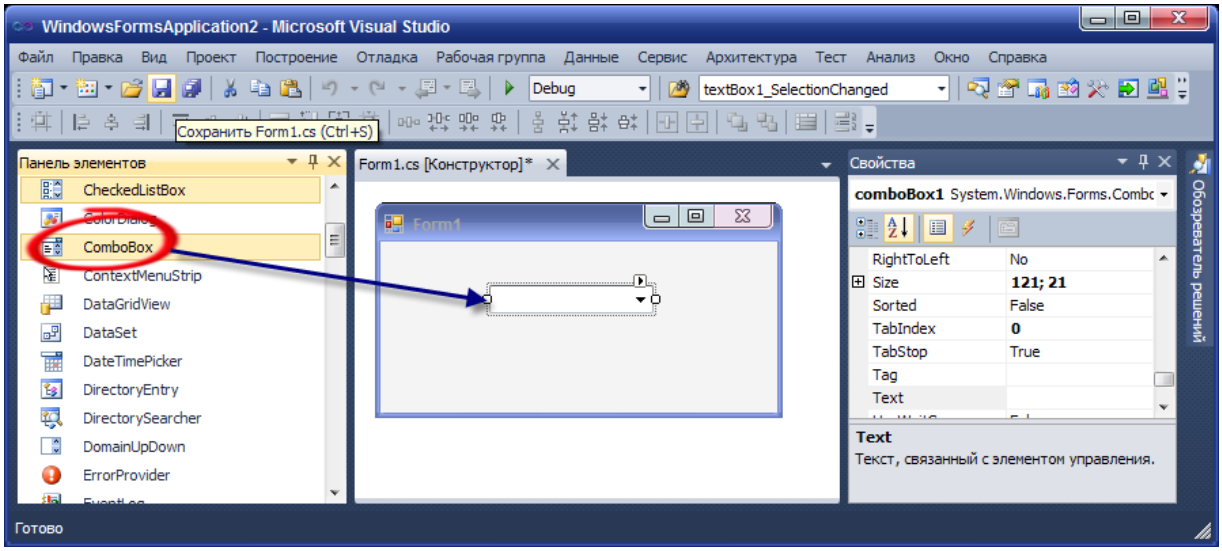
2.7-rasm

“**RichTextBox**” boshqarish elementi test dasturida test topshiruvchi savollarini o’qishi uchun foydalaniladi. Xossalari quyidagi jadvalda:

Xossasi	Bajaradiga vazifasi
Name	Komponent nomi bo’lib, u dastur holatida uning boshqa xossalaridan foydalanishda qo’llaniladi
MaximumSize	Kattalashishi mumkin bo’lgan o’lchamni belgilab beradi. Bunda “**MaximumSize.Width”**-eni, “**MaximumSize.Heigth**”-balandligi
Font	Komponentdagi matnning shrifti
ForeColor	Komponentdagi matn rangi
BackColor	Komponent foni rangi
BorderStyle	Komponent ramkasi ko’rinishi. Komponentni formaga joylashtirganimizda ramka bo’lmaydi (qiymati **None** bo’ladi). Ramkaning **Fixed3D** va **FixedSingle** degan turlari bor.
MaxLength	“**RichTextBox**” ga kiritish mumkin bo’lgan simvollarning maksimal soni. Dastlab u 2147483647 soniga teng bo’ladi.
MultiLine	Uning qiymati “**True**” bo’lsa “**TextBox**”ga bir nechta satrga matnlar kiritish mumkin, “**False**” bo’lsa faqat bir satrda matn kiritish mumkin.
ReadOnly	Uning qiymati “**True**” bo’lsa “**RichTextBox**”dagi matnni tahrirlash mumkin. “**False**” bo’lsa tahrirlay olmaymiz uni faqat nusxalab olishimizga bo’ladi [6].

ComboBox. “**ComboBox**” boshqarish elementida tahrirlash maydoni va tizim bor bo’lib, u ma’lumotlarni qo’lda kiritish yoki tizimdan saralab olish imkonini beradi. Uning qulaylilik tarafi formadan katta joy egallamaydi. Foydalanuvchi tizimni

ochib undan kerakli ma’lumotni saralab olishi yoki o’zi boshqa qiymat kiritishi mumkin. “**CombiBox**”ni formaga joylashtirish 2.8-rasmda ko’rsatilgan.



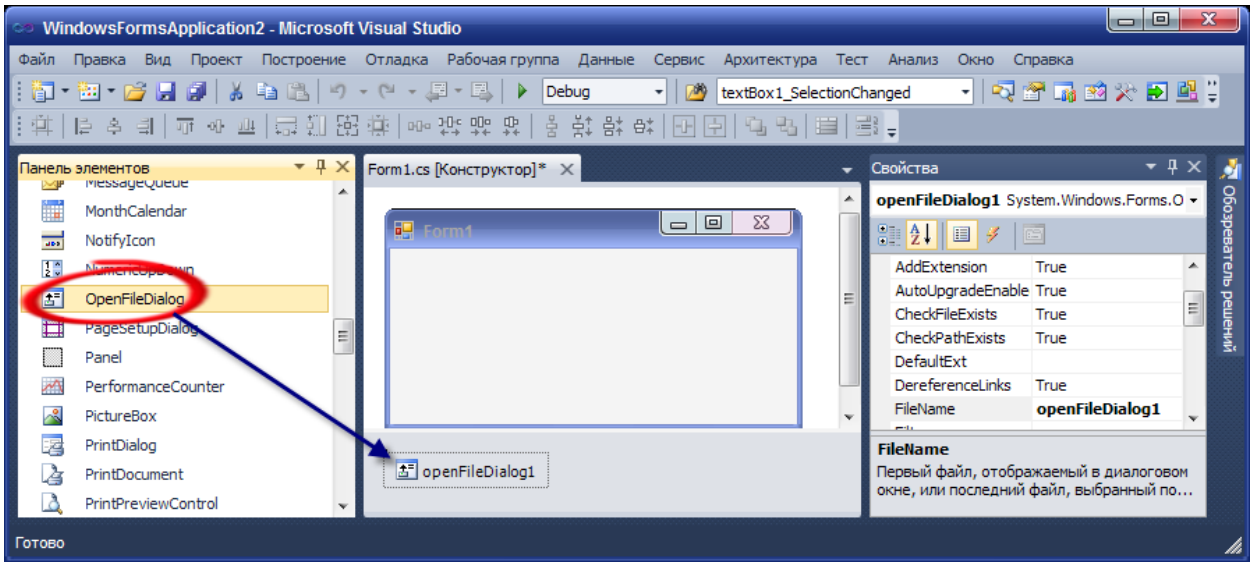
2.8 -rasm

“**ComboBox**” boshqarish elementini dasturda ko’pchilik holatlarda foydalanuvchi bir ma’lumotni qayta-qayta qo’lda termasdan tizimning ichidan saralab olish imkoniyatiga ega bo’ladi. Endi uning xossalari quyidagi jadvalda ko’rsatilgan.

Xossasi	Bajaradigan vazifasi
DropDownStyle	“**ComboBox**” turi. “**DropDown**”-kiritish maydonchasi va yig’iladigan tizim, “**Sipmle**”-kiritish maydonchasi va tizim, “**DropDownList**”- yig’iladigan tizim
Text	Yozish maydonchasidagi matn
Items	Tizim elementlari-satrlar
ItememsCount	Tizimdagi elementlar soni
SelectedIndex	Tizimdan tanlangan element nomeri. Agar hechqanday element tanlanmagan bo’lsa unda u-1ga teng bo’ladi
Sorted	Tizimga yangidan element kiritilsa avtomat turda saralanadi
MaxDropDownItems	Yig’iluvchi tizimning ko’rinib turadigan elementlar soni
Location	Forma ustida joylashgan o’rni
Size	Komponent o’lchamlari
DropDownWidth	Tizimlar sohasining eni
Font	Shrifti [6].

OpenFileDialog. Lomponent “**OpenFileDialog**” bu o’zimizga kerakli bo’lgan faylni ochish uchun mu’ljallangan muloqat oynasi. Bu komponent o’zimizga kerakli

faylni kengaytmasi orqali filtrlab berishi mumkin. Ya’ni bizga faqat kengaytmasi ***.txt, *.rtf, *.docx** bo’lgan fayllar kerak bo’lsa muloqat oynada faqat shu fayllarni ko’rsatadi. U 2.9-rasmda ko’rsatilgan.



2.9-rasm

“**OpenFileDialog**” komponenti bizning dasturimizda savollar fayli o’qitish (ochish) uchun qo’llaniladi. Ya’ni bu fayl qaysi joyda turgan yo’lini dastur bilmaydi, uni shu komponent orqali topib olishimizga bo’ladi. “**OpenFileDialog**” komponenti xossalari quyidagi jadvalda ko’rsatilgan.

Xossasi	Bajaradigan vazifasi
Title	Oynaning sarlavhalar satridagi matn. Agar ma’nosi berilmagan bo’lsa unda oynaning sarlavhalar satrida **Открыть** degan matn turadi
Filter	Bu xossa fayllarni filtrlab beradi. Tizimda faqatgina kengaytmasi ko’rsatilgan fayllargina ko’rinib turadi. Masalan, **RTF** fayllar|*.rtf qiymati tizimda faqatgina **rtf** kengaytmali fayllarnigina ko’rsatib beradi
FileName	Bu foydalanuvchi tomonidan kiritilgan yoki tizimdan saralab olingan Fayl nomi
InitialDirectory	Belgili bir katolog ichidagi fayllarni dialog oynasi ochilganda ko’rsatib beradi [6]

Timer. "Timer" boshqarish elementi o'zi nomidan belgili vaqt bilan ishlashadi. Uni "**Tick**" hodisasi orqali amalga oshiradi. Uning "**Interval**" xossalari qiymat berib necha millisekunddan "**Tick**" hodisasiga borishi kerak ekanligi ko'rsatishga bo'ladi. Dastur kodida bajarilishi kerak vazifani quyidagi "**Tick**" hodisasi ichiga yoziladi:

```
privatevoid timer1_Tick(object sender, EventArgs e)
{
}
```

"**Timer**" boshqarish elementi bizning dasturmizda asosiy vazifani bajaradi. Ya'ni test topshirishning vaqti bilan ishlashadi. Uning xossalari quyidagi jadvalda ko'rishimizga bo'ladi.

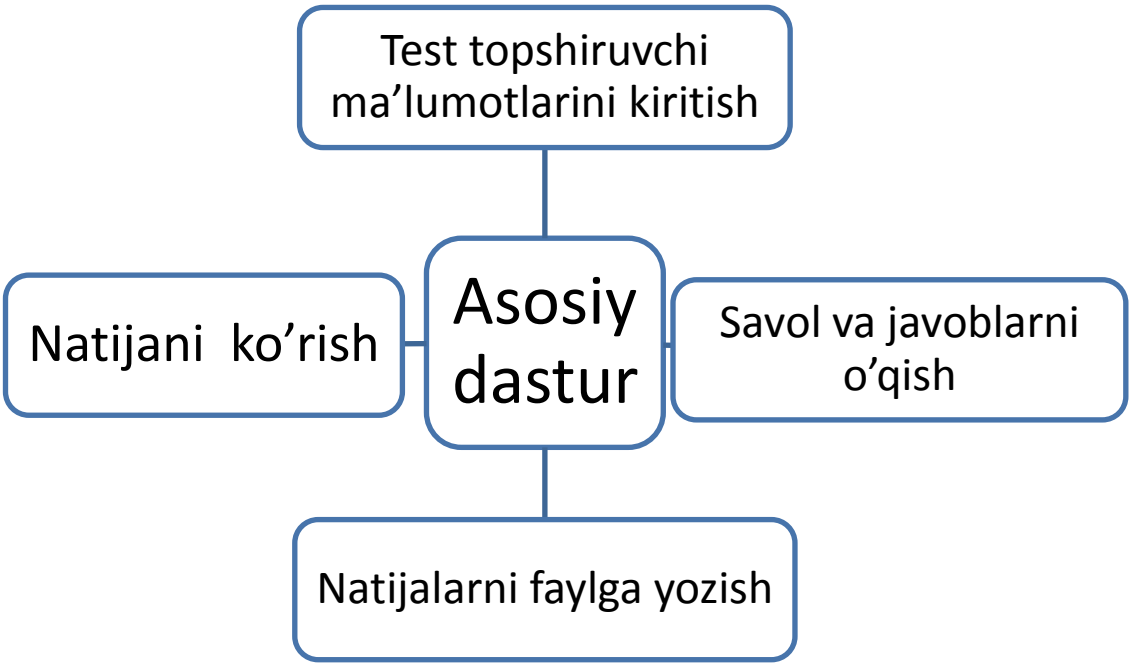
Xossasi	Bajaradigan vazifasi
Interval	Bu "**Tick**" hodisasidagi period. U millisekundlarda beriladi
Enabled	Qiymati "**False**" bo'lsa "**Timer**" vaqtida to'xtatadi, "**True**" bo'lsa qaytadan yurgizadi

3-§. TEST_RTF dasturiy ta'minotining asosiy algoritmi

Har qanday dasturni yaratishda katta kichik bo'lishiga qaramasdan eng avval bu dasturning algoritimi tuziladi. Algoritm – bu bir masalani yechish uchun ma'lum bir amallarni bajarishning ketma-ketligi bo'lib hisoblanadi. Keling biz algoritm tariflariga to'xtalib o'tirmasdan test dasturining ishlash algoritimi bilan tanishib chiqaylik. Dastlab dastur tuzilishi haqida qisqacha to'xtalib o'tsak. Dastur asosan katta 4 bo'limdan turadi.

- Javoblar faylini shifrlash
- Test savollari faylini yuklash
- Test topshiruvchi ma'lumotlarini kiritish
- Natijalarni faylga yozish

Quyidagi rasmda dasturiy ta`minotning asosiy strukturasi berilgan (3.1-rasm).



3.1.-Rasm. Dasturiy ta`minotning asosiy strukturasi

Javoblarni o’qish uchun alohida dasturiy ta’minot yaratilgan bo’lib, u javoblarni o’qiydi va binar faylga yozadi.

Yaratilgan dasturiy ta’minot quyidagi vaziyfalarni bajaradi:

Bu dasturiy ta`minoti oynasiga ikkita tugma, saveFileDialog va openFileDialog komponentalari joylashtilgan. Javoblar joylashgan fayl quyidagi ko`rinishda bo`ladi:

1-1

2-2

3-4

4-1

….

Bu yerda birinchi savolning tartib raqami va “-” belgisidan keyin ushbu savol javobi joylashgan ko`rinishda bo’ladi. Savollarni tuzuvchi ularning javoblarini adashtirmasligi uchun uning tartib raqami ishlatildi.

“**Javoblarni o’qish**” tugmasidagi quyidagi keltirilgan algoritm yordamida test savollari javoblari openFileDialog komponentasidan foydalanib ko`rsatlgan fayldan o`qiladi:

```
private void button1_Click(object sender, EventArgs e)
    {
```

```
        try
        {
            openFileDialog1.ShowDialog();
            fin = new StreamReader(openFileDialog1.FileName);
            int i = 0;
            juwap = new int[10000];

            while (!fin.EndOfStream)
            {
                str = fin.ReadLine();
                textBox1.Text += str + ";  ";
                str = str.Substring(str.IndexOf('-') + 1, 1);
                juwap[i] = System.Convert.ToInt32(str); i++;
            }
            n = i;
            fin.Close();
        }
        catch { MessageBox.Show("Fayl ochilmadi"); }
    }
```

Bu yerda istisno holatlarni qayta ishlash e`tiborga olingan bo`lib agarda fayl yuklanmasa catch blogidagi MessageBox.Show(); metodi orqali "Fayl ochilmadi" xabari ekranga chiqadi. Bu dastur kodida quyidagi ob`yektlar e`lon qilinadi va ishlatiladi:

```
int[] juwap; // javoblar massivi uchun;
    StreamReader fin;// faylni o`qish uchun
    BinarWriter fout;// binar faylga yozish uchun;
    string str;// javoblarni satrga o`qish uchun
    int n; // javoblar soni uchun ishlatiladi.
```

1. Dastlab openFileDialog1.ShowDialog() metodi orqali muloqat oyna ochiladi va ko`rsatilgan fayl nomidagi fayl ochiladi:

fin = new StreamReader(openFileDialog1.FileName);

2. Keyin javoblar massiviga xotiradan joy ajiratiladi:

juwap = new int[10000];

3. while operatori yordamida fayl oxirigacha javoblar quyidagicha o`qiladi:

fin.ReadLine() metodi satrni str ga o`zlashtiradi;

satrni textbox boshqarish elementiga chiqaradi;

str.IndexOf('-') metodi str satrdagi “-” simvoli indeksini aniqlaydi;

str.Substring(str.IndexOf('-') + 1, 1) metodi “-” simvolidan keyingi simvoldagi javob raqamini satr ko`rinishida str ga yozadi;

juwap[i] = System.Convert.ToInt32(str) metodi str satrni butun turdagi songa aylantiradi va juwap massiviga yozadi;

i ++ inkrement operatori orqali juwap massivining indeksi birga orttiriladi.

4. Javoblar soni n ga o`zlashtiriladi va fin.Close() metodi yordamida fayl yopiladi.

“Kodlash ” tugmasi yordamida binar fayl yaratiladi va javoblar yoziladi. Tugmani bosish hodissiga quyidagi kod yozilagan;

```
saveFileDialog1.ShowDialog();
fout=new
BinarWriter(File.Open(saveFileDialog1.FileName,FileMode.OpenOrCreate));
        for (int i = 0; i < n; i++)
            fout.Write(juwap[i]);
        fout.Close();
```

1. saveFileDialog1.ShowDialog() metodi yordamida saqlash dialog oynasi ochiladi va fayl nomi kiritiladi(fayl nomi savollar fayli nomi bilan bir xil bo`lishi shart).

2. Ko`rsatilgan nomda binar fayl yaratiladi:

```
fout=new
BinarWriter(File.Open(saveFileDialog1.FileName,FileMode.OpenOrCreate));
```

3. for operatori yordamida ko`rsatilgan indeksdagi javoblar fout.Write(juwap[i]) metodi orqali faylga yoziladi.

4. fout.Close() fayl yopiladi va javoblar binar faylda saqlanadi.

Shunday qilib javoblarni binar faylga yozish ya`ni Javoblar faylini shifrlash uchun alohida dasturiy ta`minot yaratilgan.

Keyingi dasturiy ta`minot asosiy TEST_RTF bo`lib hisoblanadi va u quyidagi vazifalarni bajaradi.

1. Dasturiy ta`minotda foydalanilgan bosh oynaning global o`zgaruvchilari quyidagicha e`lon qilinadi:

```
string str;//savollar fayli nomi
public Kiritish kiritish; // test topshiruvchi ma`lumotlarini kiritish oynasi
string[] savol = new string[1000];// savollar massivi
public string strr;// fayldagi fan nomi
int [] rnd = new int [1000];// savollarni aralashtiruvchi
//tasodifiy sonlar massivi
int[] javob = new int[1000];// javoblar massivi
int d = 0, d1 = 0, uzinliq=0;// savollarni o`qishda o`rni va uzunligi
int indeks = 0,ind=0,t=0,x=0;// javoblar va savollar tartib raqami uchun
string fs;// javoblar fayli nomi
bool b = true;. //tugmalarga murojaat
```

2. Faylni_yuklash() metodi yordamida savollar va shu nomdagi binar faylda javoblarni o`qish amalga oshiriladi:

```
public void Faylni_yuklash()
    {
      try
      {
        openFileDialog1.ShowDialog();
        fs = openFileDialog1.FileName;// savollar fayli
        str = fs.Substring(0, fs.IndexOf('.'))+".txt";//javoblar fayli
         javoblar(str);// javoblarni o`qish
        richTextBox1.LoadFile(fs);.. savollar faylini to`liq o`qish
```

```
                kiritish = new Kiritish();// ma`lumotlarni kiritish oynasi e`loni
                 oqiw();// har bir savolni ajiratib massivga o`qish
                 random(indeks);// aralashtirish
                richTextBox2.Rtf=savol[rnd[ind]];// aralashtirilgan savolni ko`rsatish
              }
              catch{
        MessageBox.Show("Fayl yuklanmadi"); // faylni yuklashda
// xatolik sodir bo’lsa
}
      }
```

Savollar fayli quyidagi krinishda RTF formatidagi faylda saqlanadi:

«Dasturlash asoslari» fanidan

##@

MS Visual Studio muxitida dasturni bajarish uchun qaysi klavisha qo’llaniladi?

A) ***F5***

B) ***Ctrl + F9***

C) ***Shift + F9***

D) ***Alt + F9***

##@

C++ tilidagi izoh’ to’g’ri berilgan javobni toping.

A) ***\\ C++ programmalash tili***

B) ***// C++ programmalash tili***

C) ****C++ programmalash tili ****

D) ***{ C++ programmalash tili }***

##@

...

Har bir savol “##@” simvollar ketma-ketligi bilan ajiratilgan.

3. Javoblar(str) metodi yordamida javoblar faylidagi barcha javoblar binar fayldan javob massiviga o`qib olinadi, dastur kodi quyidagicha:

```
public void javoblar(string s){
        BinarReader ss=new BinarReader(File.Open(s, FileMode.Open)) ; //
binar faylni ochish
          int i=0;
          while (ss.PeekChar() > -1)// javovlarni fayl //oxirigacha o`qish
          {
            javob[i++]= ss.ReadInt32();//javoblar massivi
          }
       }
```

4. Dasturda ikkita richTextBox boshqaruv elementi ishlatiladi. Sababi birinchisida savollar to`liq o`qiladi. Ikkinchisi har bir savolni ekranga chiqarish uchun foydalaniladi. oqiw() metodida dastlab test topshiruvchi haqida ma`lumotlar kiritish oynasini ochish amalga oshiriladi. Keyin "##@" belgisi bilan ajratilgan har bir savol savol massivi elemetntlariga yozib boriladi.

```
public void oqiw()
       {
          try
          {
            kiritish.ShowDialog();//test topshiruvchi haqida
//ma`lumotlar kiritish oynasini
              uzinliq = richTextBox1.Find("##@");
                 richTextBox1.SelectionStart = 0;
       richTextBox1.SelectionLength =  uzinliq;
          strr= richTextBox1.SelectedText;
            dannie.text = strr;
            this.Hide();
toolStripTextBox1.Text = dannie.soni.ToString();
            while (true)
```

```
                {
                    d = richTextBox1.Find("##@", d, RichTextBoxFinds.MatchCase);
                    d++;
                    d1 = richTextBox1.Find("##@", d,
RichTextBoxFinds.MatchCase);
                    richTextBox1.SelectionStart = d + 3;
                    richTextBox1.SelectionLength = d1 - d - 3;
                    savol[indeks++] = richTextBox1.SelectedRtf;
                }
            }
            catch
            {
                b = false;
            }
            ss:  if (indeks < dannie.soni)
            {
            MessageBox.Show("Savollar sonini qaytadan kiriting");
            kiritish.textBox1.Enabled = true;
              kiritish.ShowDialog(); goto ss;
              }
        }
```

5. random() metodi savol va javoblarni tasodifiy va takrorlanmaydigan qilib aralashtirish uchun foydalaniladi u quyidagi dastur kodiga ega:

```
public void random(int max)
    {
      int k = 0,n;
      Random obrnd = new Random();
      rnd[k] = obrnd.Next(max); k++;
      while (k < max)
      {
```

```
        rand: rnd[k] = obrnd.Next(max);//tasodifiy sonni tanlash
            for (int i = 0; i < k; i++)
                if (rnd[i] == rnd[k]) goto rand;// takrorlanish sodir bo'lsa
            k++;
        }
    }
```

Bu yerda max savollar soni, agarda qandayda bir raqam takrorlansa, u holda obrnd.Next() metodi bilan maxgacha boshqa tasodifiy sonni tanlanadi.

6. Har bir tugmada Savol() metodi yordamida javoblarning to`g`ri yoki noto`g`ri ekanligi aniqlanadi va tasodifi sonlar massivi rnd elementi bo'lgan indeksdagi keyingi savol richTextBox2 boshqaruv elementida ko`rsatiladi:

```
public void Savol(int j)
    {
        if (j == javob[rnd[ind]]) t++; else x++;
        ind++;
        if (ind == dannie.soni) { buttonlar(false); MessageBox.Show("Oxirgi savolga javob berdingiz \" Tugatish\" tugmasini bosing!"); goto aa; }
        richTextBox2.Rtf = savol[rnd[ind]];
    aa: ;
    }
```

Oxirgi savolga javob berilgandan keyin "Tugatish" tugmasini bosish talab qilinadi va buttonlar() metodini chaqirish bilan javoblar tugmalarga murojaat qilish to`xtatiladi.

7. "Tugatish" tugmasini bosish natijasida test topshiruvchi ma`lumotlari va javoblari Form3 oynasida ko`rsatiladi va tugmalarga keyingi test topshirish uchun murojaat qilish imkoniyati beriladi:

```
private void button5_Click(object sender, EventArgs e)
    {
        this.Hide();
        buttonlar(true);
```

```
            Form3 fjavob = new Form3();
            fjavob.label4.Text = dannie.text + "\n" + dannie.fio;
            fjavob.label1.Text = "To'g'ri javob:  " + t.ToString();
            fjavob.label2.Text = "Noto'g'ri javob:  " + x.ToString();
            fjavob.label3.Text = "Umumiy natija: " +String.Format("{0:0.##}", (((float) t
/ (x + t)) * 100)).ToString()+"%";
            fjavob.ShowDialog();
            ind = 0;
            qayta();
            x = t = 0;
        }
```

Bu yerdagi qayta() metodi yordamida keyingi test topshirish uchun savollar qaytadan tasodifiy aralashtiriladi, u quyidagi dastur kodi orqali amalgoa oshiriladi:

```
public void qayta() {
                random(indeks);
                richTextBox2.Rtf = savol[rnd[ind]];
            }
```

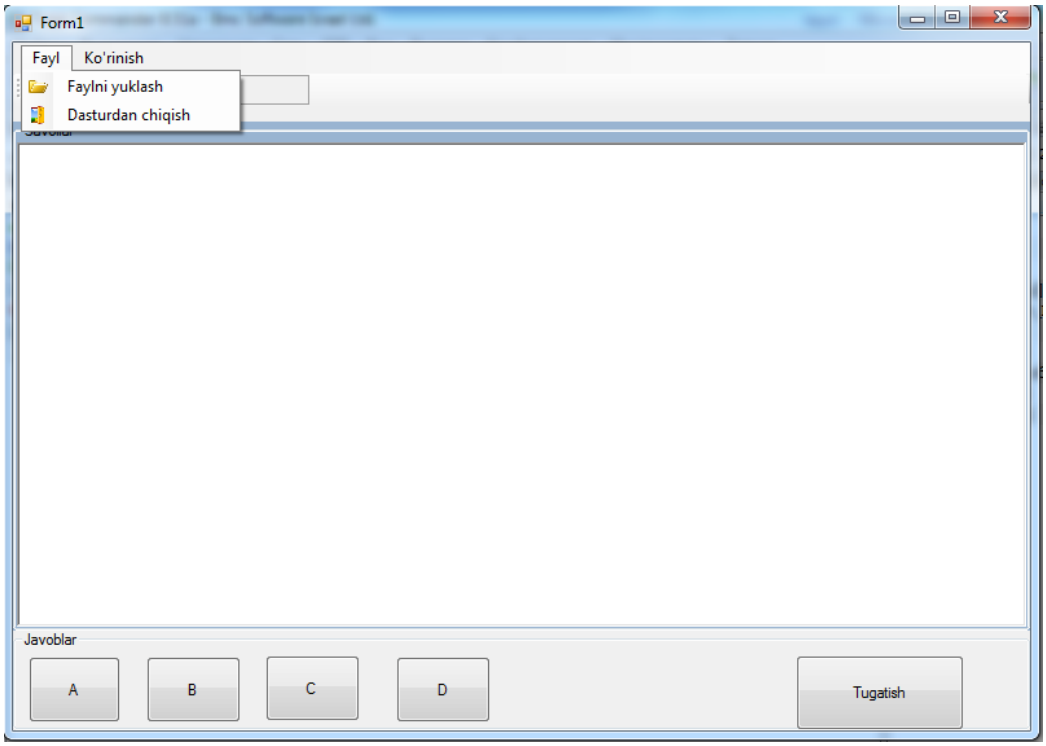
Ma`lumotlarni har bir yaratilgan formalar o`rtasida almashish yoki foydalanish uchun murojaat qilish imkoniyatini o`rnatish maqsadida loyihaga qo`shimcha sinfi qo`shilgan. O`zgruvchilar quyidagi sinfda keltirilgan izohlardagi maqsadda ishlatiladi:

```
tatic class dannie
    {
        public static string text;// Fan nomi
        public static string fio;// Test topshiruvchi FISh
        public static string kurs;// Test topshiruvchi talaba kursi
        public static string k_nomer;// Test topshiruvchi talaba guruhi
        public static int soni;// Test topshiruvchi savollar soni
        public static string fakultet; // Test topshiruvchi fakul`teti
    }
```

Bu yerdagi sinf test topshiruvchi haqidagi ma`lumotlarni foydalanib natija formasida ko`rsatish maqsadida yaratilgan.

4-§. Dastur taminotini foydalanish ko’rsatmalari

Eng dastlab test dasturini ishga tushirganimizda asosiy bo’lim ochiladi. Bu bo’lim boshqa bo’limlarni birlashtirib turadi. Ya’ni boshqa bo’limlarga kiritadigan bo’lsangiz shu bo’lim orqali amalga oshiriladi. Uning uchun asosiy bo’lim oynasidan kerakli bo’limni(4.1-rasm).



4.1-rasm. Test dasturining oynasi.

Test dasturi oynasi quyidagilardan iborat:

- Menyu satri;
- Uskunalar satri;
- Test savollari maydoni;
- Javoblar bo’limi;
- Tugatish tugmasi;

Menyu satrida **Fayl** va **Ko’rinish** menyulari joylashgan. **Fayl** menyusida **Faylni yuklash** va **Dasturdan chiqish** buyruqlari bor. **Ko’rinish** menyusida uskunalar satrini ko’rsatish yoki olib tashlash bayroqchasi joylashtirilgan.

Fayl yoklash buyrig’i elementi quyidagicha:

```
public void Faylni_yuklash()
    {
        try
        {
            openFileDialog1.ShowDialog();
            fs = openFileDialog1.FileName;
            str = fs.Substring(0, fs.IndexOf('.'))+".txt";
           // textBox1.Text = str;
            javoblar(str);
            richTextBox1.LoadFile(fs);kiritish = new Kiritish();
            oqiw();
            random(indeks);

           richTextBox2.Rtf=savol[rnd[ind]];


        }
        catch
        {
            MessageBox.Show("Файл юкланмади");

        }
    }
```

Savollarni va javoblarni o’qitish algoritimi. ***Savollarni o’qitish.*** Test dasturida savollarni va javoblarni **.rtf** formatli fayllardan o’qitamiz. Ya’ni savollar va javoblar shunday formatli fayllarga yozilgan bo’lishi kerak. Test savollari bir-biridan qandayda bir belgi yordamida ajratilgan bo’lishi kerak. Uning uchun biz har bir savol orasida savolda uchrashadigan simvollar ketma-ketligini qo’yib chiqishimiz kerak. Sababi savollarni “**RichTextBox**” ga yozamiz, “**RichTextBox**”da bo’lsa “**MatchCase**” degan metod bo’lib u biz ko’rsatgan simvollar ketma-ketligi nechanchi o’rinda turgan simvol ekanini aniqlab beradi. Shu topilgan simvollar orasidan bo’yab olsak u bizning bir savolimiz bo’ladi. Uning ishlash prinsipini tushinish uchun quyidagi test savolini qaraylik:

«Dasturlash asoslari» fanidan

##@

MS Visual Studio muhitida dasturni bajarish uchun qaysi klavisha qo’llaniladi?

E) F5

F) Ctrl + F9

G) Shift + F9

H) Alt + F9

##@

C++ tilidagi izoh to’g’ri berilgan javobni toping.

E) \\ C++ programmalash tili

F) // C++ programmalash tili

G) *C++ programmalash tili */

H) { C++ programmalash tili }

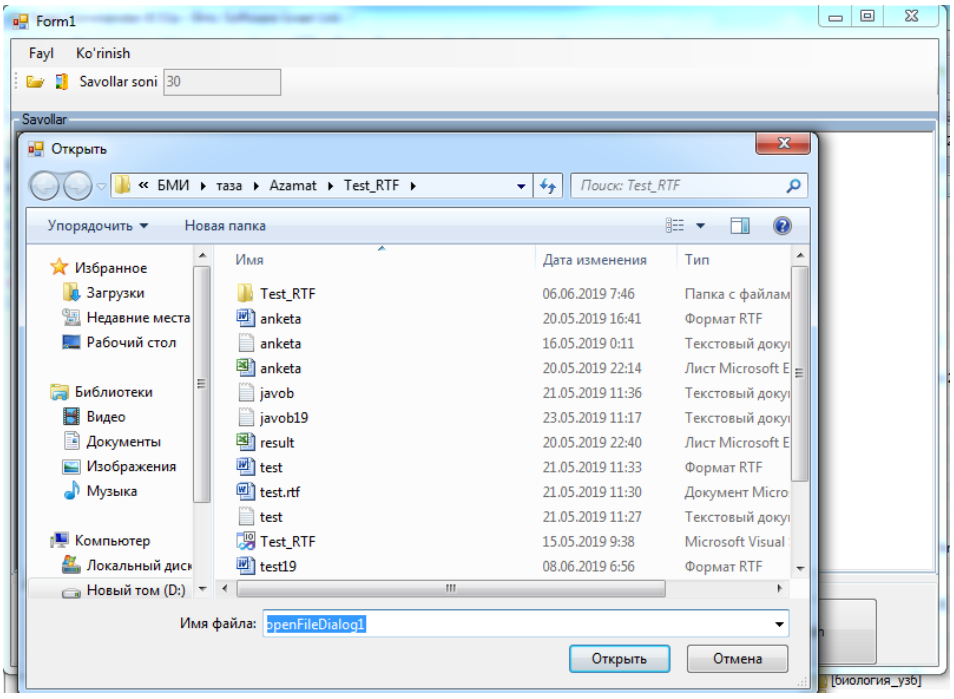
##@

Bunda savolda yo’q simvollar ketma-ketligi ##@ bo’lganligidan uni har bir savol orasida qo’yamiz.

Fayl bo’limida quyidagilar joylashgan:

- Faylni yuklash;
- Dasturdan chiqish;

Faylni yuklash tugmasi bo’silganda ***.rtf** kengaytmadagi test savollari yozilgan fayl tanlanadi(4.2-rasm).



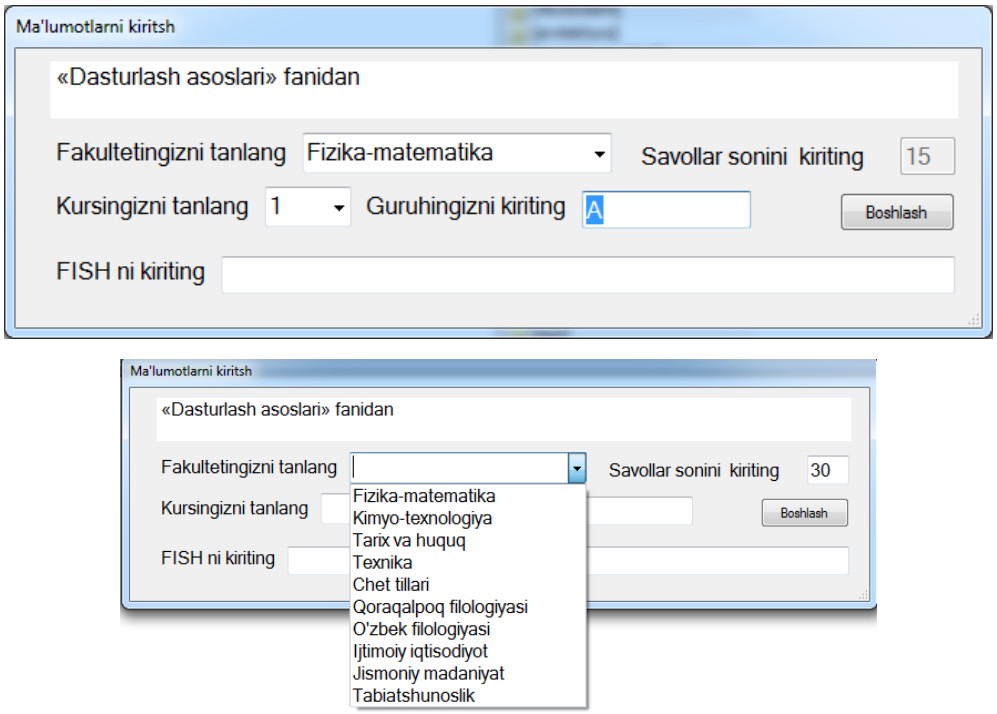
4.2-rasm. Fayni yuklash oynasi.

Agar dasturdan chiqish tugmasi bosilsa dastur ish faoliyatini tugatadi.

```
public void oqiw()
{
   try
    {

        uzinliq = richTextBox1.Find("##@");
             richTextBox1.SelectionStart = 0;

             richTextBox1.SelectionLength =  uzinliq;

        strr= richTextBox1.SelectedText;

        dannie.text = strr;
        this.Hide();
      kiritish.ShowDialog();

      toolStripTextBox1.Text = dannie.soni.ToString();
        while (true)
        {
            d = richTextBox1.Find("##@", d, RichTextBoxFinds.MatchCase);

            d++;
            d1 = richTextBox1.Find("##@", d, RichTextBoxFinds.MatchCase);
            richTextBox1.SelectionStart = d + 3;
            richTextBox1.SelectionLength = d1 - d - 3;
            savol[indeks++] = richTextBox1.SelectedRtf;
        }

    }
    catch
    {

       // MessageBox.Show("Fayl yuklandi"+indeks);
        b = false;
    }
   ss:  if (indeks < dannie.soni)
    {

    MessageBox.Show("Savollar sonini qaytadan kiriting");
    kiritish.textBox1.Enabled = true;

       kiritish.ShowDialog(); goto ss;

    }
  //  kiritish.Close();
}
private void toolStripButton1_Click(object sender, EventArgs e)
{

}

private void button5_Click(object sender, EventArgs e)
{
    this.Hide();

    buttonlar(true);

    Form3 fjavob = new Form3();
    fjavob.label4.Text = dannie.text + "\n" + dannie.fio;
    fjavob.label1.Text = "To'g'ri javob:  " + t.ToString();
    fjavob.label2.Text = "Noto'g'ri javob:  " + x.ToString();
```

```
            fjavob.label3.Text = "Umumiy natija: " +String.Format("{0:0.##}", (((float) t /
(x + t)) * 100)).ToString()+"%";
            fjavob.ShowDialog();
            ind = 0;

            qayta();
            x = t = 0;
        }
```

Ma’lumotlarni kiritish oynalari



4.3-rasm. Ma’lumatlarni kiritish oynasi.

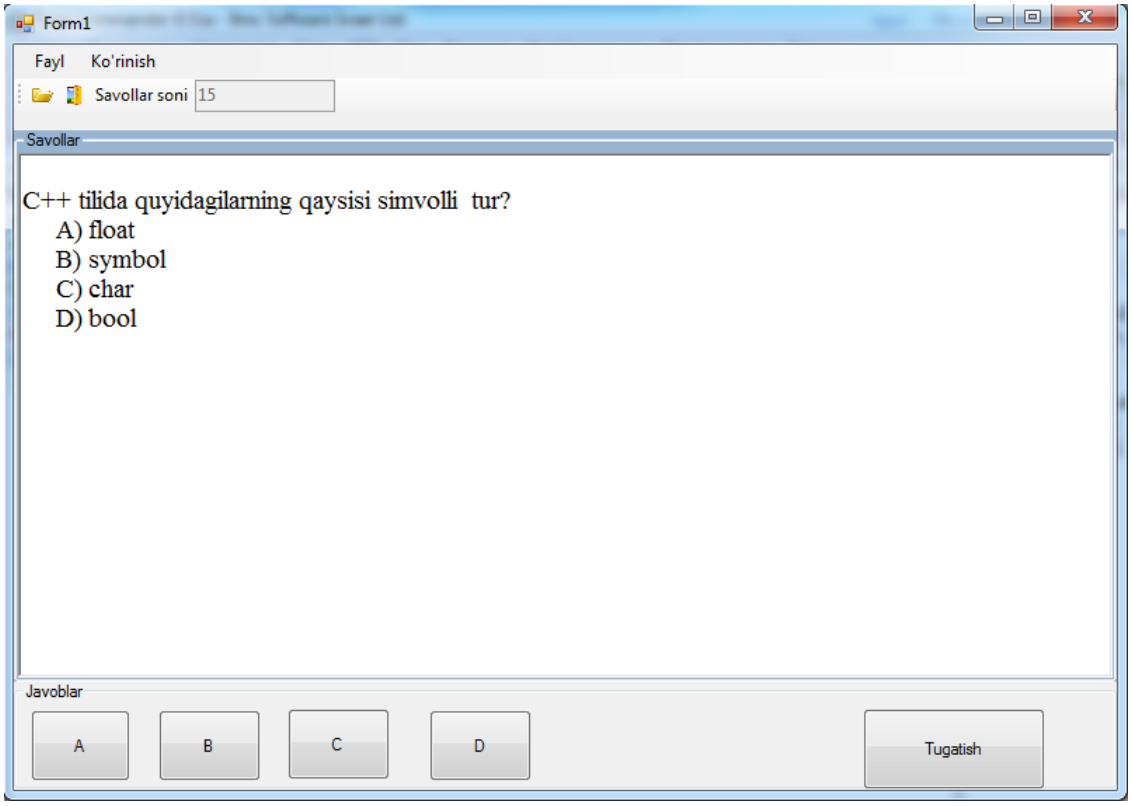
Ma’lumotlarni kiritish oynasi talaba o’zi haqidagi ma’lumotlarni kiritiladi:

- **Fakultetingizni tanlang** – bu yerda fakultetlar ro’yxati keltirilgan bo’lib o’zi o’qiydigan fakultetni tanlaydi;
- **Savollar sononi kiriting** – bunda javob beriladigan savollar soni kiritiladi;
- **Kursingizni tanlang**-bu yerda talaba kursini tanlanadi;
- **Guruhingizni kiriting**- bu bo’limda talaba guruhini kiritish kerak;
- **FISh ni kiriting**-bu bo’limda talaba familiyasi ismi va sharifini kiritish kerak;
- **Boshlash** tugma testni boshlash uchun ishlatiladi, agar yo’qoridagi bo’limlar-dan birontasi to’ldirilmasa test boshlash tugmasi ishlamaydi(4.3-rasm).

```
private void button1_Click(object sender, EventArgs e)
        {
            dannie.fakultet = comboBox1.Text;
            if (dannie.fakultet == "") { MessageBox.Show("Fakultetingizni tanlang");
goto textt; }
            ;
```

```
            if (FISH.Text == "") { MessageBox.Show("Ismi sharifingizni kiriting"); goto
textt; }
            dannie.fio = FISH.Text;
            if (kurs.Text == "") { MessageBox.Show("Kursingizni kizni tanlang"); goto
textt; }
             dannie.kurs = kurs.SelectedItem.ToString();
            if (Guruh.Text == "") {MessageBox.Show("Guruhingizni kiriting"); goto
textt;}
            dannie.k_nomer = Guruh.Text;


           if (textBox1.Text == "") {MessageBox.Show("Savollar sonini kiriting"); goto
textt;}
            dannie.soni = Convert.ToInt32(textBox1.Text);
              textBox1.Enabled = false;
            show();
            Close();
        textt: ;
        }
```

Bu joydagi ma'lumotlarni o'quvchilar to'ldirib bo'lgandan keyin "**Boshlash**" tugmasi bosiladi va test oynasi ochiladi (ma'lumotlar to'liq to'ldirilishi shart, Agar to'liq to'ldirilmasa test oynasi ochilmaydi). Test oynasi quyidagicha ko'rinishda bo'ladi(4.4-rasm):

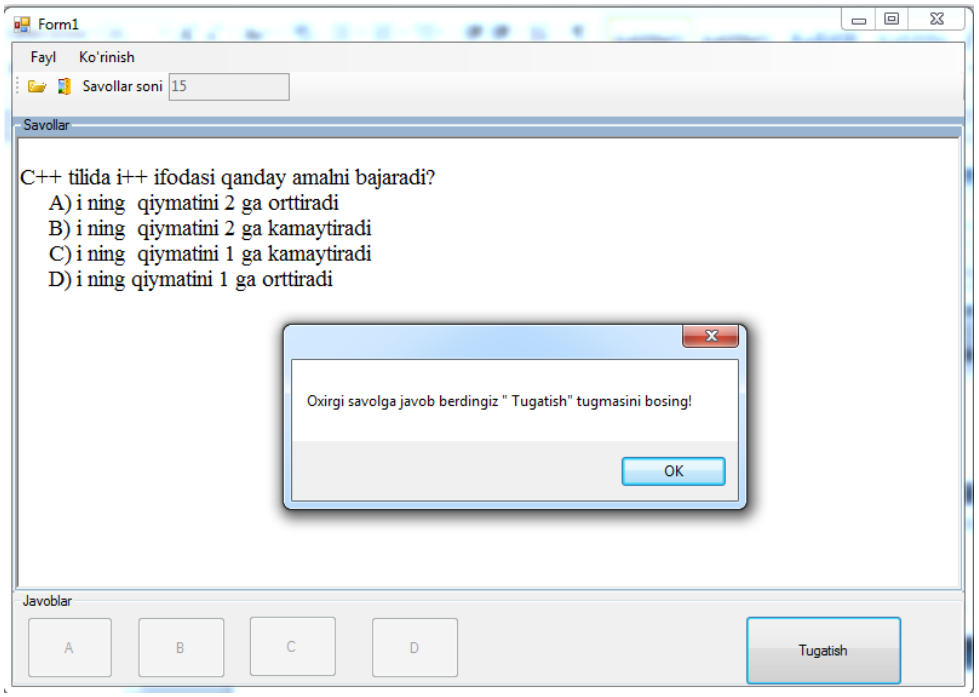


4.4-rasm. Test savollari oynasi.

Test savollari maydoni savollar matni beriladi.

Javoblar bo'limida "A", "B","C", "D" shaklida javoblar tugmalari joylashgan bo'lib, savolga to'gri ko'rsatilgan javobni tanlanadi.

Test savollarining oxirgi savoliga javob berganingizdan keyin quyidagi oyna hosil bo’ladi(4.5-rasm).



4.5-rasm. Testni tugatish oynasi.

Agar test svollarini “**Tugatish**” tugmasini tanlansa test yakunlanadi.

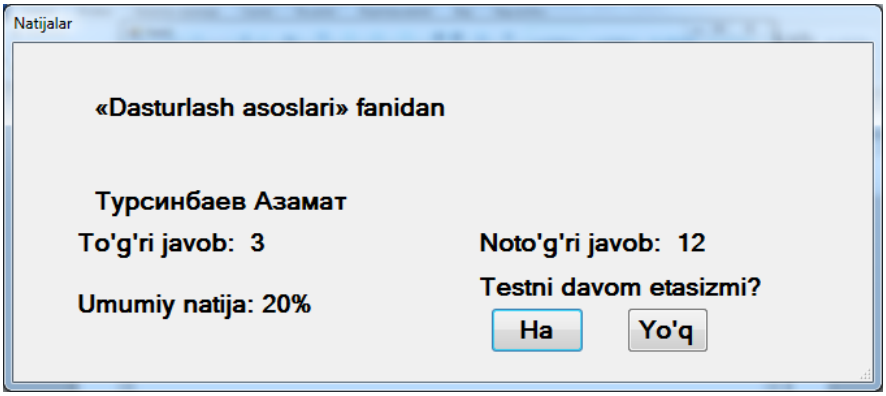
```
public void buttonlar(bool b)
 {
     button1.Enabled = b;
     button2.Enabled = b;
     button3.Enabled = b;
     button4.Enabled = b;
 }
 public void qayta() {
     try
     {
         random(indeks);
         richTextBox2.Rtf = savol[rnd[ind]];
     }
     catch { }
```

Natijalar bo’limi ko’rinishi

Natijalar oynasida quyidagilar joylashgan:

- Test topshiriladigan fan nomi;
- Talabaning familiyasi va ismi;
- To’g’ri javob;
- Oto’g’ri javob;
- Umumiy natija;
- Testni davom etasizmi?

To’g’ri javoblar soni bilan noto’g’ri javoblar soni qo’shilganda umumiy savollar soni kelib chiqadi. Umumiy **natijalar** foizda beriladi. Testni davom etasizmi-bo’limida **Ha** yoki **Yo’q** tugmalari joylashgan. Agar **Ha** tugmasi ta’nlansa testni boshlash oynasi qaytadan ochiladi, agarda **Yo’q** tanlansa test oynasida chiqib ketadi(4.6-rasm).

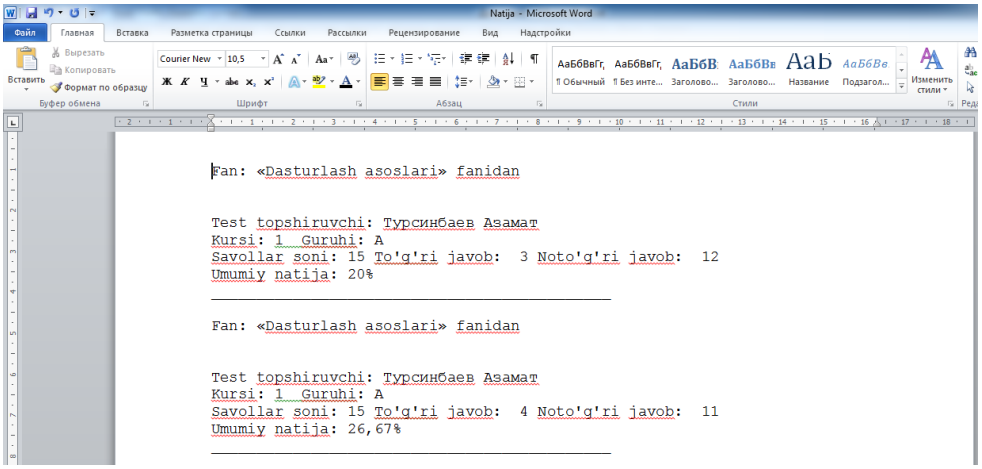


4.6-rasm. Test natijasi oynasi.

```
 public void txt_yozish() {
            StreamWriter writer = new StreamWriter("Natija.rtf", true);
            writer.WriteLine();
            writer.WriteLine("Fan: "+dannie.text);
            writer.WriteLine("Test topshiruvchi: " + dannie.fio);
            writer.WriteLine("Kursi: " + dannie.kurs+"  Guruhi: "+dannie.k_nomer);
            writer.WriteLine("Savollar soni: " + dannie.soni + " "+ label1.Text+" "
+label2.Text);
            writer.WriteLine(label3.Text);
            writer.WriteLine("_____________________________________________");
            writer.Close();
        }

        private void button1_Click(object sender, EventArgs e)
        {
           // yozish();
            txt_yozish();
            Kiritish kiritish = new Kiritish();

            this.Hide();

            kiritish.textBox1.Enabled=false;
            kiritish.textBox1.Text = dannie.soni.ToString();
            kiritish.kurs.Text =dannie.kurs;
            kiritish.Guruh.Text=dannie.k_nomer;
            kiritish.comboBox1.Text = dannie.fakultet;
            kiritish.ShowDialog();
        }

        private void label2_Click(object sender, EventArgs e)
        {

        }

        private void label3_Click(object sender, EventArgs e)
        {

        }
```

```
private void label1_Click(object sender, EventArgs e)
{

}

private void button2_Click(object sender, EventArgs e)
{
    txt_yozish();

    // yozish();
    Form GGGG = Application.OpenForms[0];
    GGGG.Dispose();
    GGGG.Close();
}

private void Form3_Load(object sender, EventArgs e)
{

}
```

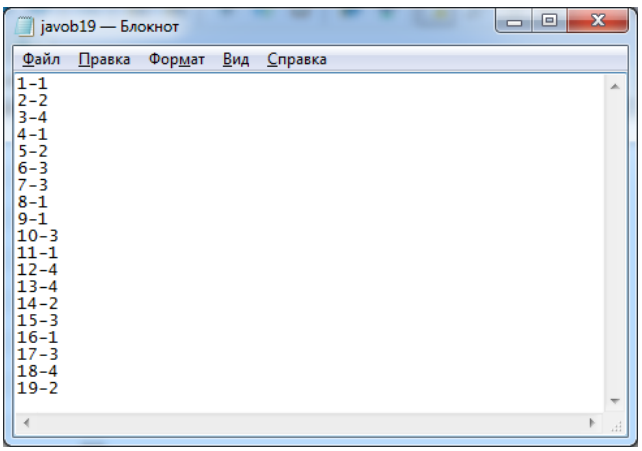
Testning natijasi **natija.rtf** faylida saqlanadi. Har bir talaba topshirgan test natijasi quyidagi shakilda saqlanadi(4.7-rasm):



4.7-rasm. Testning natijasi saqlangan fayl.

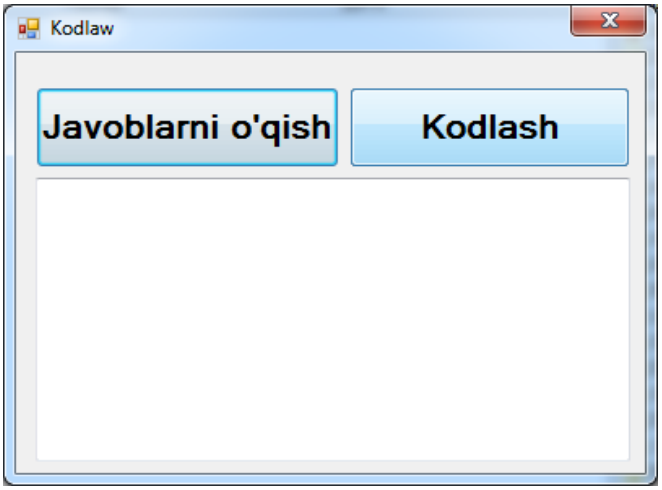
Shifrlash bo'limi ko'rinisi

Testning to'g'ri javoblarini kiritish uchun matnli ***.txt** faylda javoblar yozilgan bo'lishi kerak. Javoblarni quyidagi shaklida yoziladi(4.8-rasm):



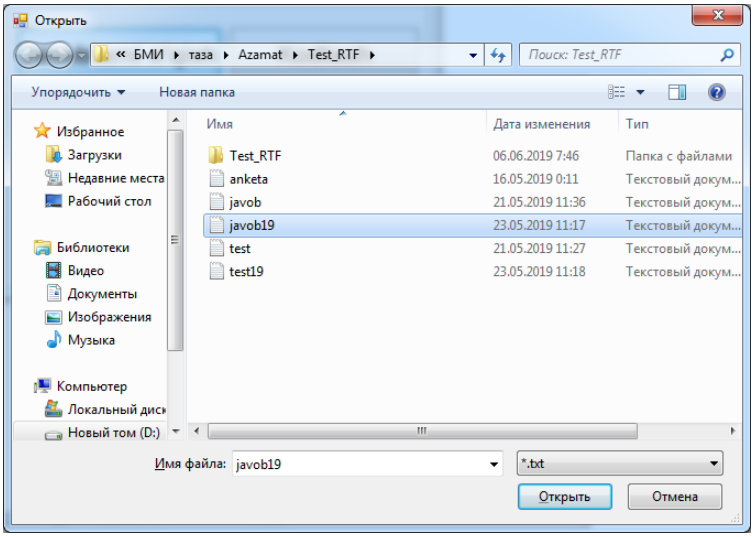
4.8-rasm. Test natijasi yozilgan matnli fayl.

Javoblarni shifrlash uchun **Kodlash** buyuruqsi ishlatiladi (4.9-rasm)



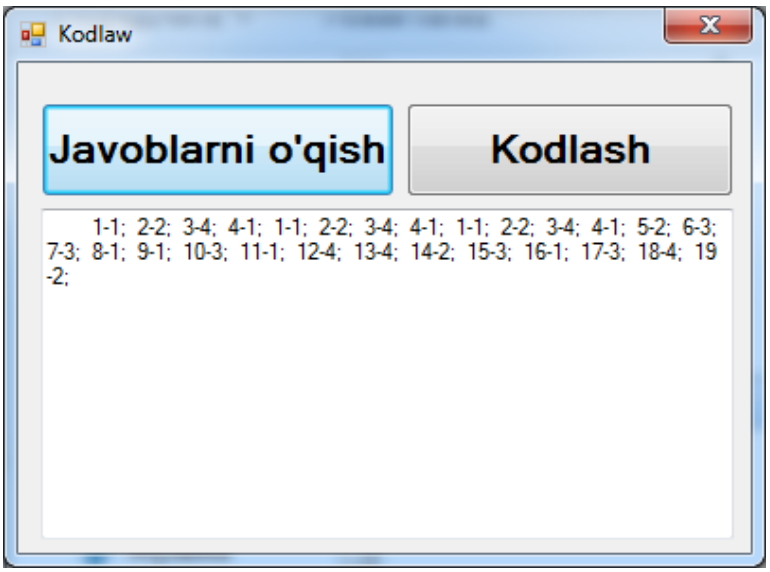
4.9-Kodlash oynasi.

Javoblarni o’qish tugmasi yordamida javoblar kiritilgan fayl tanlanadi(4.10-rasm).



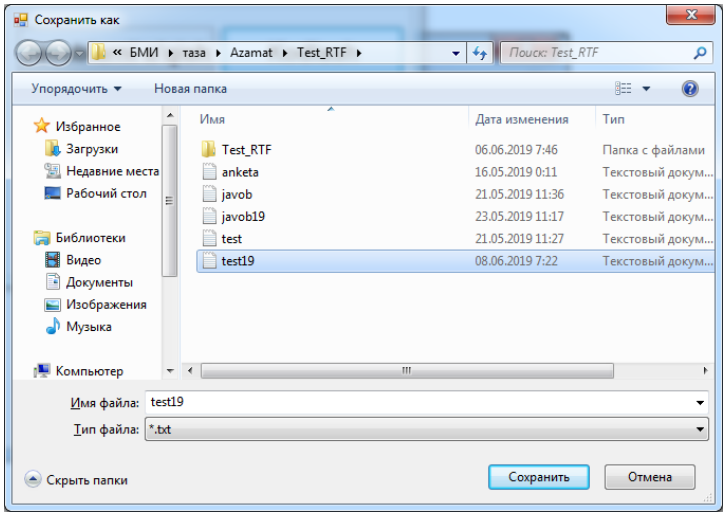
4.10-rasm. Testning to’gri javoblarini o’qish.

Javoblar kiritilgan fayl tanlangandan so’ng quyidagi oyna hosil bo’ladi. Bu yerda javoblar satr shaklida hosil boladi va bir-biridan “;” belgisi bilan ajiratilib turadi(4.11-rasm).

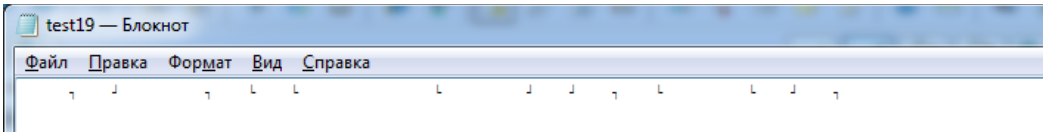


4.11-rasm. Testning to'gri javoblarni kodlashga tayyorlash.

Kodlash tugmasi javoblarni test nomi bilan bir xil nomdagi faylga shifrlab beradi.



Shifrlash natijasida to'g'ri javoblar quyidagi shaklida bo'ladi(4.12-rasm).



4.12-rasm. Kodlash natijasi

Yuqorda ko'nib turganidek test javoblarini biron bir belgi bilan almashtiradi. Bunda test tobshiruvchi savollarning javobini bilishi mumkin emas bu javoblarni faqat kompyuter o'qiy oladi. Shundan ko'rnib turibdiki test javoblari himoyalangan.

XULOSA

Bu bitiruv malakaviy ishida o’quvchilar bilimini baholashni avtomatlashtirish masalasi qaraldi. Uni amalga oshirish uchun C# dasturlashtirish tilida test dasturi tuzilishi, uning ishlash algortimi, C# tiliga foydalanishni ko’rib chiqdik.

Test dasturini yaratishdan oldin C# haqida yetarlicha tushunchalarga ega bo’lish kerak. Shundan, C# yaratilish tarixi, kelib chiqishi uning Dot.Net bilan bog’likligi haqida aytib o’tdik.

Visual C#ning komponentalari, boshqarish elementlari ularning xossalari hodisalari va ularning ish ustida qo’llanilishi bilan tanishib chiqdik. Bu dasturni tuzishdan oldin bilish kerak bo’lgan bilim va ko’nikmalar hisoblanadi. Sababi, ularni bilmasdan oldin dasturni yaratish mumkin emas.

Test dasturni tuzishdan oldin uning umumiy sxemasi, ishlash prinsiplari, ularni bir necha bo’limlarga bo’lish hamda har bir bo’limning bajaradigan vazifalariga to’xtalib o’tdik.

Har qanday dasturni tuzishdan oldin eng avval uning algoritmi tuziladi va bu dastur shu algortim bo’yicha dastur tushunadigan tilga aylantirilib dasturi tuziladi. Shundan test dasturning ishlash algoritmlari, ya’ni savollarni aralashtirish, javoblarni shifrlash, javoblarni solishtirish algortimlari haqida keltirilib o’tildi.

Test dasturni tuzishda bir necha funksiyalar va sinflar yaratdik. Funksiya va sinflar dastur hajmi kamaytiradi hamda dasturni ixcham (qulay) ko’rinishga olib keladi. Bunda bir vazifani bajarish uchun qayta- qayta dasturda kod yozib o’tirish shart emas, yaratilgan shu vazifani bajarishi yo’naltirilgan funksiyalarni chaqirish yetarli bo’ladi. Shuning uchun, test dasturdagi yaratilgan funksiyalar va ularning bajaradigan vazifalariga kengroq to’xtalib o’tdik.

Albatta, dastur yaratilgandan so’ng uni foydalanuvchilar uchun tushunarliligi, qulay tomonlarini aytib o’tish kerak. Eng so’ngi paragrafda shu masalalar qaraldi. Ya’ni bunda dastur interfeysi, har bir bo’lim haqida, savollarni o’qitish haqida, javoblarni qanday qilib shifrlash kerakligi, dastur sozlashlarini o’rnatish kerakligi haqida tanishtirib o’tdik.

Shunda ham, test dasturda Microsoft Word dokumentlarni o’qitish, test

sinovlarini o’tkazish masalalarida qaraldi.

Yakunlab aytganda bu dastur bir qancha yutuqlarga erishdi:

- ✓ O’qituvchilarning o’quvchilar bilimini baholashdagi ishini yengillashtiradi;
- ✓ O’qituvchilarning test tayyorlashda ishlaydigan ishini kamaytiradi;
- ✓ O’qituvchi vaqtini tejaydi;
- ✓ O’quvchilar o’zlarining olgan baholarini shu vaqtda bilish imkoniyatini beradi;

FOYDALANILGAN ADABIYOTLAR RO'YXATI

1. Абрамян М.Э., Visual C# на примерах. – Санкт Петербург.: БХВ - Петербург, 2008. - 496с.
2. Алекс Макки., Введение в .NET 4.0 и Visual Studio 2010 для профессионалов. - М.: Вильямс, 2010. – 416с
3. Герберт Шилдт., Полный справочник по C#. Пер. с англ.-М.: Вильямс, 2004.-752с.
4. Карли Уотсон, Кристиан Нейгел, Якоб Хаммер Педерсен, Джон Рид, Морган Скиннер., Visual C# 2010. Пер. с англ. - М.: Вильямс, 2011. - 960с.
5. Климов Л. П., Советы программистам. - СПб.: БХВ-Петербург, 2008.-544с.
6. Культин Н.Б., Microsoft Visual C# в задачах и примерах. - СПб.: БХВ - Петербург, 2009. - 320с.
7. Лабор В.В., Си Шарп: Создание приложений для Windows.-Минск.: Харвест, 2003. - 384с.
8. Торелсен Эндрю., Язык программирования C# 5.0 и платформа .NET 4.5 (6-е издание). - М.: Вильямс, 2013. - 1311с.
9. http://msdn.microsoft.com
10. http://www.wikipedia.org

ILOVALAR

(Test dastur uchun tuzilgan C# tilidagi dastur kodlari)

Test dasturining kodi

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Windows.Forms.ComponentModel;
using System.IO;
namespace Test_RTF
{


    public partial class Form1 : Form
    {
        string str;
        int count = 0;
    public Kiritish kiritish;
        string[] savol = new string[1000];
        //TextBox text1;
        public string strr;
        int [] rnd = new int [1000];
        int[] javob = new int[1000];
        int d = 0, d1 = 0, uzinliq=0;
        int indeks = 0,ind=0,t=0,x=0;
        string fs;
        bool b = true;
        public Form1()
        {
            InitializeComponent();
        }
        public void random(int max)
        {
            int k = 0,n;
            Random obrnd = new Random();
            rnd[k] = obrnd.Next(max); k++;
            while (k < max)
            {
            rand: rnd[k] = obrnd.Next(max);
                for (int i = 0; i < k; i++)
                    if (rnd[i] == rnd[k]) goto rand;
                k++;
            }

        }
        public void javoblar(string s){
         BinarReader ss=new BinarReader(File.Open(s, FileMode.Open)) ;
           int i=0;
            while (ss.PeekChar() > -1)
            {
               javob[i++]= ss.ReadInt32();
              //  textBox1.Text += javob[i-1].ToString();
            }
        }
        public void Faylni_yuklash()
        {
            try
            {
                openFileDialog1.ShowDialog();
                fs = openFileDialog1.FileName;
                str = fs.Substring(0, fs.IndexOf('.'))+".txt";
               // textBox1.Text = str;
                javoblar(str);
```

```
            richTextBox1.LoadFile(fs);kiritish = new Kiritish();
            oqiw();
            random(indeks);

           richTextBox2.Rtf=savol[rnd[ind]];

        }
        catch
        {
            MessageBox.Show("Файл юкланмади");

        }
    }

    public void buttonlar(bool b)
    {
        button1.Enabled = b;
        button2.Enabled = b;
        button3.Enabled = b;
        button4.Enabled = b;
        //button5.Enabled = b;

    }
    public void qayta() {
        try
        {
            random(indeks);
            richTextBox2.Rtf = savol[rnd[ind]];
        }
        catch { }

    }
    public void oqiw()
    {
      try
        {

            uzinliq = richTextBox1.Find("##@");
                 richTextBox1.SelectionStart = 0;

                 richTextBox1.SelectionLength =  uzinliq;

            strr= richTextBox1.SelectedText;

            dannie.text = strr;
            this.Hide();
          kiritish.ShowDialog();

          toolStripTextBox1.Text = dannie.soni.ToString();
            while (true)
            {
                d = richTextBox1.Find("##@", d, RichTextBoxFinds.MatchCase);

                d++;
                d1 = richTextBox1.Find("##@", d, RichTextBoxFinds.MatchCase);
                richTextBox1.SelectionStart = d + 3;
                richTextBox1.SelectionLength = d1 - d - 3;
                savol[indeks++] = richTextBox1.SelectedRtf;
            }

        }
        catch
        {

           // MessageBox.Show("Fayl yuklandi"+indeks);
```

```
                b = false;
            }
           ss:  if (indeks < dannie.soni)
            {

            MessageBox.Show("Savollar sonini qaytadan kiriting");
            kiritish.textBox1.Enabled = true;

               kiritish.ShowDialog(); goto ss;

            }
          //  kiritish.Close();
        }
        private void toolStripButton1_Click(object sender, EventArgs e)
        {

        }
        private void button5_Click(object sender, EventArgs e)
        {
            this.Hide();

            buttonlar(true);

            Form3 fjavob = new Form3();
            fjavob.label4.Text = dannie.text + "\n" + dannie.fio;
            fjavob.label1.Text = "To'g'ri javob:  " + t.ToString();
            fjavob.label2.Text = "Noto'g'ri javob:  " + x.ToString();
            fjavob.label3.Text = "Umumiy natija: " +String.Format("{0:0.##}", (((float) t /
(x + t)) * 100)).ToString()+"%";
            fjavob.ShowDialog();
            ind = 0;

            qayta();
            x = t = 0;
        }

        private void menuStrip1_ItemClicked(object sender, ToolStripItemClickedEventArgs e)
        {

        }

        private void uskunalarPaneliToolStripMenuItem_Click(object sender, EventArgs e)
        {
            if (uskunalarPaneliToolStripMenuItem.Checked)             panel2.Visible = true;

            else  panel2.Visible = false;
        }

        private void oqiwToolStripMenuItem_Click(object sender, EventArgs e)
        {

            Faylni_yuklash();
           // toolStripTextBox1.Text = indeks.ToString();
        }

        private void shigiwToolStripMenuItem_Click(object sender, EventArgs e)
        {
            Close();
        }

        private void richTextBox2_TextChanged(object sender, EventArgs e)
        {

        }
        public void Savol(int j)
```

```
        {
            if (j == javob[rnd[ind]]) t++; else x++;
            ind++;
            if (ind == dannie.soni) { buttonlar(false); MessageBox.Show("Oxirgi savolga
javob berdingiz \" Tugatish\" tugmasini bosing!"); goto aa; }

         //   textBox1.Text += javob[rnd[ind]].ToString();
            richTextBox2.Rtf = savol[rnd[ind]];

        aa: ;

        }
        private void button1_Click(object sender, EventArgs e)
        {

            Savol(1);
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Savol(2);

        }

        private void button3_Click(object sender, EventArgs e)
        {
            Savol(3);

        }

        private void button4_Click(object sender, EventArgs e)
        {
            Savol(4);

        }

        private void toolStripButton1_Click_1(object sender, EventArgs e)
        {
            oqiwToolStripMenuItem_Click(sender, e);
        }

        private void toolStripButton1_Click_2(object sender, EventArgs e)
        {
           oqiwToolStripMenuItem_Click(sender, e);
        }

        private void toolStripButton2_Click(object sender, EventArgs e)
        {
            Close();
        }

        private void toolStrip1_ItemClicked(object sender, ToolStripItemClickedEventArgs e)
        {

        }

        private void dsadadasdaToolStripMenuItem_Click(object sender, EventArgs e)
        {

        }


    }
}
```