

O'ZBEKISTON RESPUBLIKASI  
OLIV VA O'RTA MAXSUS TALIM VAZIRLIGI TERMIZ DAVLAT  
UNIVERSITETI

FIZIKA – MATEMATIKA FAKULTETI  
“ALAMIY MATEMATIKA” KAFEDRASI



**“Borland C++ Builder dasturlash muhitida krill va lotin imlosi yozuvlarini  
o'zaro o'giruvchi dasturiy ilova yaratish”**

BITIRUV MALAKAVIY ISHI

Bajardi: 5130200 – “amaliy  
Matematika va informatika”  
Talim yo'nalishi bitiruvchisi  
4-kurs talabasi Aliyeva D.D

---

Ilmiy rahbar: Umarzoda Sh.A

---

Bitiruv malakaviy ishi kafedradan dastlabki himoyadan o'tdi.

\_\_\_\_\_ sonli bayonnomasi “\_\_” “\_\_\_\_\_” 2018

**Termiz – 2018 yil**

## **Kirish**

### **I bob. Borland C++ Builder muhiti haqida.**

1.1. Krill va lotin imlosi yozuvlarini o'zaro o'giruvchi dastur yaratish uchun zarur algoritmlar va funksiyalar.

1.2. Krill va lotin imlosi yozuvlarini o'zaro o'giruvchi dastur yaratish jarayonida qo'llaniladigan C++ Builder komponentalari.

### **II bob. Borland C++ Builder dasturlash muhitida krill va lotin imlosi yozuvlarini o'giruvchi dasturiy vosita yaratish texnologiyasi.**

2.1. Krill va lotin imlosi yozuvlarini o'zaro o'girish jarayoni qonuniyatlari.

2.2. Krill va lotin imlosi yozuvlarini o'zaro o'giruvchi dastur kod qismi.

2.3. Dasturiy taminot interfeysini ishlab chiqish va amaliyotda qo'llash sohalari.

## **Xulosa.**

## **Adabiyotlar.**

## Mundarija

<b><u>Kirish</u></b> .....	
<b><u>I bob. Borland C++ Builder muhiti haqida</u></b> .....	
1.1. Krill va lotin imlosi yozuvlarini o'zaro o'giruvchi dastur yaratish uchun zarur algoritmlar va funksiyalar.....	
1.2. Krill va lotin imlosi yozuvlarini o'zaro o'giruvchi dastur yaratish jarayonida qo'llaniladigan C++ Builder komponentalari.....	
<b><u>II bob. Borland C++ Builder dasturlash muhitida krill va lotin imlosi yozuvlarini o'giruvchi dasturiy vosita yaratish texnologiyasi</u></b> .....	
2.1. Krill va lotin imlosi yozuvlarini o'zaro o'girish jarayoni qonunyalari.....	
2.2. Krill va lotin imlosi yozuvlarini o'zaro o'giruvchi dastur kod qismi.....	
2.3. Dasturiy taminot interfeysini ishlab chiqish va amaliyotda qo'llash sohalari...	
<b><u>Xulosa</u></b> .....	
<b><u>Adabiyotlar</u></b> .....	

## Kirish

Rivojlangan mamlakatlar qatorida O'zbekiston Respublikasida xam zamonaviy kompyuter va axborot texnologiyalari sohasida katta ishlar amalga oshirilmoqda. Fan va ta'limning barcha soxalariga infokommunikatsion texnologiyalarni keng miqyosda qo'llash, yuqori malakali dasturchi mutaxassislar tayyorlash darajasini oshirish, xalqoro axborot tizimlarini joriy etish, shu jumladan Internetdan kengaytirish maqsadida O'zbekiston Respublikasi Prezidentining 9 ta farmon va qarorlari, 18 ta hukumat qarorlari va 45 dan ortiq boshqa normativ-huquqiy hujjatlar qabul qilingan. Axborot-kommunikatsiya texnologiyalarini rivojlantirishga yunaltirilgan O'zbekiston Respublikasi Prezidentining farmoni va qarorlarini keltiramiz: 1997 yil 27 iyulda "Axborot tizimlari sohasini qayta tashkil etish va boshqarishni takomillashtirishga oid chora-tadbirlar to'g'risida" gi PF-1823-sonli farmoni, 2002-yil 30-mayda "Kompyuterlashtirishni yanada rivojlantirish va axborot kommunikatsiya texnologiyalarini joriy etish to'g'risida" gi PF – 3080 sonli farmoni, 2005 yil 2 iyunda "Axborot texnologiyalari sohasida kadrlar tayyorlash tizimini takomillashtirish to'g'risida"gi PQ-91-sonli qarori, 2005 yil 8 iyulda "Axborot-kommunikatsiya texnologiyalarini yanada rivojlantirishga oid qo'shimcha chora-tadbirlar to'g'risida"gi PQ-117-sonli qarori, 2005 yil 28 sentyabrda "O'zbekiston respublikasining jamoat ta'lim axborot tarmog'ini tashkil etish to'g'risida" gi PQ-191-sonli qarori, 2007 yil 3 aprelda "O'zbekiston Respublikasida axborotni kriptografik muhofaza qilishni tashkil etish chora-tadbirlari to'g'risida"gi PQ-614-sonli qarori, 2012 yil 21 martda "Zamonaviy axborot-kommunikatsiya texnologiyalarini yanada joriy etish va rivojlantirish chora-tadbirlari to'g'risida"gi PQ-1730-sonli qarori, 2013 yil 20 sentyabrda "Mamlakatimizning dasturiy ta'minot vositalari ishlab chiquvchilarini rag'batlantirishni yanada kuchaytirish chora-tadbirlari to'g'risida"gi PQ-2042-sonli qarori, 2013 yil 25 sentyabrda "Milliy geografik axborot tizimini tashkil etish investitsiya loyihasini amalga oshirish chora-tadbirlari to'g'risida"gi PQ-2045-sonli qarorlari bu boradagi ishlarni yanada rivojlantirish imkonini berdi.

O'zbekiston Respublikasi birinchi Prezidentining "Malakali pedagog kadrlar tayyorlash hamda o'rta maxsus, kasb-xunar ta'limi muassasalarini malakali kadrlar bilan ta'minlash tizimini yanada takomillashtirishga oid chora-tadbirlar to'g'risida" gi qarorida axborot kommunikatsiya texnologiyalari, elektron ta'lim resurslari va multimediya taqdimotlaridan foydalanish borasidagi chet el tajribasini chuqur xar tomonlama o'rganib chiqish hamda ta'lim jarayonida qo'llash xaqida aloxida ta'kidlab o'tilgan. Zamonaviy axborot va kommunikatsiya texnologiyalari o'qitishning yangi shakl va metodlarini ishlab chiqish uchun keng imkoniyat yaratildi.

Kompyuter, dasturiy mahsulotlarsiz foydalanuvchi uchun hech qanday qiymatga ega bo'lmaydi. Foydalanuvchi uchun aynan dasturiy mahsulotlar shunday afzalliklar va qulayliklar beradiki, ular tufayli biz o'z ishimizni tezroq, aniqroq, chuqurroq va nihoyat, samaraliroq bajaramiz. Shuning uchun dasturiy mahsulotlarni tanlash, kompyuterdan foydalanuvchilar uchun, ayniqsa dasturiy mahsulotlar bozorida foydalanuvchilarga yuzlab dastur mahsulotlaridan birini tanlash imkoniyati taklif etilayotgan va u tanlangan dasturga ko'nikib qolgandan keyin kelajakda undan voz kechish qiyin bo'ladigan hozirgi paytda, juda muhimdir.

Amaliy dasturlar orasida tilshunoslik tarjimalarini amalga oshiruvchi dasturiy mahsulotlar alohida o'rin egallaydi. Kompyuterlar tomonidan bajariladigan tarjimalar tarixi AQSH ning Djordjtaun universitetida 1954 yilda o'tkazilgan tadqiqotga borib taqaladi. O'shanda jahonda ilk bor matn (to'g'ri, bir nechtagina jumladan iborat) rus tilidan ingliz tiliga EHM yordamida tarjima qilingan edi. Bugungi kunda turli mamlakatlarda tarjima bo'yicha kamida yuzlab tijoriy kompyuter tizimlari ma'lum va mashhur, tadqiqiy tajriba tizimlari esa juda ko'p. Davriy nashrlarda tez-tez kompyuter tarjimasi tizimlarini qo'llash masalalari ko'tarilmoqda, ularning tavsifnomalari berilmoqda, tilshunoslik tarjimalarini amalga oshiruvchi dasturiy mahsulotlar taqqoslanmoqda.

Bizga ma'lumki kompyuter dasturlarida matn tarjimasida muammolarga duch kelamiz, yani matn tarjimasida matnning ma'no mazmuni buziladi. Buni bartaraf etish muhim ahamiyatga ega.

Bizga ma'lumki dasturlash tillarini o'qitish bir muncha qiyinchiliklardan iboratdir. Bu qiyinchiliklarni bartaraf etish uchun biz interfaol metodlardan foydalanishimizga to'g'ri keladi. Zamonaviy pedagogik texnologiya o'zining pedagogika va boshqa fan yutuqlari bilan bog'liq xususiy nazariyasiga ega: u birinchi galda o'quv-tarbiyaviy jarayonni ilmiy asosda qurishga yo'naltirilgan, o'qitishning axborotli vositalaridan, didaktik materiallardan, faol metodlardan keng foydalanishga asoslangan o'qituvchi va talabalarning birgalikdagi faoliyatiga zamin yaratadi.

Hozirgi yetishib kelayotgan yosh avlod uchun dasturlash tillarini o'rganish va uni amalda tadbiq etish dolzarb vazifalardan biridir. Albatta bu dolzarb vazifalarni bajarishda dasturlash tillarini o'rgatuvchi va o'z bilimini tekshiruvchi tizimlarning o'rni judayam muhim. Dasturash tillarini o'rganishda judayam ko'plab o'quv qo'llanmalar, kitoblar, monografiyalar va uslubiy ko'rsatmalar ishlab chiqilgan. Bularni o'zlashtirishda o'quvchidan sabr, iroda va matonat talab etiladi. O'quvchiga bularni tezda o'rganishda zamonaviy pedagogik texnologiyalarni qo'llash zamon talablaridan biridir. Dars samaradorligini oshirishda o'qituvchi avvalo maqsad

belgilab olishi, noan'anaviy va an'anaviy darslarning farqini ko'ra olishi shart. Noan'anaviy darslardan maqsad talabalarning o'zlashtirish ko'rsatkichini ko'tarish uchun, fanni o'qitishda yanada mazmunli, qiziqarli va tushunarli o'tkazishda o'qituvchi va talaba faoliyatiga yangilik kiritib, interfaol metodlardan foydalanish hamda talabalarni tez va samarali baholashga erishishdir. Shu bilan birga noan'anaviy darslarda o'quv jarayonida talabalar mustaqil fikrlay olishga, ijodiy yondashishga, izlanishga, tahlil eta olishga, o'zlari xulosa qilib, o'z-o'zini hamda guruhni va guruh talabani baholay olishga o'rgatadi. O'qituvchi esa ularning bu faoliyatiga imkoniyat va sharoit yaratib bera olishi kerak. Bu esa o'qitish jarayonining asosi hisoblanadi.

Dasturlashni o'qitish uchun ta'lim oluvchilarni birinchi navbatda guruhlariga ajratib olamiz:

1-guruh. Boshlovchilar (maktabning 7-9 sinf o'quvchilari). Bu bosqichdagi o'quvchilarning yoshini hisobga olgan holda quyidagi o'qitish usullarni qo'llashimiz mumkin: Puzzle (jumboq) – usuli. Bu usulda o'qituvchi to'g'ri tuzilgan dasturni oladi hamda uni satrlarini o'rnini almashtirib yuboradi va o'quvchilarga taqdim etadi. O'z navbatida o'quvchilar dasturning satrlarini to'g'ri dastur ko'rinishiga olib kelishlari kerak bo'ladi; Hand tracing (qadamlarni tekshirish) – usuli. Bu usulda o'qituvchi biror to'g'ri tuzilgan dasturni olib uni qadamma-qadam bajarilish jarayonini hamda o'zgaruvchilarning qiymatlarini o'zgarishini jadval ko'rinishida ko'rsatib beradi.

Bu usulning yutuqli tarafi o'quvchilar dasturni tushunishlari oson va unga qiziqishlari ortadi.

2-guruh. O'rta guruh (Akademik lisey va kasb-hunar kollejlari o'quvchilari). Ushbu guruhda o'quvchilar dasturlashdan xabardor bo'ladi. Bu guruh yoshidagilar uchun quyidagi usullarni qo'llash maqsadga muvofiq: Teamwork (jamoaviy ishlash) – usuli. Ushbu usulda o'quvchilar kichik guruhlariga bo'linadi hamda har bir guruhga sardor tayinlanadi. O'qituvchi tomonidan kichik loyiha yoki masala o'rta tashlanadi va har bir jomoa o'z usulida yondashib natijaga erishishi lozim bo'ladi. Work in pairs (juft bo'lib ishlash) – usuli. Bu usulda o'quvchilar juft-juft bo'lib bo'linishadi va o'qituvchi tomonidan beriladigan topshiriqni birgalikda bajarishadi. Yuqoridagi usullarning ikkisi ham o'quvchilarni jamoaviy ishlash va o'z fikrini jamoadoshlariga yetkaza olish qobiliyatini shakllantiradi. Undan tashqari dasturlashga bo'lgan qiziqishlari yanada ortadi.

3-guruh. Ilg'or guruh (Oliy ta'lim muassasalari talabalari). Ushbu guruh juda muhim xisoblanib bunda talabalarni mustaqil ishlash qobiliyatini shakllantirish lozim bo'ladi. Bu guruh ta'lim oluvchilariga qisman yuqorida ta'kidlab o'tilgan

usullarni ham qo'llash mumkin. Ammo "Automata approach" (Avtomatlashtirilgan yondashuv) usuli ko'proq samarali bo'ladi. Bu usulda talabalar berilgan topshiriqlarni avtomatlashtirilgan tizimlar yordamida tekshiradi va eng to'g'ri yechimga erishish darajasiga yetadi. Bunday tizimga <http://codeforces.com/> sayтини misol qilib keltirishimiz mumkin.

Dasturlash yoki kod yozish deb odatda kompyuterga ba'zi operatsiyalarni qanday bajarishni buyurish tushuniladi. Dasturlash o'z ichiga ko'plab matematik hisob-kitob va strukturani olsada, aslida u yuqori darajada ijodiy ish hisoblanadi: bir qator dastur kodi bilan bir vaqtda mahsulotning tayyor holati haqida o'ylaysiz. Dasturlashni o'qitishda ko'p mamlakatlar dasturiy yechim to'g'riligini avtomatik testlovchi tizimlardan foydalanilgan holda o'tishadi. Hozirgi dasturlash bo'yicha jahon olimpiadalarida Rossiya mamlakatidan qatnashuvchilar doimo yuqori natijalarni ko'rsatib kelishmoqda. Ular bunday natijalarga qanday erishishadi? Quyida biz unga javob olishimiz mumkin. Rossiyada dasturlash tillari maktab davridan boshlanadi. Ular maktab ta'limida 7-sinfdan boshlab dasturlash tillarini o'qitishni boshlashadi. Ularning ta'lim tizimi barcha mamlakatlarning ta'lim tizimi bilan bir xil bo'lishiga qaramasdan ularning yutuq jihatlari nimada ekan deb o'ylashimiz mumkin. Ularning asosiy yutuqlaridan biri shundaki ular dasturlash tillarini o'qitishni boshlashgandan so'ng ma'lum bir vaqt oralig'i(masalan, har 1 oy)da online kontestlar o'tkazib turishadi. Bu online kontestlar ikki ko'rinishda yakka tartibda yoki komandaviy ko'rinishda bo'ladi. Har bir kontest tugatilgandan so'ng natijalar tahlil qilinadi va har bir masala muhokama qilinadi. Bu usul ularning eng yutuqli jihatlari hisoblanadi. Ular o'tkazadigan online kontestlar komandaviy holatda ham ko'p bo'lib turishi ularning dasturlash bo'yicha xalqaro komandaviy olimpiadalarda yuqori natija olishlariga tayorgarlik bo'ladi. Rossiyaning dasturlashni o'qitish bo'yicha yana bir yutuqli jihati shundaki ular bir yil davomida 2 marotaba Yozgi va Qishki lagerlar tashkil qilinishidir. Bu lagerda ma'lum kun tartibi bo'yicha ish ko'riladi. Har kuni ertalab kontest o'tkaziladi va kontest yakunlangandan so'ng bu masalalar muhokama qilinadi. Masalani muhokama qilishda barcha qatnashadi, yani bir kishi biror masalani muhokama qiladi. Agarda bu masalaga boshqa kontestchida boshqacha qulayroq va samaraliroq usul bo'lsa u ham chiqib o'z fikrini bildirishi mumkin. Bu usul orqali barcha lager qatnashchilari bir masalani yechish uchun bir nechta usullarni bilib olishadi. Lager davomida lagerchilar faqatgina tayorgarlik emas balki ular uchun qo'shimcha mashg'ulotlar, sport bilan ham shug'ullanishadi. Rossiya dasturlash bo'yicha xalqaro olimpiadalarda qatnashishi bo'yicha yuqori o'rinlarda turadi. Ular asosan ACM ICPC olimpiadasida yuqori natijalarni ko'rsatib kelishmoqda. Association for Computing Machinery (ACM) bu Hisoblash mashinalar uyushmasidir. International Collegiate Programming Contest (ICPC) bu dunyodagi universitet talabalari

oʻrtasidagi dasturlash boʻyicha xalqaro olimpiada hisoblanadi. Xitoy davlati bizga maʼlumki rivojlangan davlatlardan biri hisoblanadi. Bu davlatda ham dasturlashga juda katta eʼtibor beriladi. International Olympiad in Informatics (IOI) bu dasturlash boʻyicha maktab oʻquvchilari oʻrtasidagi xalqaro olimpiada boʻlib, bu olimpiadada Xitoy medallar soni boʻyicha birinchi oʻrinda turadi(1-rasm). Endi dasturlashni yurtimizda qanday oʻtilishiga toʻxtalib oʻtadigan boʻlsak. Oʻquvchilarga dasturlash tillari 9-sinfdan boshlab oʻtila boshlanadi. Respublikamizning har bir viloyatlarida har xil dasturlash tillari oʻtiladi. Masalan, Basic, Paskal, Delphi – dasturiy vosita, C/C ++, Java va boshqa dasturlash tillari oʻtiladi. Ular har xil dasturlash tillarida oʻz darslarini olib borishadi. Lekin ularning dasturlash tillarida olib boradigan dasturlari deyarli bir xil. Yaʼni bir-biriga yaqin saviyadagi masalalarni ishlashadi. Maktab oʻquvchilari oʻrtasi(9-sinf)da Informatika (dasturlash) fani boʻyicha Tuman, viloyat fan olimpiadalari oʻtkazilib kelinmoqda. Olimpiadadan faxrli oʻrinni egallagan oʻquvchilar akademik litsey va kasb hunar kollejlari imtiyozli davlat test markazi imtixonini topshirmagan holatda oʻqish imkoniyatini qoʻlga kiritadilar. Akademik litsey va kasb hunar kollejlari ham dasturlash fanlari oʻtilib ular oʻrtasida ham tuman, viloyat va Respublika fan olimpiadalari oʻtkazilib kelinadi. Undan tashqari Akademik litsey va kasb hunar kollejlari oʻquvchilari oʻrtasida Informatika va dasturlash boʻyicha Butun Rossiya maktab komandalari oʻrtasidagi olimpiadada qatnashishlari mumkin boʻladi. Bu olimpiadada Oʻzbekiston yoshlari 2006-yildan beri qatnashib kelishmoqda. Bu olimpiadada Oʻzbekiston yoshlarining natijalari unchalik qoniqarli boʻlmagan boʻlsada yildan yilga ularning natijalari yuqorilab bormoqda. 2014-yilgi olimpiadada esa Oʻzbekistonning Buxoro davlat universiteti qoshidagi 3-sonli Qorakoʻl akademik litseyi 3 nafardan iborat komandasi Sharipov Abdulla, Jabborov Ahror, Atoyev Abubakir 3-darajali diplom bilan taqdirlanishgan.

	Country	IOI Host	Medals ▼			
			G	S	B	Total
	China	2000	75	24	12	111
	Russia	2016	55	33	12	100
	United States of America	2003	45	34	15	94
	Poland	2005	36	37	28	101
?	Republic of Korea	2002	35	35	26	96
?	Romania		28	45	26	99
?	Bulgaria	1989, 2009	24	41	31	96
	Slovakia		24	37	27	88
	Iran		21	49	22	92
	Taiwan	2014	18	43	24	85
	Japan		18	19	8	45
?	Thailand	2011	15	33	44	92
?	Belarus		14	34	35	83
?	Germany	1992	14	26	37	77

1-rasm.

Davlatlarning IOI da olgan medallar soni IOI olimpiadasida Oʻzbekiston yoshlari 2015-yildan boshlab qatnashib kelishmoqda. 2017-yil yanvar oyida Qozogʻistonda oʻtkazilgan dasturlash boʻyicha olimpiadaga Oʻzbekistondan 2 nafar oʻquvchi borib qatnashib kelishdi. Bular, Toshkent axborot texnologiyalari



universiteti qoshidagi 2-akademik litsey 2-kurs o‘quvchisi Xo‘jayev Temur va Namangan davlat universiteti qoshidagi 2-akademik litsey 3-kurs o‘quvchisi Hasanov Husaynlardir. Bu olimpiadada Xo‘jayev Temur faxrli qatnashib “Bronza” medalini qo‘lga kiritib qaytdi. Bu maktab o‘quvchilari o‘rtasida dasturlash bo‘yicha xalqaro olimpiadadagi birinchi yutug‘imiz bo‘ldi. codeforces.com saytida online musobaqalar bo‘lib turadi. Bu tizim orqali har bir foydalanuvchiga reyting beriladi. Bu reytinglar kesimida mamlakatlar reytingini ham ko‘rishimiz mumkin. Hozirda O‘zbekiston bu reytingda 43-o‘rinni egallab turibdi.

Rank	Name	Member count	Rating
1	Russia	1693 (5222)	3109 (3117)
2	China	1778 (6108)	3027 (3042)
3	Japan	289 (866)	2806 (2842)
4	Korea, Republic of	262 (489)	2800 (2819)
5	Belarus	274 (806)	2787 (2813)
6	Ukraine	378 (1245)	2755 (2779)
7	Poland	170 (407)	2751 (2770)
8	United States (USA)	298 (878)	2596 (2746)
9	Taiwan	140 (375)	2579 (2672)
10	Romania	243 (594)	2512 (2550)
11	Croatia	81 (156)	2499 (2535)
12	Iran	395 (1345)	2470 (2531)
13	Switzerland	17 (48)	2434 (2522)
13	Australia	35 (73)	2434 (2445)
15	Canada	72 (182)	2398 (2422)
16	Korea, DPR	18 (45)	2382 (2404)
17	Vietnam	450 (1209)	2381 (2401)
18	Brazil	286 (610)	2364 (2394)

2-rasm.

codeforces.com saytida mamlakatlar reytingi Agarda biz dasturlashni o‘qitishda jahon tajribasidan foydalanishni yo‘lga qo‘ysak bizda ham rivojlanish sezilarli darajada o‘zgaradi.

**Ilmiy tahmin.** Elektron hujjatlar bilan ishlovchi yuridik va jismoniy shaxslarda Microsoft Office dasturlar paketidan foydalanishda krill va lotin alifbosi munosabatlari orqali yuzaga keluvchi muommolarni qisqa vaqt oralig‘iga bartaraf qilinadi.

### **Bitiruv malakaviy ishning maqsadi.**

– ta’lim bo‘yicha nazariy va amaliy bilimlarni mustahkamlash va kengaytirish, olingan bilimlarni muayyan ilmiy, texnikaviy, ishlab chiqarish, iqtisodiy, ijtimoiy, madaniy vazifalarni hal etishda ijodiy ishlash, hal etilayotgan masalaning (muammoning) qo‘yilish jarayonidan boshlab, uni nihoyasiga etkazish bo‘yicha qaror qabul qilishda bo‘lgan mas’uliyatni his etishga o‘rgatish;

– zamonaviy ishlab chiqarish, iqtisodiyot, texnika va madaniyatning rivojlanishi sharoitida talabalarni mustaqil ishlashga tayyorgarligini ta’minlashdan iborat.

- Microsoft Office dasturlar paketidan doimiy foydalanuvchilar uchun, hozirgi kunda ko'p uchrayotgan muammolardan biri bu krill va lotin alifbosi orasidagi munosabat bo'lib, Word, Excel va boshqa amaliy dasturlarda tayyorlangan elektron hujjatlarni krill alifbosidan lotin alifbosiga yoki aksincha lotin alifbosidan krill alifbosiga o'g'irishdan iborat.

**Bitiruv malakaviy ish vazifalari.** Microsoft Office dasturlaridan foydalanishda alifbolar orasidagi o'girish jarayonida hosil bo'ladigan muommoni hal qilish.

**Bitiruv malakaviy ish ob'ekti.** Tadqiqot ob'ekti etib – yuridik va jismoniy shaxslar tomonidan elektron hujjatlar bilan ishlash jarayoni olingan. Chunki mehnat samaradorligini oshirishda vaqtni tejash jiddiy ahamiyatga ega.

**Bitiruv malakaviy ish predmeti.** Tadqiqot ishimizda BMI predmeti bo'lib, C++ Builder 6 dasturida foydalanuvchilar uchun amaliy dastur ishlab chiqish, metodlari va vositalari hisoblanadi.

**Tadqiqot metodlari.** Nazariy (pedagogik-psixologik va o'quv-metodik adabiyotlar (Oliy oquv yurtlari dasturlash asoslari darsliklari), o'quv-metodik qo'llanmalar, me'yoriy hujjatlar (yillik mavzuviy-taqvimiy rejalari va uzviylashtirilgan davlat ta'lim standarti va o'quv dasturi, ixtisoslashtirilgan davlat umumta'lim maktab-internat hamda ayrim fanlar chuqurlashtirilib o'qitiladigan yo'nalishlar uchun dasturlash fanidan uzviylashtirilgan o'quv reja va dasturlar), ta'lim jarayoni uchun mo'ljallangan pedagogik dasturiy vositalarni tahlil qilish. Amaliy (talabalar va o'qituvchilar bilan suhbatlar o'tkazish, talabalarina amaliy dasturlar yaratish jarayonida ilg'or pedagogik tajribalarni o'rganish va umumlashtirish, natijalarni tahlil qilib pedagogik dasturiy vositalardan o'quv jarayonida foydalanish).

**Tadqiqotning metodologik asosini** «Ta'lim to'g'risida»gi Qonun, «Kadrlar tayyorlash Milliy dasturi», O'zbekiston Respublikasi ta'lim tizimini axborotlashtirishni rivojlantirish bo'yicha qabul qilingan direktiv va me'yoriy hujjatlar, pedagogika, informatika va axborot texnologiyalari sohasidagi taniqli olimlarning ilmiy ishlari tashkil etadi.

**Ilmiy yangiligi.** Dasturiy vositalar ishlab chiqarish tizimini zamon talabi darajasida tashkillashtirish dasturchidan ko'proq mehnat talab qilib, ishlab chiqarishni sifatli bajarishni taqozo etadi. BMIda:

- talabalarning bilimlarini mustahkamlash, o'zlashtirish jarayonini faollashtirishga xizmat qiluvchi, kursga oid mavzularni o'rgatishning samarali yo'llari ishlab chiqildi;

- amaliy dasturlar ishlab chiqarish jarayonini pedagogik dasturiy vositalar asosida tashkil etish, ta`lim tizimining zamonaviy talablariga mosligi ilmiy nuqtai nazardan asoslab berildi;

- alaliy dasturlar ishlab chiqarish jarayoni bo'yicha pedagogik dasturiy vosita ishlab chiqildi va ularning ishlab chiqarish jarayonidagi ahamiyati ilmiy asoslab berildi;

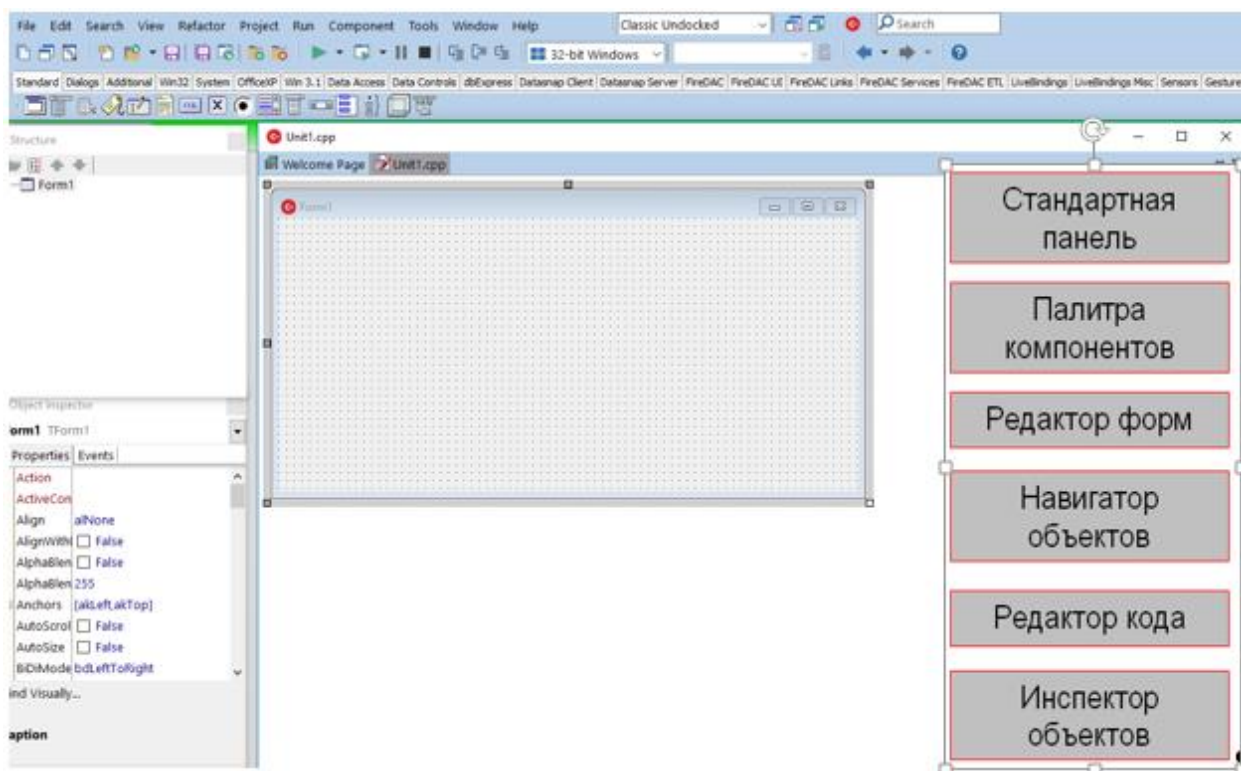
- pedagogik dasturiy vositalarni qo'llash orqali, malum bo'lgan muammolarni amaliy jihatdan bartaraf etuvchi dasturlar, mehnat jarayonini intensivlashtiruvchi va optimallashtiruvchi imkoniyatlari ko'rsatildi.

**Bitiruv malakaviy ish natijalarining ilmiy va amaliy ahamiyati.** Microsoft Office dasturlar paketidan doimiy foydalanuvchilar uchun, hozirgi kunda ko'p uchrayotgan muammolardan biri bu krill va lotin alifbosi orasidagi munosabat bo'lib, Word, Excel va boshqa amaliy dasturlarda tayyorlangan elektron hujjatlarni krill alifbosidan lotin alifbosiga yoki aksincha lotin alifbosidan krill alifbosiga o'g'irish bo'lib, bu krill alifbosida tayyorlangan hujjatlarni yana qaytadan lotin alifbosida tayorlashni talab qilinadigan vaziyatlardir. Quyidagi muommoni hal qilish uchun ko'pgina amaliy dasturlar va Microsoft Office amaliy dasturida VBA tilida yozilgan makroslar mavjud bo'lib ularda bazi kamchiliklar kuzatiladi. Bitiruv malakaviy ishda tavsiya qilingan Borlan C++ Builder 6 dasturlash muhitida yaratilgan dastur bitiruv malakaviy ishda yoritilgan muammoli holatlardan holidir.

**Ishning tuzilishi.** Bitiruv malakaviy ish kirish qismi, asosiy qism (I-bob, II-bob,), xulosa va takliflar, foydalanilgan adabiyotlar ro'yxati va internet saytlardan iborat.

### I bob. Borland C++ Builder muhiti haqida.

**C++ BUILDER bilan dastlabki tanishuv.** Ishlab chiqishning integratsiyalashgan muhiti Komponentlar palitrasini birlashtiradi. Shakllar Muharriri, Kod Muharriri, Ob'ektlar Noziri, Ob'ektlar Xazinasi - bular hammasi kod va zahiralardan ustidan to'liq nazoratni ta'minlovchi dasturiy ilovalarni tez ishlab chiqish instrumentlari(3-rasm).



3-rasm. Ishlab chiqish muhitining tuzilishi

■ **Komponentlar Palitrasi** ilovalarni qurishda taklif qilinadigan 100 dan ortiq takroran qo'llanadigan komponentlardan iborat.

■ **Shakllar Muharriri** dasturning foydalanuvchi bilan interfeysini yaratish uchun mo'ljallangan.

■ **Kod Muharriri** dastur matnini, xususan, voqealarga ishlov berish funksiyalarini yozish uchun mo'ljallangan.

■ **Ob'ektlar Noziri** qotib qolgan chigal dasturlash zaruratisiz ob'ektlar xususiyatlarini vizual o'rnatish imkonini beradi hamda shunday voqealarni o'z ichiga oladiki, bu voqealarni ularning paydo bo'lishiga nisbatan ob'ektlar reaksiyasi kodlari bilan bog'lash mumkin bo'ladi.

■ **Ob'ektlar Xazinasi** ma'lumotlarning shakl va modullari kabi ob'ektlarga ega bo'lib, ular ishlab chiqishda muvaqqat sarflarni kamaytirish maqsadida ko'plab ilovalar bilan bo'linadi.

C++Builder ilovalarni qurishning vizual metodikasini Komponentlar Palitrasidan kerakli boshqarish elementlarini tanlab olish vositasida joriy etadi. Har

bir komponenta (masalan, tugmacha) bilan ushbu komponenta turini va xulq-atvorini o'zgartiradigan xususiyatlar bog'liq bo'ladi. Har qanday komponenta ushbu komponentaning turli xildagi ta'sirlarga reaksiyasini (munosabatini) aniqlab beradigan voqealar seriyasini keltirib chiqarishi mumkin. Bunday keyin => belgilari siz C++Builder muhitida amalga oshiradigan xatti-harakatlarni bildiradi.

=>C++Builder ni chaqiring va bosh menyudagi File | New Application komandasi bo'yicha yangi ilovalar ustida ishlashni boshlang.

=>sichqonchani Komponentalar Palitrasining qo'shimcha ilovalari ustida bosib, foydalanuvchi ish ko'radigan dastur interfeysi elementlarining mavjud assortimentini ko'rib chiqing.

Palitraning bir qo'shimcha ilovasidan ikkinchisiga o'tib, kirish mumkin bo'lgan komponentlar to'plami o'zgarayotganining guvohi bo'lishimiz mumkin. Sichqoncha kursori komponentlar belgisi ustida to'xtaganda, aytib turish nomi paydo bo'ladi. Agar F1 klavishasini bossak, tizimning ma'lumotnomalar xizmati tanlab olingan komponenta haqida to'liq ma'lumot chiqarib beradi.

**Vizual loyihalas:** Bizning birinchi ilovamiz bolalarning «O'nta negr bolasi» sanoq she'rini generatsiya qiladi. Dastlabki versiyada faqat uchta ob'ekt kerak bo'ladi: ro'yxat, tahrir qilish maydoni va tugmacha. Komponentalarni loyihalash shakliga olib o'tamiz hamda ilovani asta-sekin rivojlantira boshlaymiz. Tashib olib o'tish metodi (drag-and-drop) quyidagilardan iborat: sichqoncha tugmachasini tanlab olingan komponenta ustida bosing, kursorni shaklning to'g'ri kelgan eriga o'tkazing, keyin esa sichqoncha tugmachasini yana bosing. Boshida faqat «standart» Palitra Komponentlari bilan cheklanamiz:

=> Standard qo'shimcha ilovani tanlab oling.

=> Ro'yxat komponentasini ListBox shakliga olib o'ting.

=> Tahrir qilinatgan kiritish maydoni EditBox ni olib o'ting.

=> Button tugmachasi komponentasini olib o'ting.

=>Komponentalarni o'zingizning ilovangizdagi darchada qanday ko'rmoqchi bo'lsangiz, shunday joylashtiring va o'lchamlarini shunday o'zgartiring.

Ob'ekt Noziri yordamida komponentalar xususiyatlarining boshlang'ich qiymatlarini aniqlang. Items ro'yxatining xususiyatlar qiymatlari katagida tugmachani bosing, ochilgan muharrir darchasida she'rning dastlabki 7 satrini kiriting. SHakl va tugmachaning Caption xususiyatida ularning ma'noli nomlarini ko'rsating (mos ravishda, «O'nta negr bolasi» va «Natija»). Tahri qilish

maxdonining Text xususiyatida natijani aytib berish satrini kiriting («To‘qqizta negr bolasi»).

Endi Kod Muharririga ulanish hamda, avval qabul qilinganidek, C++tilidagi har qanday dasturni yozish mumkin, shu jumladan, ANSI/ISO standartining so‘nggi kengaytmalarini ham. Biroq, avval ilovalarni tez ishlab chiqishning yangi vositalari hamda C++Builder da mavjud bo‘lgan qo‘shimcha komponentalar atributlaridan foydalanishga harakat qilib ko‘ramiz.

### ***Ikki yo‘nalishli ishlanma texnologiyasi***

#### ***Loyihaviy shablonlarni qo‘llash***

#### ***Xususiyatlar, metodlar va voqealar:***

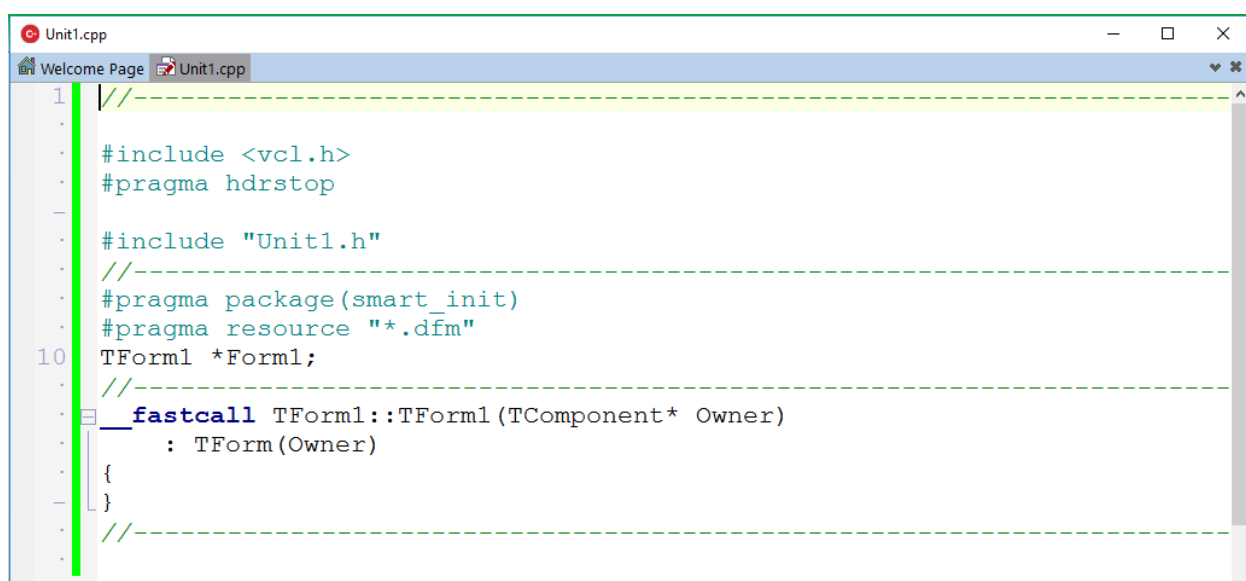
Ilovalarning tez ishlab chiqilishi ob‘ektli mo‘ljallangan dasturlash doirasida xususiyatlar, metodlar va voqealarning qo‘llab-quvvatlanishini bildiradi. *Xususiyatlar* komponentalarning nomlar, matniy aytib berishlar yoki ma‘lumotlar manbalari kabi turli xildagi tavsiflarini osongina o‘rnatish imkonini beradi. *Metodlar* (a‘zo-funksiyalar) komponentadagi ob‘ekt ustida ma‘lum operatsiyalarni amalga oshiradi. Bunday operatsiyalar jumlasida qayta tiklash yoki multimedia qurilmasini qayta o‘rash kabi murakkab operatsiyalarni ham ko‘rsatish mumkin. *Voqealar* komponentalarga foydalanuvchi ko‘rsatayotgan faollashtirish (aktivizatsiya), tugmalarni bosish yoki tahrir qilinadigan kiritish kabi ta’sirlarni ushbu ta’sirlarga sizning munosabat kodlaringiz bilan bog‘laydi. Bundan tashqari voqealar komponentalar holatlarida sodir bo‘ladigan ayrim o‘ziga xos o‘zgarishlar paytida ham yuzaga kelishi mumkin. Bunday o‘ziga xos o‘zgarishlar qatorida ma‘lumotlar bazasiga kirishning interfeysli elementlarida ma‘lumotlarni yangilashni ko‘rsatib o‘tish kifoya. Xususiyatlar, metodlar va voqealar, birgalikda ish olib borar ekan, ular Windows uchun ishonchli ilovalarni intuitiv tarzda dasturlash muhiti - RAD ni hosil qiladi.

=> Tanlangan ob‘ekt bilan assotsiatsiyalanadigan (birgalikda yodga olinadigan) voqealarni ko‘rish uchun, Ob‘ektlar Nozirida Voqealar (Events) qo‘shimcha ilovasini ko‘rsating.

=> O‘zingiz shaklga joylashtirgan tugma komponentasini sichqoncha bilan ikki marta uring

=> Ochilgan Kod Muharriri darchasida kursor ButtonIClick funksiyasi tanasiga instruktsiyalarni kiritish uchun pozitsiyani ko‘rsatadi. Bu funksiya esa tugmachani bosishda yuzaga keladigan OnClick voqeasiga ishlov berish uchun mo‘ljallangan.

4-rasmda oddiy kod ko'rsatilgan bo'lib, u «Natija» tugmasini yana bir bor bosilishiga javoban avval turgan plev aytib berishini ro'yxat oxiriga, hamda navbatdagi next aytib berishini tahrir qilish maydoniga qo'shadi. ListBoxI->Items->Append(prev) yo'riqnomasi, Append metodi yordamida, rrev satrini ListBoxI ro'yxati ob'ektning Items xususiyatiga qo'shadi. EditI->Text=next yo'riqnomasi tahrir qilinayotgan EditI kiritish ob'ektning Text xususiyatiga next satrini taqdim etadi. Aytib berish satrlari ikki o'lchamli count massivida saqlanadi va static turdagi butun o'zgaruvchi tomonidan indekslanadi. Bu o'zgaruvchi esa ButtonI tugmasini bosish bilan yuzaga keladigan voqeaga ishlov berish funksiyasining chaqirilishlari o'rtasida o'zining joriy qiymatini saqlaydi.



```

1  //-----
.
.  #include <vcl.h>
.  #pragma hdrstop
.
.  #include "Unit1.h"
.  //-----
.  #pragma package(smart_init)
.  #pragma resource "*.dfm"
10 TForm1 *Form1;
.  //-----
.  __fastcall TForm1::TForm1(TComponent* Owner)
.      : TForm(Owner)
.  {
.  }
.  //-----

```

4-rasm. Kod muharriri bajarilayotgan modul matnining Unit1.cpp faylida kiritilishi va tahrir qilinishini ta'minlaydi.

Birinchi versiyali ilovani loyihalash bosqichi shuning bilan tugallanadi va ishchi dasturni yaratishga kirishish mumkin bo'ladi.

=>Run | Run bosh menyusi komandasi bilan ilovani kompilyatsiya qilish (ko'chirish) va yig'ish jarayonini ishga tushirib yuboring

=>Dastur chaqirilgach, bir necha marta «Natija» («Rezultat») tugmasini bosing.

### ***Ikki yo'nalishli ishlanma texnologiyasi:***

C++Builder dasturchi va uning kodi o'rtasida hech qanday to'siqlarni qo'ymaydi. Two-Way Tools ikki yo'nalishli ishlanma texnologiyasi vizual loyihalash instrumentlari va Kod Muharriri o'rtasida moslashuvchan, integrallashgan va sinxronlashtirilgan o'zaro aloqa vositasida sizning kodingiz

ustidan nazoratni taminlaydi. Ikki yo‘nalishli ishlanma instrumentlari qanday amal qilishini kuzatib borish uchun, quyidagi operatsiyalarni bajaring:

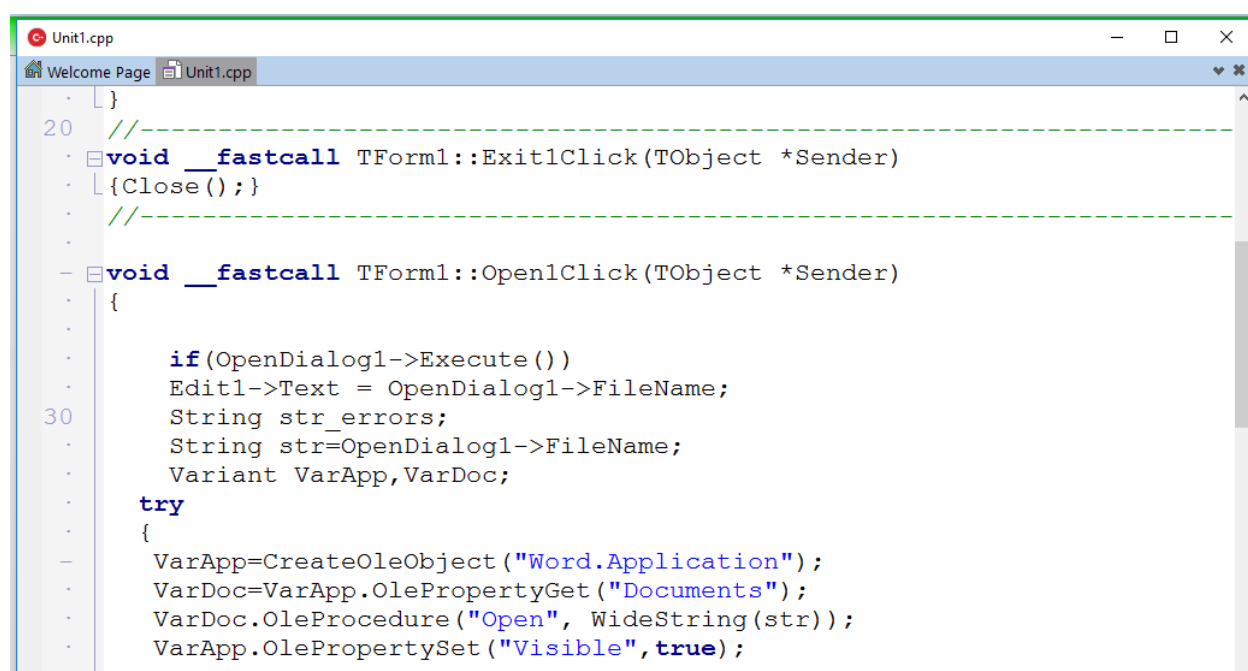
=>Sichqonchaning o‘ng tugmasini bosib, Kod Muharririning kontekstli menyusini oching, keyin Swap Cpp/Hdr Files operatsiyasi yordamida Unit1.h. e‘lonlar fayliga ulaning.

=> Instrumentlarning ekrandagi aksini shunday tashkil qilingki, bunda Kod Muharriri darchasida bir paytning o‘zida loyihalanayotgan shakl va Unit.h fayli ko‘rinsin.

=>OK Button tugmasining yana bitta komponentasini shaklga olib o‘ting. Tugmaning Caption xususiyatida uning ma‘noli nomi «Yangi band» deb ko‘rsating.

Quyidagilarni kuzatib boring: siz tugmani shaklga olib o‘tishingiz bilan, shu ondayoq Unit1.h faylida Button2 ob‘ektining e‘loni paydo bo‘lishi kerak, OnClick voqeaning aniqlanishi esa ushbu voqeaning qayta ishlovchisi bo‘lgan Button2Click metodining e‘lon qilinishini generatsiyalaydi. shaklni loyihalashning va kodni avtomatik generatsiyalash jarayonlarining mana shunday sinxronlashtirilishi C++ilovaning vizual ishlanmasini haqiqatan ham tezlashtiradi va shuning bilan birga dasturning dastlabki matni ustidan nazoratni to‘la saqlab qoladi.

O‘zimizning birinchi ilovamizni ishlashda yana bir qadam qo‘yamiz - uni she‘r bandini avtomatik tarzda generatsiyalashga majbur qilamiz. Buning uchun OnClick voqeasi ishlanmasining funktsiyasini «Yangi band» tugmasini bosib, mazmun bilan to‘ldirishga to‘g‘ri keladi.

A screenshot of a code editor window titled 'Unit1.cpp'. The editor shows C++ code for a button click event handler. The code is as follows:

```
20 //-----  
· void __fastcall TForm1::Exit1Click(TObject *Sender)  
· {Close();}  
· //-----  
·  
- void __fastcall TForm1::Open1Click(TObject *Sender)  
· {  
·  
·     if (OpenDialog1->Execute())  
·     Edit1->Text = OpenDialog1->FileName;  
30 String str_errors;  
· String str=OpenDialog1->FileName;  
· Variant VarApp,VarDoc;  
· try  
· {  
- VarApp=CreateOleObject("Word.Application");  
· VarDoc=VarApp.OlePropertyGet("Documents");  
· VarDoc.OleProcedure("Open", WideString(str));  
· VarApp.OlePropertySet("Visible",true);  
· }  
· }
```



### 5-rasm. *Unit1.cpp* faylida voqeaning yangi qayta ishlatgichi

5-rasmda oddiy kod ko'rsatilgan bo'lib, u «Yangi band» tugmasining navbatdagi bosilishiga javoban yangi bandning ketma-ket yettita satrini chiqarib beradi, bunda birinchi va uchinchi satrlar prev o'zgaruvchisidan olinadi. Bu o'zgaruvchi qiymatini «Natija» tugmasi voqeasining qayta ishlatgichi o'zlashtirib olishi tufayli, bu qiymatni shakl sinfining foydalanuvchilar e'lonlarida public seksiyasida qayta aniqlashga to'g'ri keldi. Bu ish ikkala tugma voqealarining qayta ishlatgichlariga bu qiymatga kirish uchun imkon yaratish maqsadida qilindi.

She'rni butunicha ko'rib chiqish imkonini yaratish maqsadida ro'yxat vertikal aylantirish chizig'iga ega bo'ldi.

C++Builder har bir ilova bilan yashirin nomlari quyidagicha bo'lgan uchta dastlabki faylni eslatishini yodda saqlab qolish kerak:

- UnitI.cpp ilovangizning bajarilayotgan ishga tushirish kodini saqlaydi. Aynan shu erda siz foydalanuvchining komponentalar ob'ektlariga ta'siri paytidagi dastur reaksiyasiga javob beradigan voqealarning qayta ishlatgichlarini yozib qo'yasiz.

- UnitI.h barcha ob'ektlar va ularning konstruktorlarining e'lonlariga ega. Voqealarni qayta ishlash funktsiyalari e'lonlaridagi \_fastcall kalit-so'zga e'tibor bering (C++Builder bu funktsiyalarni avtomatik tarzda generatsiya qiladi). \_fastcall tufayliparametrlar stek orqali emas, balki markaziy protsessor registrarlari orqali uzatiladi. Voqealarni qayta ishlatgichlarning chaqirishlari tez-tez ro'y berib turadi, Shuning uchun stek xotirasidan parametrlarni tanlab olishga sraflanadigan vaqtning tejalishi ancha sezilarli natijalarni beradi. C++Builder komplyatsiya qiladigan va to'playdigan ilovalarning yuqori darajada tez harakatlanishining sabablaridan biri ham shu erda yashiringan.

- ProjectI.cpp ilovada mujassamlangan barcha ob'ektlarga xizmat ko'rsatadi. Har qanday yangi shakl, dasturiy modul yoki ma'lumotlar moduli avtomatik tarzda loyihaviy faylga kiritiladi. Siz bosh menyu komandasi - View | Project Source yordamida yoki Loyiha Administratorining kontekstli menyusidan shu nomdagi opsiyani tanlab olib, Kod Muharriri darchasida loyihaviy fayl dastlabki matnining mazmunini ko'rib chiqishingiz mumkin. Hech qachon loyihaviy faylni qo'lda tahrir qilmang!

Balki siz, birinchi ilova ishlanmasini tugatib, dastlabki fayllarni keyingi seans uchun saqlab qolishni xohlarsiz. Buning uchun quyidagi xatti-harakatlardan birini bajarish kerak:

=>File | Save All komandasi ilovaning hamma dastlabki fayllarini saqlaydi.

File | Save komandasi dasturiy modulning ikkala komandasini saqlaydi, File | Save As komandasi esa ularga yangi nom berishga ruxsat etadi.

File | Save Project As komandasi, fayllarning joriy nomlaridan foydalanib, loyihaviy fayl tarkibiy qismlarining hammasidagi o'zgarishlarni saqlaydi.

### ***Ikki yo'nalishli ishlanma texnologiyasi***

#### ***Loyihaviy shablonlarni qo'llash:***

Ob'ektlar Xazinasidagi tayyor loyihaviy shablonlardan foydalanar ekansiz, siz dasturni ishlab chiqishda ko'pchilik ilovalar uchun tipik bo'lgan operatsiyalarni chetlab o'tish imkoniyatiga ega bo'lasiz. Bu qanday operatsiyalar dersiz. Bular, masalan, menyu va tez chaqirib olish tugmalari panelini tuzish, standart chaqirishlar dialogi va fayllarni tuzishni tashkil etish bilan bog'liq operatsiyalardir. Siz shablonga kiritgan o'zgartirishlar xuddi shu loyihaviy shablondan boshqa ishlab chiquvchilarning foydalanishiga ta'sir qilmaydi.

Ko'p hujjatli interfeys (MDI) rejimida ishlash uchun loyihaviy shablon asosida ilova prototipini yaratish uchun quyidagi xatti-harakatlarni amalga oshiring:

Filtrlar muharriri darchasida TOpenDialog komponentasining Filter xususiyati qiymatlari ustunida matniy hujjatlar fayllarining nomlari va kengayishlarini ko'rsating.

Agar siz shunday ilovani kompilyatsiya qilib, to'play olsangiz, bu holda u faqat MDI rejimida darchalar bilan amallar bajara olishini hamda, darchani tanlab olingan fayllarning matniy mazmuni bilan to'ldirmay turib, fayllarni ochish dialogini chaqirib olishni «bilishi»ni ko'rishingiz mumkin. Ya'ni prototip nofunktsional va amaliy jihatdan befoyda bo'lib qoldi. Ilova qandaydir ongli xulq-atvorga ega bo'lishi uchun, quyidagi xatti-harakatlarni bajaring:

Bosh menyudan **View | Forms** komandasini bering va ro'yxatdan MDIChild nomli shaklni tanlab oling.

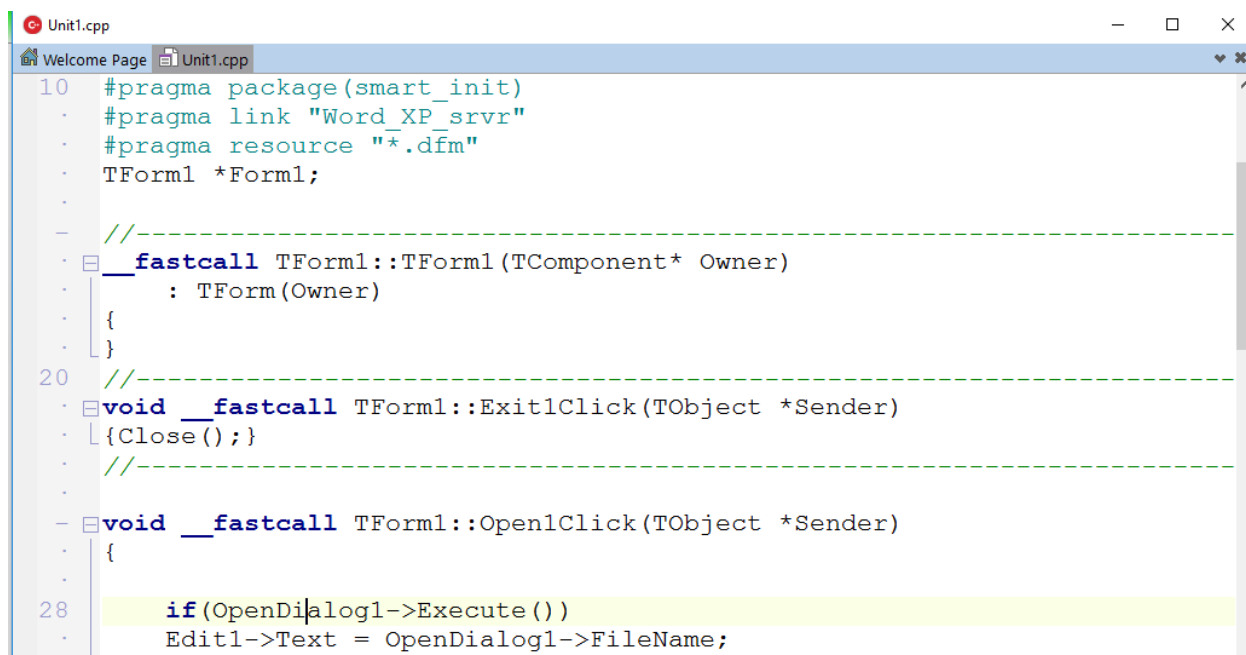
Memo tahrir qilish maydonining ko'p satrli komponentasini Palitraning **Standart** qo'shimcha ilovasidan sho'ba shaklga olib o'ting.

Lines xususiyatli satriy muharrirni tugmani bosish bilan chaqirib olib, TMemo komponentasining tahrir maydonini tozalang. Tahrir maydoni sho'ba darchasining hammasini egallashi uchun, Align xususiyatli alClient qiymatini o'rnatib. Uzun matniy fayllarni ko'rib chiqishni osonlashtirish maqsadida, ScrolBars xususiyatli Ssboth qiymatini o'rnatib.

=>Bosh shaklni sichqoncha yordamida faollashtirib, yana unga qayting hamda ilovalar menyusidan **File|Open** komandasini tanlab oling.

=>Kod Muharriri darchasida kursor menyuning tegishli elementini tanlashda yuzaga keladigan OnClick voqeasining qayta ishlatgichiga yo‘riqnomani kiritish uchun kerakli pozitsiyani ko‘rsatib beradi. C++Builder TOpenDialog bosh shakli (komponentalar Palitrasining Dialogs qistirmasidan) komponentasi uchun ushbu funksiyaning e‘lonini avtomatik tarzda generatsiya qiladi.

6 - rasmda shu voqeaning qayta ishlatgichi bo‘lgan FileOpenItemClick funksiyasi tanasini tashkil qiluvchi zarur yo‘riqnomalar ko‘rsatilgan.



```
10 #pragma package(smart_init)
   . #pragma link "Word_XP_srvr"
   . #pragma resource "*.dfm"
   . TForm1 *Form1;
   .
   . //-----
   . fastcall TForm1::TForm1(TComponent* Owner)
   . : TForm(Owner)
   . {
   . }
20 //-----
   . void __fastcall TForm1::Exit1Click(TObject *Sender)
   . {Close();}
   . //-----
   . void __fastcall TForm1::Open1Click(TObject *Sender)
   . {
28     if(OpenDialog1->Execute())
   .         Edit1->Text = OpenDialog1->FileName;
```

6-rasm. Main.cpp faylida sho‘ba darcha yuklanishining amalga oshirilishi

Ajratib olingan yo‘riqnoma Child sho‘ba darchasi Memo1 ob’ektining Lines satrlarini OpenDialog->FileName nomli ochiq matniy faylning ichidagilari bilan yuklatadi.

Bu faylning ishlanishi hali tugallanganicha yo‘q, albatta. Siz uni kompilyatsiya qilib, to‘plab bo‘lsangiz, bir paytning o‘zida bir necha darchalardagi matniy fayllarni tahrir qila olasiz. Biroq natija beruvchi fayllarning saqlanishi hozircha ko‘zda tutilgan emas - o‘quvchining o‘zi File [Save va File | Save As menyulari komandalari uchun osongina kod yozib oladi.

Ilovani mantiqan eng sodda matniy muharrirga aylantirish uchun bu Edit nomli bosh menyu elementining tushib qoluvchi ro‘yxatiga qidirish va almashtirish komandalarini qo‘shish kerak.

### 1.1. Krill va lotin imlosi yozuvlarini o'zaro o'giruvchi dastur yaratish uchun zarur algoritmlar va funksiyalar.

Krill va lotin imlosi yozuvlarini o'zaro o'giruvchi dastur yaratish uchun dastlab satrlar va fayllar bilan ishlashga imkon beruvchi kutubhonalar e'lon qilinishi hamda ularda doir funksiyalar haqida malumotlarga ega bo'lishimiz zarur. Shu o'rinda fayllar bilan ishlash jarayonida qo'llaniladigan funksiyalarni ham yoritib o'tamiz.

C++ tili ikki xildagi belgilar majmuasini qo'llab quvvatlaydi. Birinchi toifaga, an'anaviy, «tor» belgilar deb nomlanuvchi 8-bitli belgilar majmuasi kiradi, ikkinchisiga 16-bitli «keng» belgilar kiradi. Til kutubxonasida har bir guruh belgilari uchun maxsus funksiyalar to'plami aniqlangan.

C++ tilida satr uchun maxsus tur aniqlanmagan. Satr char turidagi belgilar massivi sifatida qaraladi va bu belgilar ketma-ketligi satr terminatori deb nomlanuvchi 0 kodli belgi bilan tugaydi ('\0'). Odatda, nol-terminator bilan tugaydigan satrlarni ASCIIZ-satrlar deyiladi.

Quyidagi jadvalda C++ tilida belgi sifatida ishlatilishi mumkin bo'lgan o'zgarmaslar to'plami keltirilgan.

1-jadval. C++ tilidagi belgi o'zgarmaslar

Belgilar sinflari	Belgi o'zgarmaslar
Katta harflar	'A' ...'Z', 'A'...'Я'
Kichik qarflar	'a' ...'z', 'a'...'я'
Raqlamlar	'0' ...'9'
Bo'sh joy	gorizontal tabulyatsiya (ASCII kodi 9), satrni o'tkazish (ASCII kodi 10), vertikal tabulyatsiya (ASCII kodi 11), formani o'tkazish (ASCII kodi 12) karetkani qaytarish (ASCII kodi 13)
Punktuasiya belgilari (ajratuvchilar)	! " # \$ % & ' ( ) * + - , . / : ; < = > ? @ [ \ ] ^ _ {   } ~
Boshqaruv belgilari	ASCII kodi 0...1Fh oralig'ida va 7Fh bo'lgan belgilar
Probel	ASCII kodi 32 bo'lgan belgi
O'n oltilik raqlamlar	'0' ... '9', 'A' ... 'F', 'a' ... 'f'

Satr massivi e'lon qilinishida, satr oxiriga terminator qo'yilishi va natijada satrga qo'shimcha bitta bayt bo'lishini inobatga olinishi kerak:

```
char satr[10];
```

Ushbu e'londa satr satri uchun jami 10 bayt ajratiladi - 9 satr hosil qiluvchi belgilar uchun va 1 bayt terminator uchun.

Satr o'zgaruvchilar e'lon qilinishida boshlang'ich qiymatlarni qabul qilishi mumkin. Bu holda kompilyator avtomatik ravishda satr uzunligini hisoblaydi va satr oxiriga terminatorni qo'shib qo'yadi:

```
char Hafta_kuni[]="Juma";
```

Ushbu e'lon quyidagi e'lon bilan ekvivalent:

```
char Hafta_kuni[]={ 'J', 'u', 'm', 'a', '\0' };
```

Satr qiymatini o'qishda oqimli o'qish operatori ">>" o'rniga getline() funksiyasini ishlatgan ma'qul hisoblanadi, chunki oqimli o'qishda probellar inkor qilinadi (garchi ular satr belgisi hisoblansa ham) o'qilayotgan belgilar ketma-ketligi satrdan «oshib» ketganda ham belgilarni kiritish davom etishi mumkin. Natijada satr o'ziga ajratilgan o'lchamdan ortiq belgilarni «qabul» qiladi. Shu sababli, getline() funksiyasi ikkita parametrga ega bo'lib, birinchi parametr o'qish amalga oshirilayotgan satrga ko'rsatkich, ikkinchi parametrda esa o'qilishi kerak bo'lgan belgilar soni ko'rsatiladi. Satrni getline() funksiyasi orqali o'qishga misol ko'raylik:

```
#include <iostream.h>

int main()
{
    char satr[6];
    cout<<"Satrni kiriting:"<<"\n";
    cin.getline(satr[6]);
    cout<<"Siz kiritgan satr:"<<satr;
    return 0;
}
```

Programmada ishlatilgan satr satri 5 ta belgini qabul qilishi mumkin, ortiqchalari tashlab yuboriladi. getline() funksiyasiga murojaatda ikkinchi parametr qiymati o'qilayotgan satr uzunligidan katta bo'lmasligi kerak.

Satr bilan ishlaydigan funksiyalarning aksariyati «string.h» kutubxonasida jamlangan. Nisbatan ko'p ishlatiladigan funksiyalarning tavsifini keltiramiz.

Satrlar bilan ishlashda, aksariyat hollarda satr uzunligini bilish zarur bo'ladi. Buning uchun «string.h» kutubxonasida strlen() funksiyasi aniqlangan bo'lib, uning sintaksisi quyidagicha bo'ladi:

```
size_t strlen(const char* string )
```

Bu funksiya uzunligi hisoblanishi kerak bo'lgan satr boshiga ko'rsatkich bo'lgan yagona parametrga ega va u natija sifatida ishorasiz butun sonni qaytaradi. strlen() funksiyasi satrning real uzunligidan bitta kam qiymat qaytaradi, ya'ni nol-terminator o'rnini hisobga olinmaydi.

Xuddi shu maqsadda sizeof() funksiyasidan ham foydalanish mumkin va u strlen() funksiyasidan farqli ravishda satrning real uzunligini qaytaradi. Quyida keltirilgan misolda satr uzunligini hisoblashning har ikkita varianti keltirilgan:

```
#include <iostream.h>
#include <string.h>
int main()
{
    char Str[]="1234567890";
    cout << "strlen(Str)="<<strlen(Str)<<endl;
    cout<< "sizeof(Str)="<<sizeof(Str)<<endl;
    return 0;
}
Programma ishlashi natijasida ekranga
strlen(Str)=10
sizeof(Str)=11
```

xabarlari chiqadi.

Odatda sizeof() funksiyasidan getline() funksiyasining ikkinchi argumenti sifatida ishlatiladi va satr uzunligini yaqqol ko'rsatmaslik imkonini beradi:

```
cin.getline(Satr, sizeof(Satr));
```

Satrlarni solishtirish ularning mos o'rindagi belgilarning kodlarini o'zaro solishtirish bilan aniqlanadi. Buning uchun «string.h» kutubxonasida standart funksiyalar mavjud.

strcmp() funksiyasi sintaksisi

```
int strcmp(const char* str1, const char* str2)
```

ko'rinishiga ega bo'lib, funksiya str1 va str2 solishtirish natijasi sifatida son qiymatni qaytaradi (masalan, butun i o'zgaruvchisida) va ular quyidagicha izohlanadi:

a)  $i < 0$  - agar str1 satri str2 satridan kichik bo'lsa;

b)  $i = 0$  - agar str1 satri str2 satriga teng bo'lsa;

s)  $i > 0$  - agar str1 satri str2 satridan katta bo'lsa.

Funksiya harflarining registrini farqlaydi. Buni misolda ko'rishimiz mumkin:

```
char satr1[80]="programmalash tillari:C++,pascal.";
```

```
char satr2[80]="programmalash tillari:C++,pascal.";
```

```
int i;
```

```
i=strcmp(satr1,satr2);
```

Natijada i o'zgaruvchisi musbat qiymat qabul qiladi, chunki solinshtirilayotgan satrlardagi «pascal» va «Pascal» satr qismlarida birinchi harflar farq qiladi. Keltirilgan misolda i qiymati 32 bo'ladi. Bu farqlanuvchi harflar kodlarining ayirmasi. Agar funksiya

```
i= strcmp(satr2 ,satr1) ;
```

ko'rinishida murojaat qilinsa i qiymati manfiy son -32 bo'ladi.

Agar satrlardagi bosh yoki kichik harflarni farqlamasdan solishtirish amalini bajarish zarur bo'lsa, buning uchun strcmpi() funksiyasidan foydalanish mumkin. Yuqorida keltirilgan misoldagi satrlar uchun

```
i= strcmpi(satr2 ,satr1) ;
```

amali bajarilganda i qiymati 0 bo'ladi.

strncmp( ) funksiyasi sintaksisi

```
int strncmp(const char* str1, const char* str2,size_t num);
```

ko'rinishida bo'lib, str1 va str2 satrlarni boshlang'ich num sonidagi belgilarini solishtiradi. Funksiya harflar registrini inobatga oladi. Yuqorida misolda aniqlangan str1 va str2 satrlar uchun

```
i=strncmp (satr1,satr2,31);
```

amali bajarilishida i qiymati 0 bo'ladi, chunki satrlar boshidagi 31 belgilar bir xil.

strncmpi() funksiyasi strncmp() funksiyasidek amal qiladi, farkli tomoni shundaki, solishtirishda harflarning registrini hisobga olinmaydi. Xuddi shu satrlar uchun

```
i= strncmpi(satr1 ,satr2,32);
```

amali bajarilishi natijasida i o'zgaruvchi qiymati 0 bo'ladi.

Berilgan satrdagi kichik harflarni bosh harflarga yoki teskarisiga almash-tirishga mos ravishda `_strupr()` va `_strlwr()` funksiyalar yordamida amalga oshi-rish mumkin. Kompilyatorlarning ayrim variantlarida funksiyalar nomidagi tagchiziq ('\_') bo'lmasligi mumkin.

`_strlwr()` funksiyasi sintaksisi

```
char _strlwr(char* str);
```

ko'rinishida bo'lib, argument sifatida berilgan satrdagi bosh harflarni kichik harflarga almashtiradi va hosil bo'lgan satr adresini funksiya natijasi sifatida qaytaradi. Quyidagi programma bo'lagi `_strlwr()` funksiyasidan foydalanishga misol bo'ladi.

```
char str[] = "10 TA KATTA HARFLAR";
```

```
_strupr(str);
```

```
cout<<str;
```

Natijada ekranga

10 ta katta harflar

satri chop etiladi.

`_strupr()` funksiyasi xuddi `_strlwr()` funksiyasidek amal qiladi, lekin satrdagi kichik harflarni bosh harflarga almashtiradi:

```
char str[] = "10 ta katta harflar";
```

```
_strupr(str);
```

```
cout<<str;
```

Natijada ekranga

10 TA KATTA HARFLAR

satri chop etiladi. .

Programmash amaliyotida belgilarni qaysidir oraliqqa tegishli ekanligini bilish zarur bo'ladi. Buni «ctype.h» sarlavha faylida e'lon qilingan funksiyalar yordamida aniqlash mumkin. Quyida ularning bir qismining tavsifi keltirilgan:

`isalnum()` - belgi raqam yoki harf (true) yoki yo'qligini (false) aniqlaydi;

`isalpha()` - belgini harf (true) yoki yo'qligini (false) aniqlaydi;

`isascii()` - belgini kodi 0..127 oraliq'ida (true) yoki yo'qligini (false) aniqlaydi;



isdigit() - belgini raqamlar diapazoniga tegishli (true) yoki yo'qligini (false) aniqlaydi.

Bu funksiyalardan foydalanishga misol keltiramiz.

```
#include <iostream.h>
#include <ctype.h>
#include <string.h>

int main()
{
    char satr[5];
    int xato;
    do
    {
        xato=0;
        cout<<"\nTug'ilgan yilingizni kiriting: ";
        cin.getline(satr,5);
        for (int i=0;i<strlen(satr) && !xato; i++)
        {
            if(isalpha(satr[i]))
            {
                cout<<"Siz harf kiritdingiz !";
                xato=1;
            }
            else
            {
                if(iscntrl(satr[i]))
                {
                    cout<<"Siz boshqaruv belgisini kiritdingiz !";
                    xato=1;
                }
                else
```

```

    if(ispunct(satr[i]))
    {
        cout<< "Siz punktuatsiya belgisini kiritdingiz!";
        xato=1;
    }
    else
    if(!isdigit(satr[i]))
    {
        cout<< "Siz raqamdan farqli belgi kiritdingiz!";
        xato=1;
    }
}
if(!xato)
{
    cout<< "Sizni tug'ilgan yilingiz: "<< satr;
    return 0;
}
}while (1);
}

```

Programmada foydalanuvchiga tug'ilgan yilini kiritish taklif etiladi. Kiritilgan sana satr o'zgaruvchisiga o'qiladi va agar satrning har bir belgisi (satr[i]) harf yoki boshqaruv belgisi yoki punktuatsiya belgisi bo'lsa, shu haqda xabar beriladi va tug'ilgan yilni qayta kiritish taklif etiladi. Programma tug'ilgan yil (to'rtta raqam) to'g'ri kiritilganda "Sizni tug'ilgan yilingiz: XXXX" satrini chop qilish bilan o'z ishini tugatadi.

C++ tilidagi standart va foydalanuvchi tomonidan aniqlangan turlarning muhim xususiyati shundan iboratki, ularning oldindan berilgan chekli elementlardan iboratligidir. Hatto berilganlar dinamik aniqlanganda ham, operativ xotiraning amalda cheklanganligi sababli, bu berilganlar miqdori yuqoridan chegaralangan elementlardan iborat bo'ladi. Ayrim bir tadbiqiy masalalar uchun oldindan berilganning komponentalari sonini aniqlash imkoni yo'q. Ular masalani yechish jarayonida aniqlanadi va yetarlicha katta hajmda bo'lishi mumkin. Ikkinchi tomondan, programmada e'lon qilingan o'zgaruvchilarning qiymatlari sifatida aniqlangan berilganlar faqat programma ishlash paytidagina mavjud bo'ladi va programma o'z ishini tugatgandan keyin yo'qolib ketadi. Agar programma yangidan ishga tushirilsa, bu berilganlarni yangidan hosil qilish zarur bo'ladi. Aksariyat tadbiqiy masalalar esa berilganlarni doimiy ravishda saqlab turishni talab qiladi. Masalan, korxona xodimlarining oylik maoshini hisoblovchi programmada xodimlar ro'yxatini, shtat stavkalari va xodimlar tomonidan olingan maoshlar haqidagi ma'lumotlarni doimiy ravishda saqlab turish zarur. Bu talablarga fayl turidagi obyektlar (o'zgaruvchilar) javob beradi.

*Fayl* - bu bir xil turdagi qiymatlar joylashgan tashqi xotiradagi nomlangan sohadir.

Faylni, boshida ketma-ket ravishda joylashgan yozuvlar (masalan, musiqa) bilan to'ldirilgan va oxiri bo'sh bo'lgan yetarlicha uzun magnit tasmasiga o'xshatish mumkin.

$F$	$F_1$	$F_2$	$F_3$	...
-----	-------	-------	-------	-----

7 -rasm. Fayl tasvirlash

7-rasmda  $F$ - fayl nomi,  $F_1, F_2, F_3$  - fayl elementlari (komponentalari). Xuddi yangi musiqani tasma oxiriga qo'shish mumkin bo'lgandek, yangi yozuvlar fayl oxiriga qo'shilishi mumkin. Yana bir muhim tushunchalardan biri fayl ko'rsatkichi tushunchasidir. Fayl ko'rsatkichi - ayni paytda fayldan o'qilayotgan yoki unga yozilayotgan joy (yozuv o'rnini) ko'rsatib turadi, ya'ni fayl ko'rsatkichi ko'rsatib turgan joydan bitga yozuvni o'qish yoki shu joyga yangi yozuvni joylashtirish mumkin (8-rasm).

$F$	$F_1$	$F_2$	$F_3$	...
-----	-------	-------	-------	-----

8-rasm. Fayl kursatkichi

Keltirilgan rasmda fayl ko'rsatkichi fayl boshini ko'rsatmoqda.

Fayl yozuvlariga murojaat ketma-ket ravishda amalga oishiriladi: p- yozuvga murojaat qilish uchun p-1 yozuvni o'qish zarur bo'ladi. Shuni ta'kidlab o'tish zarurki, fayldan yozuvlarni o'qish jarayoni qisman «avtomatlashgan», unda

i - yozuvni o'qilgandan keyin, ko'rsatkich navbatdagi i+1 yozuv boshiga ko'rsatib turadi va shu tarzda o'qishni davom ettirish mumkin (massivlardagidek indeksni oshirish shart emas). Fayl - bu berilganlarni saqlash joyidir va shu sababli uning yozuvlari ustida to'g'ridan-to'g'ri amal bajarib bo'lmaydi. Fayl yozuvi ustida amal bajarish uchun yozuv qiymati operativ xotiraga mos turdagi o'zgaruvchiga o'qilishi kerak. Keyinchalik, zarur amallar shu o'zgaruvchi ustida bajariladi va kerak bo'lsa natijalar yana faylga yozilishi mumkin.

Operasion sistema nuqtai-nazaridan fayl hisoblangan har qanday fayl C++ tili uchun moddiy fayl hisoblanadi. MS DOS uchun moddiy fayllar <fayl nomi>.<fayl kengaytmasi> ko'rinishidagi «8.3» formatidagi satr (nom) orqali beriladi. Fayl nomlari satr o'zgarmaslar yoki satr o'zgaruvchilarida berilishi mumkin. MS DOS qoidalariga ko'ra fayl nomi to'lik bo'lishi, ya'ni fayl nomining boshida adres qismi bo'lishi mumkin: C:\USER\KURS1\Misol.cpp”, ”A:\matn.txt”.

C++ tilida mantiqiy fayl tushunchasi bo'lib, u fayl turidagi o'zgaruvchini anglatadi. Fayl turidagi o'zgaruvchilarga boshqa turdagi o'zgaruvchilar kabi qiymat berish operatori orqali qiymat berib bo'lmaydi. Boshqacha aytganda fayl turidagi o'zgaruvchilar ustida hech qanday amal aniqlanmagan. Ular ustida bajariladigan barcha amallar funksiyalar vositasida bajariladi.

Fayllar bilan ishlash quyidagi bosqichlarni o'z ichiga oladi:

- fayl o'zgaruvchisi albatta diskdagi fayl bilan bog'lanadi;
- fayl ochiladi;
- fayl ustida yozish yoki o'qish amallari bajariladi;
- fayl yopiladi;
- fayl nomini o'zgartirish yoki faylni diskdan o'chirish amallarini bajarilishi mumkin.

C++ tili C tilidan o'qish-yozish amalini bajaruvchi standart funksiyalar kutubxonasini vorislik bo'yicha olgan. Bu funksiyalar <stdio.h> sarlavha faylida e'lon qilingan. O'qish-yozish amallari fayllar bilan bajariladi. Fayl matn yoki binar (ikkilik) bo'lishi mumkin.

*Matn fayl* - ASCII kodidagi belgilar bilan berilganlar majmuasi. Belgilar ketma-ketligi satrlarga bo'lingan bo'ladi va satrning tugash alomati sifatida CR (karetkani qaytarish yoki '\r') LF (satrni o'tkazish yoki '\n') belgilar juftligi hisoblanadi. Matn fayldan berilganlarni o'qishda bu belgilar juftligi bitta belgiga - CR belgisiga

almashtiriladi va aksincha, yozishda CR belgisi ikkita CR va LF belgilariga almashtiriladi. Fayl oxiri #26 (^z) belgisi bilan belgilanadi.

Matn faylga boshqacha ta'rif berish ham mumkin. Agar faylni matn tahririda ekranga chiqarish va o'qish mumkin bo'lsa, bu matn fayl. Klaviatura ham Kompyuterga faqat matnlarni jo'natadi. Boshqacha aytganda programma tomonidan ekranga chiqariladigan barcha ma'lumotlarni stdout nomidagi matn fayliga chiqarilmoqda deb qarash mumkin. Xuddi shunday klaviaturadan o'qilayotgan har qanday berilganlarni matn faylidan o'qilmoqda deb hisoblanadi.

Matn fayllarining komponentalari satrlar deb nomlanadi. Satrlar uzluksiz joylashib, turli uzunlikda va bo'sh bo'lishi mumkin. Faraz qilaylik, T matn fayli 4 satrdan iborat bo'lsin:

1-satr#13#10	2-satr uzunroq #13#10	#13#10	4-satr#13#10#26
--------------	-----------------------	--------	-----------------

9-rasm. To'rtta satrdan tashkil topgan matn fayli

Matnni ekranga chiqarishda satr oxiridagi #13#10 boshqaruv belgilari juftligi kursorni keyingi qatorga tushiradi va uni satr boshiga olib keladi. Bu matn fayl ekranga chop etilsa, uning ko'rinishi quyidagicha bo'ladi:

1-satr[13][10]

2-satr uzunroq[13][10]

[13][10]

4-satr[13][10]

[26]

Matndagi [n] - n- kodli boshqaruv belgisini bildiradi. Odatda matn tahrirlari bu belgilarni ko'rsatmaydi.

*Binar fayllar* - bu oddiygina baytlar ketma-ketligi. Odatda binar fayllardan berilganlarni foydalanuvchi tomonidan bevosita «ko'rish» zarur bo'lmagan hollarda ishlatiladi. Binar fayllardan o'qish-yozishda baytlar ustida hech qanday konvertatsiya amallari bajarilmaydi.

Oqim tushunchasi berilganlarni faylga o'qish-yozishda ularni belgilar ketma-ketligi yoki oqimi ko'rinishida tasavvur qilishdan kelib chiqqan. Oqim ustida quyidagi amallarni bajarish mumkin:

- oqimdan berilganlar blokini operativ xotiraga o'qish;
- operativ xotiradagi berilganlar blokini oqimga yozish (chiqarish);

- oqimdagi berilganlar blokini yangilash;
- oqimdan yozuvni o'qish;
- oqimga yozuvni chiqarish.

Oqim bilan ishlaydigan barcha funksiyalar buferli, formatlashgan yoki formatlashmagan o'qish-yozishni ta'minlaydi.

Programma ishga tushganda o'qish-yozishning quyidagi standart oqimlar ochiladi:

- stdin – o'qishning standart vositasi;
- stdout - yozishning standart vositasi;
- stderr- xatolik haqida xabar berishning standart vositasi;
- stdprn - qog'ozga chop qilishning standart vositasi;
- stdaux - standart yordamchi qurilma.

Kelishuv bo'yicha stdin - foydalanuvchi klaviaturasi, stdout va stderr - terminal (ekran), stdprn- printer bilan, hamda stdaux - kompyuter yordamchi portlariga bog'langan hisoblanadi. Berilganlarni o'qish-yozishda stderr va stdaux oqimidan boshqa oqimlar buferlanadi, ya'ni belgilar ketma-ketligi operativ xotiraning bufer deb nomlanuvchi sohasida vaqtincha jamlanadi.

Hozirdagi operatsion sistemalarda klaviatura va displeylar matn fayllari sifatida qoraladi. Haqiqatdan ham berilganlarni klaviaturadan programmaga kiritish (o'qish) mumkin, ekranga esa chiqarish (yozish) mumkin. Programma ishga tushganda standart o'qish va yozish oqimlari o'rniga matn fayllarni tayinlash orqali bu oqimlarni qayta aniqlash mumkin. Bu holatni o'qishni (yozishni) qayta adreslash ro'y berdi deyiladi. O'qish uchun qayta adreslashda '<' belgisidan, yozish uchun esa '>' belgisidan foydalaniladi. Misol uchun gauss.exe bajariluvchi programma berilganlarni o'qishni klaviaturadan emas, balki massiv.txt faylidan amalga oshirish zarur bo'lsa, u buyruq satrida quyidagi ko'rinishda yuklanishi zarur bo'ladi:

```
gauss.exe < massiv.txt
```

Agar programma natijasini natija.txt fayliga chiqarish zarur bo'lsa

```
gauss.exe > natija.txt
```

satri yoziladi.

Va nihoyat, agar berilganlarni massiv.txt faylidan o'qish va natijani natija.txt fayliga yozish uchun

```
gauss.exe < massiv.txt > natija.txt
```

buyruq satri teriladi.

Umuman olganda, bir programmaning chiqish oqimini ikkinchi programmaning kirish oqimi bilan bog'lash mumkin. Buni *konveyrli jo'natish* deyiladi. Agar ikkita junat.exe programmasi qabul.exe programmasiga berilganlarni jo'natishi kerak bo'lsa, u holda ular o'rtasiga '|' belgi qo'yib yoziladi:

Fayl oqimi bilan o'qish-yozish amalini bajarish uchun fayl oqimini ochish zarur. Bu ishni, prototipi

```
FILE * fopen(const char * filename, const char *mode);
```

ko'rinishida aniqlangan fopen() funksiyasi orqali amalga oshiriladi. Funksiya filename nomi bilan faylni ochadi, u bilan oqimni bog'laydi va oqimni identifikatsiya qiluvchi ko'rsatkichni javob tariqasida qaytaradi. Faylni ochish muvoffaqiyatsiz bo'lganligini fopen() funksiyasining NULL qiymatli javobi bildiradi.

2-jadval. Fayl ochish rejimlari

mode qiymati	Fayl ochilish holati tavsifi
R	Fayl faqat o'qish uchun ochiladi.
W	Fayl yozish uchun ochiladi. Agar bunday fayl mavjud bo'lsa, u qaytadan yoziladi (yangilanadi)
A	Faylga yozuvni qo'shish rejimi. Agar fayl mavjud bo'lsa, fayl uning oxiriga yozuvni yozish uchun ochiladi, aks holda yangi fayl yaratiladi va yozish rejimida ochiladi.
r+	Mavjud fayl o'zgartirish (o'qish va yozish) uchun ochiladi.
w+	Yangi fayl yaratilib, o'zgartirish (o'qish va yozish) uchun ochiladi. Agar fayl mavjud bo'lsa, undagi oldingi yozuvlar o'chiriladi va u qayta yozishga tayyorlanadi.
a+	Faylga yozuvni qo'shish rejimi. Agar fayl mavjud bo'lsa, uning oxiriga (EOF alomatidan keyin) yozuvni yozish (o'qish) uchun ochiladi, aks holda yangi fayl yaratiladi va yozish rejimida ochiladi.

Matn fayli ochilayotganligini bildirish uchun fayl ochilish rejimi satriga 't' belgisini qo'shib yozish zarur bo'ladi. Masalan, matn fayl o'zgartirish (o'qish va yozish) uchun ochilayotganligini bildirish uchun "rt+" satri yozish kerak bo'ladi.

Xuddi shunday binar fayllar ustida ishlash uchun 'b' belgisini ishlatish kerak. Misol uchun fayl ochilishining "wb+" rejimi binar fayl yangilanishini bildiradi.

Fayl o'zgartirish (o'qish-yozish) uchun ochilganda, berilganlarni oqimdan o'qish, hamda oqimga yozish mumkin. Biroq yozish amalidan keyin darhol o'qib bo'lmaydi, buning uchun o'qish amalidan oldin fseek() yoki rewind() funksiyalari chaqirilishi shart.

Faraz qilaylik "C:\USER\TALABA\iatlkuz.txt" nomli matn faylni o'qish uchun ochish zarur bo'lsin. Bu talab

```
FILE * talabaf=fopen("C:\USER\TALABA\iatlkuz.txt","r+");
```

ifodasini yozish orqali amalga oshiraladi. Natijada diskda mavjud bo'lgan fayl programmada talabaf o'zgaruvchisi nomi bilan aynan bir narsa deb tushuniladi. Boshqacha aytganda, programmada keyinchalik ustida bajarilgan barcha amallar, diskdagi "iatlkuz.txt" fayli ustida ro'y beradi.

Fayl oqimi bilan ishlash tugagandan keyin u albatta yopilishi kerak. Buning uchun fclose() funksiyasidan foydalaniladi. Funksiya prototipi quyidagi ko'rinishga ega:

```
int fclose(FILE * stream);
```

fclose() funksiyasi oqim bilan bog'lik buferlarni tozalaydi (masalan, faylga yozish ko'rsatmasi berilgan, lekin shu paytgacha buferda saqlanayotgan berilganlarni diskdagi faylga yozadi) va faylni yopadi. Agar faylni yopish xatolikka olib kelsa, funksiya EOF qiymatini, normal holatda 0 qiymatini qaytaradi.

fgetc() funksiyasi prototipi

```
int fgetc(FILE *stream);
```

ko'rinishida aniqlangan bo'lib, fayl oqimidan belgini o'qishni amalga oshiradi. Agar o'qish muvaffaqiyatli bo'lsa, funksiya o'qilgan belgini int turidagi ishorasiz butun songa aylantiradi. Agar fayl oxirini o'qishga harakat qilinsa yoki xatolik ro'y bersa, funksiya EOF qiymatini qaytaradi.

Ko'rinib turibdiki, getc() va fgetc() funksiyalari deyarli bir xil ishni bajaradi, farqi shundaki, getc() funksiyasi belgini standart oqimdan o'qiydi. Boshqacha aytganda, getc() funksiyasi, fayl oqimi sifatida standart qurilma bo'lgan holdagi fgetc() funksiyasi bilan aniqlangan makrodir.

fputc() funksiyasi

```
int fputc(int c,FILE *stream);
```

prototipi bilan aniqlangan. fgetc() funksiyasi fayl oqimiga argumentda ko'rsatilgan belgini yozadi (chiqaradi) va u amal qilishida putc() (funksiyasi bilan bir xil.



Fayl oqimidan satr o'qish uchun

```
char * fgetc(char * s, int n, FILE *stream)
```

prototipi bilan fgetc() aniqlangan. fgetc() funksiyasi fayl oqimidan belgilar ketma-ketligini s satriga o'qiydi. Funksiya o'qish jarayonini oqimdan n-1 belgi o'qilgandan keyin yoki keyingi satrga o'tish belgisi '\n' uchraganda to'xtatadi. Oxirgi holda '\n' belgisi ham s satrga qo'shiladi. Belgilarni o'qish tugagandan keyin s satr oxiriga, satr tugash alomati '\0' belgisi qo'shiladi. Agar satrni o'qish muvaffaqiyatli bo'lsa, funksiya s argument ko'rsatadigan satrni qaytaradi, aks holda NULL.

Fayl oqimiga satrni fputs() funksiyasi yordamida chiqarish mumkin. Bu funksiya prototipi

```
int fputs (const char *s, FILE *stream);
```

ko'rinishida aniqlangan. Satr oxiridagi yangi satrga o'tish belgisi va teminatorlar oqimga chiqarilmaydi. Oqimga chiqarish muvaffaqiyatli bo'lsa, funksiya nomanfiy son qaytaradi, aks holda EOF.

feof() funksiyasi aslida makros bo'lib, fayl ustida o'qish-yozish amallari bajarilayotganda fayl oxiri belgisi uchragan yoki yo'qligini bildiradi. Funksiya

```
int feof(FILE *stream);
```

prototipiga ega bo'lib u fayl oxiri belgisi uchrasa, noldan farqli sonni qaytaradi, boshqa holatlarda 0 qiymatini qaytaradi.

## 1.2. Krill va lotin imlosi yozuvlarini o'zaro o'giruvchi dastur yaratish jarayonida qo'llaniladigan C++ Builder komponentalari.

C++Builder 32 razryadli takomillashtirilgan Vizual Komponentalar Kutubxonasi VCL (Visual Component Library) bilan birgalikda etkazib beriladi. Bu kutubxona eng murakkab ilovalarni qurish uchun mo'ljallangan 100 dan ortiq takroran qo'llanadigan komponentalardan iborat. Kutubxonaning asosiy komponentalari Palitralar komponentalarining instrumental Panelida berilgan. Komponentalar belgilari dasturingiz shakliga olib o'tiladi. Quyida krill va lotin imlosi yozuvlarini o'giruvchi dastur yaratish jarayonida qo'llaniladigan komponentalar keltirilgan.

Kutubxona Windows va Windows 95 operatsiya tizimlaridagi Foydalanuvchi Grafik Interfeysi standart interfeys ob'ektlarining to'liq inkapsulalanishini o'z ichiga oladi. Ular orasida, ixtisoslashgan komponentalar bilan bir qatorda, relyatsion ma'lumotlar bazasini boshqarish uchun mo'ljallangan komponentalar alohida o'rin egallaydi. Ishonchli va samarali dasturlarni yaratishda C++Builder ob'ektga mo'ljallangan dasturlash (OMD) imkoniyatlaridan to'liq foydalanadi. C++ Builder bu OMD ekan, OLE (OCX) boshqaruvchi elementlarni kiritish uncha qiyinchilik tug'dirmaydi. O'z masalalaringiz talablarini kerakli darajada qondirish uchun, Kutubxonaning mavjud komponentalaridan foydalaning, hosila komponentalar imkoniyatlarini kengaytiring.

C++Builder bosh xususiyati avvalambor uning dasturni vizual ishlash jarayonida nafaqat tayyor komponentalardan foydalanish, balki yangi komponentalarni yaratish qobiliyatida ham namoyon bo'ladi. Yangi komponentalar, dastlabki komponentalar kabi, sodda bo'lishi mumkin, bunda ularning funksional imkoniyatlari sal-pal kengaytirilgan yoki o'zining mutlaqo o'ziga xos ko'rinishi, xulq-atvori va kodining mazmuni bilan farqlanadigan bo'ladi. Komponentalarning yaratilishi OMD ning vorislik mexanizmiga tayanadi, cheklanishlarga deyarli ega bo'lmaydi hamda quyidagi bosqichlardan o'tadi:

- \* mavjud komponenta turiga vorislik;
- \* yangi xususiyatlar, metodlar va voqealarni aniqlash;
- \* yaratilgan komponentani qayd etish.

Qidirish oson bo'lishi uchun, Palitra funksional jihatdan o'xshash komponentalarni birlashtiradigan qo'shimcha ilovalar bilan bo'lingan. Tanlab olingan komponentaning kontekst menyusini unga sichqonchaning o'ng tugmasini bosib ochish mumkin.

**Standart komponentalar:** Komponentalar palitrasining Standard qo'shimcha ilovalari komponentalari sizning dasturingizga Windows standart interfeysli elementlarning 14 tasining ulanishini amalga oshiradi.

### ***TMainMenu***

Bosh menyu komandolari panelini va ularga mos keladigan tushib qoladigan menyularni yaratadi. Barcha menyu komandalarining identifikatorlari menyuning har qanday konkret komandasiga kirish huquqiga ega bo'lgan Items xususiyati bilan aniqlanadi, AutoMerge xususiyati Merge va Unmerge metodlari bilan birgalikda turli shakldagi menyularning birlashish jarayonini boshqaradi.

### ***TpopupMenu***

Shakl yoki birona boshqa komponenta uchun maxsus menyu yaratadi. E'tiborga oling, aynan shu maqsad uchun har qanday boshqa komponenta PopupMenu xususiyatiga ega bo'lib, bu xususiyatda siz uning bilan bog'liq menyuga iqtibos qilishingiz mumkin.

Agar siz sichqonchani o'ng tugmasini shaklga yoki berilgan komponenta mansub bo'lgan biron boshqa elementga bosish bilan maxsus menyu ekranda paydo bo'lishini xohlasangiz, AutoPopup xususiyatining true qiymatini o'rnatish. Voqea qayta ishlatgichi – OnPopup yordamida bevosita maxsus menyuning paydo bo'lishi oldindan bajariladigan protsedurani aniqlash mumkin.

### ***Tlabel***

Shaklda tahrir qilib bo'lmaydigan statik matnning to'rtburchak sohasini aks ettiradi. Odatda matn boshqa komponenta nomidan iborat bo'ladi.

Nom matni Caption xususiyatining qiymatidir. Alignment xususiyati matnni tekislash usulini aniqlaydi. Shrift o'lchami avtomatik tarzda sohaning maksimal to'ldirilishiga mos kelishi uchun, AutoSize xususiyatining true qiymatini o'rnatish. Kalta soha ichida matnning hammasini ko'rish imkoniga ega bo'lish uchun, WordWrap xususiyatining true qiymatini bering. Transparent xususiyatining true qiymatini o'rnatish, boshqa komponentaning bir qismini to'g'ri uning ustida joylashtirilgan nom orasidan ko'rinib turadigan qilishingiz mumkin.

### ***Tedit***

Axborot yakka satrining tahrir qilinayotgan kiritishidagi to'rtburchak sohani shaklda aks ettiradi. Tahrir sohasining ichidagi boshlang'ich narsalarni Text xususiyatining qiymati bo'lgan satr aniqlaydi.

Tedit komponentasi TcustomEdit sinfining to'g'ridan-to'g'ri hosilasi bo'lib, uning barcha xususiyatlari, metodlari va voqelariga vorislik qiladi.

### ***Tbutton***

Yozuvli to'rtburchak tugmani yaratadi. Tugmachaga bosilganda, dasturda biron-bir xatti-harakat nomlanadi (initsiallashtiriladi).

Tugmachalar ko‘proq dialogli darchalarda qo‘llanadi. Default xususiyatining true qiymati tomonidan tanlab olingan yashirin tugmacha, dialog darchasida har gal Enter klavishchasi bosilganda, OnClick voqea qayta ishlatgichini ishga tushiradi. Cancel xususiyatining true qiymati tanlab olgan uzish tugmachasi, dialog darchasida har gal Escape klavishchasi bosilganda, OnClick voqea qayta ishlatgichini ishga tushiradi.

Tbutton komponentasi TbuttonControl sinfining hosilasi hisoblanadi.

### ***TcheckBox***

Ikkita holatga hamda tavsifiy matnga ega bo‘lgan kvadrat chek-boksni yaratadi (bunda tavsifiy matn chek-boksning vazifasini spetsifikatsiya qiladi).

Boks holatini bildiruvchi «check» biron-bir variantning tanlanishiga mos keladi (boks ustidan tortilgan check bilan belgilanadi), «unchecked» holati esa tanlov olib tashlanishiga mos keladi – bunda Checked komponentasining xususiyati mos ravishda o‘zgaradi hamda OnClick voqeasi yuzaga keladi. Tavsifiy matn Caption xususiyatida saqlanadi. AllowGrayed xususiyatining true qiymatini o‘rnatib, boksni ko‘qroq rangli (masalan, kulrang) qilish mumkin. State xususiyati joriy holatni va boks rangini aks ettiradi.

TcheckBox komponentasi TbuttonControl sinfining hosilasidir.

### ***Tpanel***

Boshqa komponentlarni o‘z ichiga olishi mumkin bo‘lgan bo‘sh panelni yaratadi. Siz Tpanel dan o‘z shaklingizda instrumentlar paneli yoki holatlar satrlarini yaratish uchun foydalanishingiz mumkin.

Tpanel panel komponentasi TcustomPanel sinfining hosilasi bo‘lib, uning barcha xususiyatlar, metodlari va voqealari to‘liq vorislik qiladi.

### **Windows komponentalari**

Windows komponentalari sizning dasturingizga Windows ning 12 ta interfeys elementlarining ulanishini amalga oshiradi.

### ***TOpenDialog* va *TSaveDialog***

Juda ko‘p dasturlarda ***TOpenDialog*** (faylni ochish) va ***TSaveDialog*** (faylni saqlash) komponentalaridan foydalaniladi. Bu komponentalarning xususiyatlari birxil faqat bajarish vazifalari turlicha faylni ochish va faylni saqlash.

Foydalanuvchi tomonidan tanlangan faylni satr tipida qaytarish uchun FileName xususiyati tanlanadi. Shuningdek ushbu xususiyatning qiymati dialogga kirishdan oldin ham o‘rnatilishi mumkin. Shundan so‘ng dialog oynasida Fayl nomi ko‘rsatiladi.

Filter xususiyati orqali qidirish oynasida mavjud bo‘lgan fayllardan faqat tanlangan tipdagi fayllarni ko‘rinishini taminlaydi. FilterIndex xususiyati dialog

oynasi ochilganda foydalanuvchiga ko'rsatiladigan filtr sonini bildiradi. Misol uchun, *FilterIndex* = 1 qiymati standart birinchi filtni o'rnatadi.

*InitialDir* xususiyati foydalanuvchi dialog oynasini ochganda odatda qaysi katalogni ochish kerakligini bildiradi. Agar foydalanuvchi tomondan xususiyatga qiymat berilmagan bo'lsa dialog oynasi ochilganda foydalanuvchi tomonidan oxirgi marta ochilgan katalogga murojat qilinadi.

*DefaultExt* xususiyati standart fayl kengaytmasi qiymatini belgilaydi. Agar foydalanuvchi ushbu xususiyatga qiymat bermasa muloqot oynasida faylning to'liq nomi va tupini ko'rsatishi lozim.

*Title* xususiyati muloqot oynasi sarlavhasini nomlashga imkon beradi. Agar bu xususiyatga qiymat berilmagan bo'lsa muloqot oynasi operatsion sistemaga mos sarlovha bilan ochiladi.

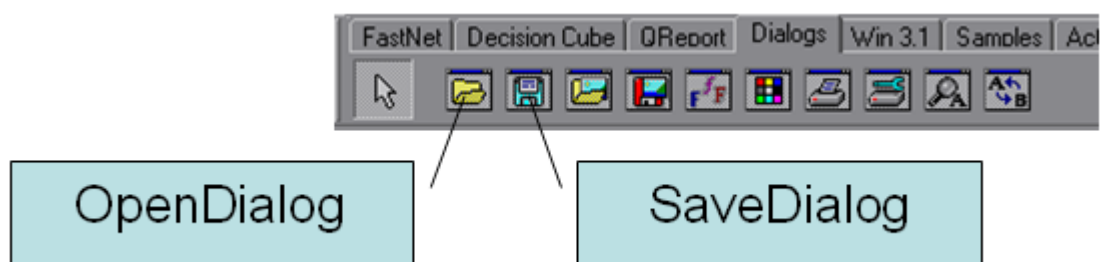
*Options* xususiyati faylni tanlash shartlarini belgilaydi. Ko'p foydalaniladigan xususiyatlarni o'z ichiga oladi, ular quidagilar:

3-jadval

ofAllowMultiSelect	Foydalanuvchiga bir nechta faylni tanlash imkonini beradi.
ofCreatePrompt	Agar foydalanuvchi mavjud bo'lmagan fayl nomini kiritisa, kiritilgan nom bilan yangi fayl hosil qilish haqida so'rov paydo bo'ladi.
ofEnableIncludeNotify	Muloqot oynasiga habar yuborishga ruhsat beradi.
ofEnableSizing	Foydalanuvchiga muloqot oynasi o'lchamlarini o'zgartirishga ruhsat beradi.
ofExtensionDifferent	Foydalanuvchi tomonidan tanlangan fayl kengaytmasi <i>DefaultExt</i> dan farq qilishini ko'rsatuvchi bayroqcha.
ofFileMustExist	Agar foydalanuvchi mavjud bo'lmagan fayl nomini kiritisa xatolik haqida habar beradi.
ofHideReadOnly	Muloqot oynasidan "faqat o'qish uchun" indikatorini o'chiradi.
ofNoChangeDir	OK tugmasi bosilgandan keyin faylni qidirish jarayonida ochilgan kataloglarni tiklaydi.
ofNoDereferenceLinks	Muloqot oynasida "tezkor tugmacha"lardan foydalanishni taqiqlaydi.
ofNoLongNames	Ismning faqat 8 ta belgisi va kengaytmaning uchta belgisi ko'rsatiladi.
ofNoNetworkButton	Muloqot oynasidan qidirish tugmasini olib tashlaydi. <i>ofOldStyleDialog</i> bayroqchasi o'rnatilgan holdagina ishlaydi.
ofNoReadOnlyReturn	Agar foydalanuvchi faylni faqat o'qish uchun tanlagan bo'lsa, xato xabari paydo bo'ladi.
ofNoTestFileCreate	Faylni saqlashda tarmoqdagi himoyalangan fayllarni va mavjud bo'lmagan disklarni tanlashni oldini oladi.

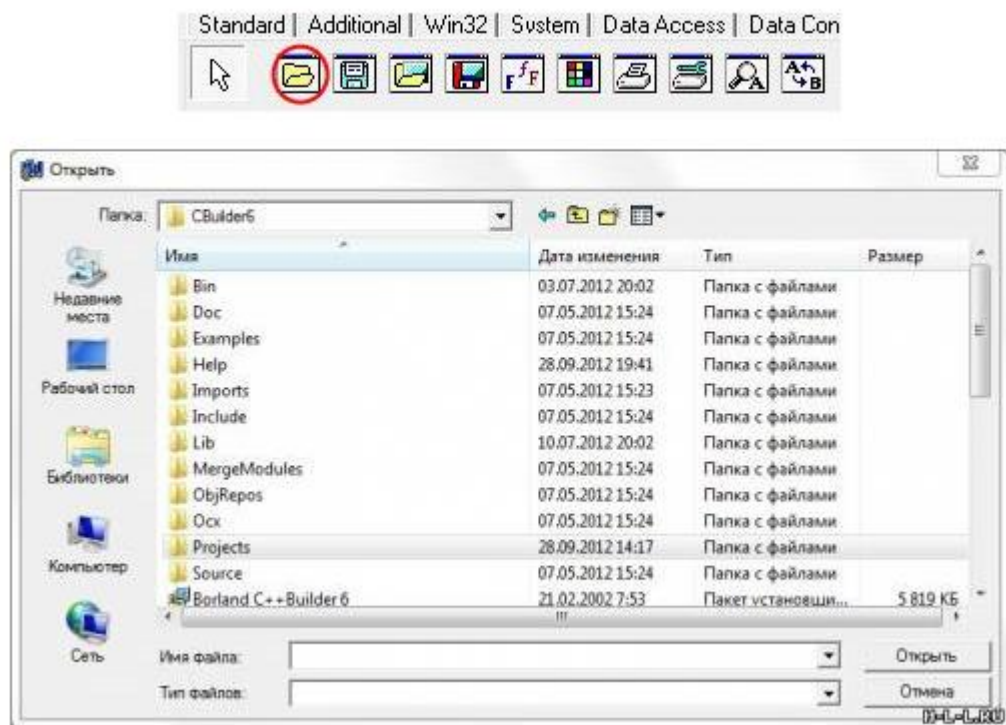
ofNoValidate	Xatlanmaydigan belgilar fayl nomlariga yozilishiga ruxsat bermaydi, ammo siz hal qilinmagan belgilar bilan fayllarni tanlashingizga to'sqinlik qilmaydi
ofOldStyleDialog	Eski uslubda fayl tanlovi dialogini hosil qiladi.
ofOverwritePrompt	Agar faylni saqlashda foydalanuvchi mavjud fayl nomini yozgan bo'lsa, unda o'sha faylga ega bo'lgan fayl bor va foydalanuvchi mavjud faylni qayta yozish istagi so'raladi.
ofPathMustExist	Agar foydalanuvchi fayl nomidagi mavjud bo'lmagan katalogni ko'rsatgan bo'lsa, xato xabari yaratadi.
ofReadOnly	Odatiy bo'lib, dialog oynasi ochilganda Read-Only indikatorini ochiladi
ofShareAware	Kollektiv kirish shartlarining buzilishidagi xatolar inobatga olinmaydi va ularga qaramasdan faylni tanlashga imkon beradi.
ofShowHelp	Muloqot oynasidagi Yordam tugmasini ko'rsatadi.

Windows operatsiya tizimida fayllarni qidirish uchun ularni saqlash va ochishning universal dialogli darchalari ko'zda tutilgan bo'lib, ulardan foyladanishda Dialog qo'shimcha ilovadan tegishli komponentalarni shaklda joylashtirish kerak.



10-rasm.

Bu komponentalarni shaklda joylashtirgach, fayllar bilan ishlashning standart dialogli darchalarini chaqirib olish mumkin.

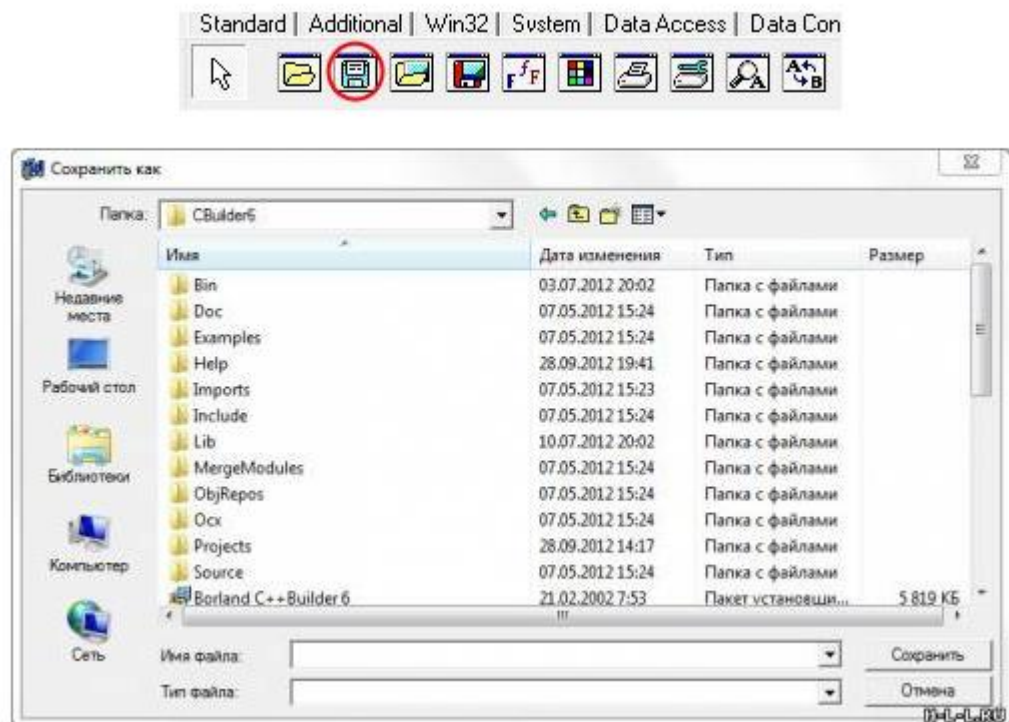


11-rasm.

```

if(OpenDialog1->Execute())
{
    .....
}

```



12-rasm.

```
        if( SaveDialog1->Execute() )  
    {  
        .....  
    }
```



## **II bob. Borland C++ Builder dasturlash muhitida krill va lotin imlosi yozuvlarini o'giruvchi dasturiy vosita yaratish texnologiyasi.**

Bitiruv malakaviy ishda tayyorlangan dastur loihadini tuzushda jekson metodiga asoslanilgan.

**Masalani qo'yilishi:** Programmalarni loyihalashtirish va ishlab chiqishning birinchi bosqichi masalani qo'yilishini ishlab chiqishdir. Kompyuter programmalari uchun masalani qo'yilishini ishlab chiqishda quyidagi ishlar bajarilishi lozim:

1. Muammoni yechish yoki maqsadga erishish uchun zarur bo'lgan talablarni (xususiyatlar, sifat va imkoniyatlar) ishlab chiqish;
2. O'z ichiga quyidagilarni oluvchi spetsifikatsiyalarni ishlab chiqish:
  - programmaning maqsadi;
  - sistema funktsiyalarini tavsiflash;
  - kiritiladigan va chiqariladigan ma'lumotlarning spetsifikatsiyalari;
  - nazorat talablari (test holatlarini belgilash);
  - hujjatlar tipi va soni.

Bu jarayonda, sistema tayyor holda ega bo'lishi lozim bo'lgan xususiyatlar aniqlanadi, sistemaning funktsiyalari va interfeysning xarakteristikalarini tavsiflanadi.

Masalani yechishni boshlashdan avval uni aniq ta'riflash lozim. Birinchi navbatda - bu boshlang'ich va natijaviy ma'lumotlarni aniqlash hamda nima berilgan; nimani aniqlash lozim, degan savollarga javob topishni anglatadi.

Masalani qo'yilishini yanada mufassalroq ko'rib chiqishda quyidagi savollarga javob berish lozim:

### **- yechimni qanday aniqlash kerak?**

Krill va lotin alifbosi orasidagi munosabatlarni to'g'ri amalga oshiruvchi dasturning yaratilishi uchun va yechimning aniqlanishi uchun dastlab shu turga mansub bo'lgan dasturlar va ulardagi kamchiliklar o'rganilishi kerak.

### **- qanday ma'lumotlar yetishmayapti va ularning hammasi lozimmi;**

yaratilgan dasturiy vosita yangicha dasturlarga (mavjud dasturlarning yangi avlodlariga) mutanosibligiga e'tibor berish zarur.

**Programmani loyihalashtirish:** Dastavval, programm sistemaning arxitekturasi loyihalashtiriladi. Bu loyihalashtirishning boshlang'ich (umumiy) bosqichini ko'zda tutadi va sistema strukturasiga qo'yiladigan talablarni sinchiklab o'rganish bilan tugallanadi. Odatda, modullar pog'onasida har bir modulga talablar majmui ishlab chiqiladi:

- modulga nom beriladi, hamda formal parametrli modullar funktsiyalari haqida taklif kiritiladi;
- modul bajaradigan ishlar obzori beriladi;

- qaysi modullar shu modulga murojaat qiladi va mazkur modul qaysi modullarga murojaat qiladi;
- formal va haqiqiy parametrlar, global, lokal bir-biri bilan bog'liq bo'lgan (bir qancha modul uchun umumiy bo'lgan) o'zgaruvchilar;
- moduldagi ko'rsatilishi foydali bo'lgan kommentariylar.

Keyingi bajaradigan ishimiz – sinchiklab loyihalashtirishdir. Bu bosqichda programma protseduralar bo'yicha tavsiflanadi, hamda har bir modulni amalga oshirish uchun algoritmini tanlash va baholash ishlari bajariladi. Loyihalashtirish uchun sistemaga bo'lgan talablar asos qilib olinadi.

Programmarni loyihalashtirishning turli metodlari mavjud. Zamonaviy metodlar dekompozitsiyaga asoslangan. Dekompozitsiya esa o'z navbatida abstraktsiyalarga asoslanadi. Dekompozitsiyaning maqsadi – aniq, oddiy qoidalarga asoslanib o'zaro munosabatda bo'ladigan modullarni yaratishdan iboratdir. Dekompozitsiya, programmani, keyin birlashtirilishi mumkin bo'lgan komponentlarga bo'lish uchun ishlatiladi.

Arxitekturani loyihalash metodlari ikki guruhga bo'linadi:

- ishlov berishga mo'ljallangan;
- ma'lumotlarga mo'ljallangan.

**Ishlov berishga mo'ljallangan metodlar** o'z ichiga quyidagilarni oladi:

- a) modulli programmalash;
- b) funktsional dekompozitsiya;
- v) ma'lumotlar oqimidan foydalanib loyihalash.
- g) loyihani strukturaviy tahlili texnologiyasi

Modulli programmalash. Asosiy kontseptsiyalari:

- har bir modul yagona mustaqil funktsiyani amalga oshiradi;
- yagona kirish chiqish nuqtasiga ega;
- modulning razmeri iloji boricha kichiklashtiriladi;
- har bir modul boshqa modullardan mustaqil holda ishlab chiqiladi;

Mazkur tamoyilga asoslansak, har bir modul alohida testlanadi, so'ngra kodlashtirilib va testlangandan so'ng ular birlashtiriladi va butun sistema testlanadi.

Funktsional dekompozitsiya. Har bir modul loyihachining sub'ektiv qarori bilan xarakterlanadi. Aloqa, yaxshi tashkil etilgan interfeyslar yordamida amalga oshiriladi.

Ma'lumotlar oqimidan foydalanib loyihalash. Ma'lumotlar oqimidan programmani loyihalashning asosi kabi foydalaniladi.

Yuqoridan pastga qadamma-qadam detallashtirish strukturali loyihalash elementlaridan iborat:

- ma'lumotlar oqimini ekspertiza qilish va ma'lumotlar oqimi grafigida aks ettirish;
- ma'lumotlar oqimining kiritiladigan markaziy va chiqariladigan elementini tahlil qilish;
- programmaning iyerarxik strukturasini tuzish;
- programma strukturasini detallashtirish va optimallashtirish.

Loyihani strukturaviy tahlili texnologiyasi. Bu, sistemaning ob'ektlari o'rtasidagi iyerarxik funktsional aloqalarni tuzishda maxsus grafik vositadan foydalanib, strukturaviy tahlil qilishga asoslangan. Ushbu usul sistemani yaratishning boshlang'ich bosqichlarida samarali hisoblanadi, chunki diagrammalar ham oddiy, ham tushunarli bo'ladi.

**Ma'lumotlar strukturasidan foydalanib loyihalash metodlari** quyidagilardir:

- a) Jekson metodologiyasi;
- b) Uorner metodologiyasi;
- v) iyerarxik diagrammalar metodi;
- g) loyihalashning ob'ektga mo'ljallangan metodologiyasi.

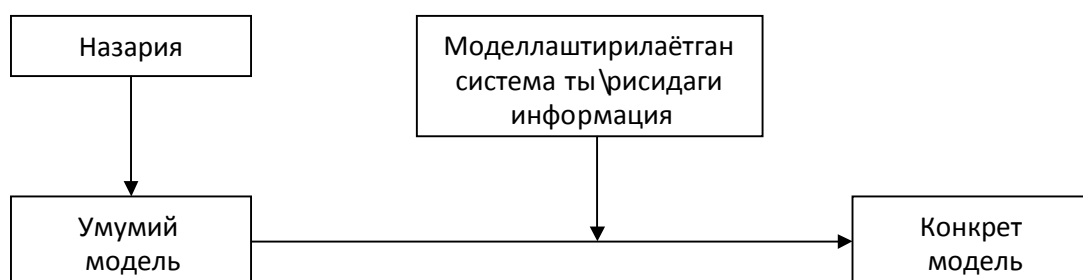
*Jekson metodologiyasi.* Bunda ma'lumotning strukturasini loyihani tuzishda tayanch element hisoblanadi. Programmaning strukturasini ishlov berilishi lozim bo'lgan ma'lumotlar strukturasiga qarab aniqlanadi. Bunda programma kiritiladigan ma'lumotlarni chiqariladigan ma'lumotlarga aylantiruvchi mexanizm sifatida qaraladi. Bu metod quyidagilarni ko'zda tutadi:

- kiritiladigan va chiqariladigan ma'lumotlar strukturasini ishlab chiqish va aks ettirish;
- ushbu strukturaviy elementlarning tasvirini bir-biri bilan bog'lash yo'li bilan programma strukturasini aks ettirish;
- ma'lumotlar strukturalari ustida bajariladigan diskret operatsiyalarni aniqlash;
- ma'lumotlar strukturalariga ishlov berish algoritmini tuzish.

**Modelni tuzish** Ko'pchilik hollarda modelni tuzish, murakkab masala hisoblanadi. Modellashtrishda tajriba orttirish uchun iloji boricha ko'p, ma'lum (mashhur) va muvaffaqiyatli chiqqan modellarni o'rganish lozim. Modellarni tuzishda, odatda, ikkita tamoyildan foydalaniladi:

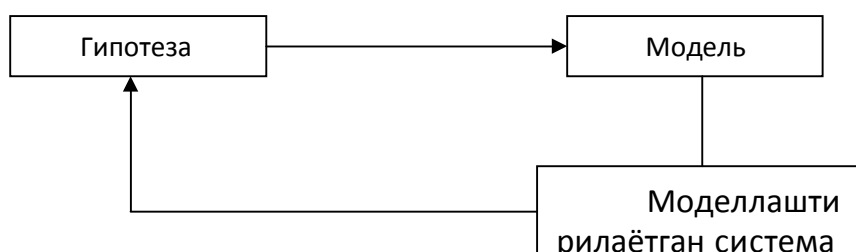
- deduktiv (umumiydan xususiyga);
- induktiv (xususiydan umumiyga).

Deduktiv usulda ma'lum fundamental modelning xususiyligi ko'rib chiqiladi (13-rasm). Bu yerda, berilgan farazlar asosida ma'lum model modellashtirilayotgan ob'ekt shart-sharoitlariga moslashtiriladi.



13-rasm. Deduktiv usulda modelni tuzish sxemasi.

Induktiv usulda (14-rasm) gipotezani taklif qilish, murakkab ob'ektni dekompozitsiya, tahlil, so'ngra sintez qilishdan iboratdir. Bunda sistema xatti-harakatlaridagi qonuniyatlarni aniqlash maqsadida modellashtirish va xulosa chiqarish usullaridan keng foydalaniladi. Gipotezani oydinlashtirib olish



14-rasm. Induktiv usulda modelni tuzish sxemasi.

Induktiv usulda modelni tuzish texnologiyasi:

- 1) empirik, ya'ni tajribaga asoslangan bosqich:
  - a) xulosa chiqarish;
  - b) intuitsiya;
  - c) faraz qilish;
  - d) gipoteza.
- 2) modellashtirish uchun masalaning qo'yilishi;
- 3) baholashlar (miqdoriy va sifat tavsifi);
- 4) modelni tuzish.

**Algoritmni ishlab chiqish.** Algoritmni ishlab chiqish – eng murakkab va ko'p mehnat talab etuvchi, lekin ijodiy nuqtai nazardan qaraganda eng qiziqarli jarayondir.

Ishlab chiqish metodini tanlash, masalaning qo'yilishi va uning modeliga bog'liq. Ushbu bosqichda algoritmning to'g'riligini tekshirish lozim. Bu esa oson emas, ko'p mehnat talab etadi. Algoritm to'g'riligini isbotlashning keng tarqalgan protsedurasi – bu turli testlar majmuidan uni o'tkazishdir. Ammo bular ham programmaning to'g'ri ishlashiga kafolat bera olmaydi. Algoritm to'g'riligini isbotlashning umumiy metodikasiga asosan algoritm, qadamlar ketma-ketligi ko'rinishida tavsiflanadi. Har bir qadam uchun hamma kiritilishi (ushbu qadamgacha bo'lgan shartlar asosida) va chiqarilishi (ushbu qadamdan keyingi shartlar hisobga olingan holda) lozim bo'lgan ma'lumotlar asosida uning to'g'riligini isbotlovchi bironta isbot taklif etiladi. So'ngra yakuniy kiritiladigan va chiqariladigan ma'lumotlar bilan algoritmning tugallanganligi isboti taklif etiladi.

**Algoritmni va uning murakkabligini tahlil qilish.** Bu bosqich foydalaniladigan kompyuter resurslarini aniq ma'lumotlarga ishlov berish vaqtini baholash, hamda mahalliy tarmoq va telekommunikatsiyalarda ishlashga moslashish uchun zarurdir. Bundan tashqari ushbu masala uchun bironta miqdoriy kriteriy bo'lsa maqsadga muvofiq bo'ladi. Chunki bir necha algoritmlar solishtirilib ulardan eng oddiy va samaralisini tanlash mumkin bo'ladi. Programmadan foydalanishdan avval, u, sozlash va testlash bosqichidan o'tishi lozim.

**Programmistlarning eng muhim qoidasi** – Har bir tayyor programm mahsulot uchun yordam (help) va matnli fayl (readme.txt) ko'rinishida hujjatlashtirilgan kuzatuv zarur.

## 2.1. Krill va lotin imlosi yozuvlarini o'zaro o'girish jarayoni qonuniyatlari.

Krill va lotin imlosi yozuvlarini o'zaro o'girish jarayoni qonuniyatlari o'rganish ular orasidagi farqlarni ko'rsatishdan boshlanadi. O'zbek lotin alifbosi 29 ta harf va ' (tutuq) belgisidan iborat, Krill alifbosi esa 33 ta harfdan tashlik topgan. Ko'tinib turibdiki alifbolar hariflari soni o'rtasidagi tafovud yaratilishi kerak bo'lgan dasturning murakkablik jarayonini oshiradi.

### ***Krilldan lotinga:***

1. Krill va lotin alifbosi yozilishida farq qiladigan so'zlar bor, misol uchun сентябрь-sentabr, бюджет-budjet kabi so'zlar harflar to'g'ridan to'g'ri bir alifbodan boshqasiga o'tqazilganda xato so'zlarni tashlik qiladi.
2. Undan tashqari so'z o'rtasida unlidan keyin kelgan "e" xarfi ham muommolidir. Bilamizki imlo qoidasi bo'yicha unlidan keyin kelgan "e" harfi "ye" ko'rinishida yoziladi. Buni asosan familiyalarda ko'rishimiz mumkin. Misol uchun, Атоев-Atoyev, Боқиев-Boqiyev. Huddi shunday so'zlarni Atoev, Boqiev deb o'girilsa xato hisoblanadi.
3. "Ц" harfi bian bog'liq muommo ham bor. "Ц" harfi kichik harf bilan so'z boshida kelganda "ts" deya hato o'irilishi mumkin. Masalan, цирк – tsirk. Aslida "sirk" bo'lishi kerak. Kata harflar bilan yozilganda esa muammo yo'q.
4. Shuning dek imlo qoidasi bo'yicha "ц" harfi so'z o'rtasida undosh harfdan so'ng kelganda "ts" emas, balki "s" bo'lishi kerak. Masalan, акция – aksiya. Shu holatda dasturga maxsus bandlar kirtilmasa, noto'g'ri o'giradi yani "aktsiya" deb. Unli harflardan keyin kelgan "ц" harflarini o'girishda muammo yo'q.

### ***Lotindan krillga:***

1. kata harflar bilan yozilganda "YU", "YO", "YA", "SH", "CH" kabi birikmalarni "ЙУ", "ЙО", "ЙА", "СХ" deb o'girilishi xato bo'ladi. Masalan, "AQSH"ни "АҚСХ" deb o'g'irilishi. Kichik harflar bilan yozilganda bunday muommolar yo'q..

2. krill va lotinda turli xil yoziladigan so'zlara xatoliklar yuz berishi mumkin ularga yuqorida ko'rsatilgan so'zlarnikeltirishimiz mumkin sentabr, budjet.

3. satr boshida kelgan "Э" harfini "E" deb o'giradi. Masalan, "Erkin–Эркин" so'zini "Еркин" deb o'girilishi. Ammo bu faqat satr boshida gina kelganda. Matn o'rtasida kelganda muammo yo'q.

4. "Ц" harfi bilan bog'liq muammo ham bor. "Ц" harfi so'z boshida kelganda, "C" deya noto'g'ri o'giriladi. Masalan, Сирк– Сирк. Aslida "Цирк" bo'lishi kerak. Bu xatolar kata harfbilan yozilganda ham kichik harf bilan yozlganda ham takrorlanadi.

Shunigek, so'z o'rtasida unlidan keyin kelgan barcha "ц" harfini "tc" qilib o'giradi. Masalan, militsiya – милитсия, sivilizatsiya – сивилизатсия.

Undoshdan keyin kelgan barcha "ц" harfini "c" qilib o'giradi. Masalan, aksiya – аксия.

Birinchi noqulaylik o‘ va g‘ harflari bilan bog‘liq. Bu harflarni klaviaturada alohida tugmachalarga bog‘lab qo‘yishning iloji yo‘q. Qolaversa, ‘ belgisi tutuq belgisi ’ bilan shaklan o‘xshash. Ularni ifodalash uchun biz klaviaturadagi bir necha (birvarakayiga 5 ta!) tugmalar kombinatsiyasidan foydalanamiz yoki matnni terib bo‘lgach, o‘ va g‘ harflarini «avtozamena» funksiyasi orqali umumiy almashtiramiz. Bu — masalaning texnik jihat. Amalda esa, masalan, «to‘g‘ri» so‘zidagi ikki qo‘shimcha belgidan birining tushib qolishi ko‘p uchraydi (yo «to‘gri», yo «tog‘ri»), hatto savodli yozilgan matnlarda ham. Bu harflar matn terishda noqulaylik tug‘diradi.

Ikkinchi noqulaylik sh va ch harflari bilan bog‘liq: bu harflar ingliz yoki boshqa tillar uchun qulaydir (chunki ularda aksar so‘zlar qisqa, sh, ch harflarining bir so‘zda ikki marta ishlatilishi yoki o‘zakka shu harflar ishtirokida qo‘shimcha qo‘shilishi uchramaydi), ammo o‘zbek tilida bu harflar noqulaylik keltirib chiqarmoqda. Shu undoshlar bir necha marta ishtirok etgan qashshoq, cho‘chish, shishish, pashsha, ishchi, uchrashish kabi so‘zlarni boshqa alifboga o‘girishda ham noqulayliklar kuzatiladi.

Uchinchi noqulaylik — «ts» tovushining hozirgi lotin alifbosida alohida harf bilan berilmagani. Biz «ts»ni o‘rniga qarab «ts» yoki «s» bilan ifodalaymiz. To‘g‘ri, «ts» turkiy tillar uchun yot tovush. Ammo til taraqqiyoti natijasida bu tovush ishtirokidagi internatsional so‘zlar, terminlardan keng foydalanayapmiz. Ko‘pchilik «ts»ni hamma o‘rinda «s» deb talaffuz etish va yozishni taklif qilayapti. Lekin shunday atamalar borki, «s» bilan yozilganda, omonim bo‘lib qoladi: – sanga, – seriy, – sokol va hokazo. Bu aynan shunday matnlarni ko‘p yozadigan mutaxassislar, abituriyentlar va talabalar tomonidan ma`qullanmayapti.

To‘rtinchi noqulaylik «ye», «yo», «yu», «ya» kabi «yo»lashgan unlilar imlosi bilan bog‘liq. Shu paytgacha, masalan, qachon «yu» va qachon «u» yozilishi bu harflarning kelgan o‘rni va bajaradigan vazifasiga bog‘liq deb kelingan (Rahmatullayev Sh. O‘zbek tilining yangi alifbosi va imlosi. – Toshkent: Universitet, 1999). Lekin bu qarash ushbu harf birikmalarining imlosini murakkablashtirib yubordi. Natijada hatto imlo lug‘atlarida ham variativlik ko‘payib ketdi: sentabr – sentyabr, budjet – byujet, rejissor – rejissyor. «Ye» grafemasi «men», «teng» kabi so‘zlarda, ya‘ni bo‘g‘in o‘rtasida kelsa, bitta «E» tovushini ifodalaydi, «yelka», «yengil» «poyezd» kabi so‘zlarda esa «YE» tovushlarini ifodalaydi. Lotin yozuvida “мўътабар”, “мўъжиза”, “мўъжаз”, “мўътадил” kabi so‘zlar yozilganda “ў” harfidan keyingi ayirish belgisi tushub qoladi. Yani, “mo‘tabar”, “mo‘jiza” kabi. Manashunday so‘zlar bilan bog‘liq muammolar ham bartaraf etildi.

## 2.2. Krill va lotin imlosi yozuvlarini o'zaro o'giruvchi dastur kod qismi.

Krill va lotin alifbosida yozilgan elektron hujjatlarni (\*.doc, \*.docx, \*.txt) krilldan lotinga va aksincha lotindan krillga o'tqazuvchi dastur ko'dini keltiramiz.

```
#include <vcl.h>
```

```
#pragma hdrstop
```

```
#include "Unit1.h"
```

```
#include <string.h>
```

```
#include <string>
```

```
//-----
```

```
#pragma package(smart_init)
```

```
#pragma link "Word_XP_srvr"
```

```
#pragma resource "*.dfm"
```

```
TForm1 *Form1;
```

```
//-----
```

```
__fastcall TForm1::TForm1(TComponent* Owner)
```

```
    : TForm(Owner)
```

```
{
```

```
}
```

```
//-----
```

```
void __fastcall TForm1::Exit1Click(TObject *Sender)
```

```
{Close();}
```

```
//-----
```

```
void __fastcall TForm1::Open1Click(TObject *Sender)
```

```
{
```



```

        if(OpenDialog1->Execute())
        Edit1->Text = OpenDialog1->FileName;

        String str_errors;

        String str=OpenDialog1->FileName;

        Variant VarApp,VarDoc;

    try
    {
        VarApp=CreateOleObject("Word.Application");
        VarDoc=VarApp.OlePropertyGet("Documents");
        VarDoc.OleProcedure("Open", WideString(str));
        VarApp.OlePropertySet("Visible",true);

    }

    catch(Exception &ex)
    {
        str_errors=ex.Message;
        ShowMessage(str_errors);

    }

}

}

//-----

void __fastcall TForm1::BitBtn2Click(TObject *Sender)
{
    WideString macr;

    if(ComboBox1->ItemIndex==0)        macr="Cyr3Lat";

```

*else macr="Lat3Cyr";*

*WordApplication1->Run(macrc\_bstr());*

*}*

*//-----*

*void \_\_fastcall TForm1::D1Click(TObject \*Sender)*

*{*

*ShowMessage("Amaliy matematika va informatika talim yonalishi talabasi Aliyeva  
Dilnoza!");*

*}*

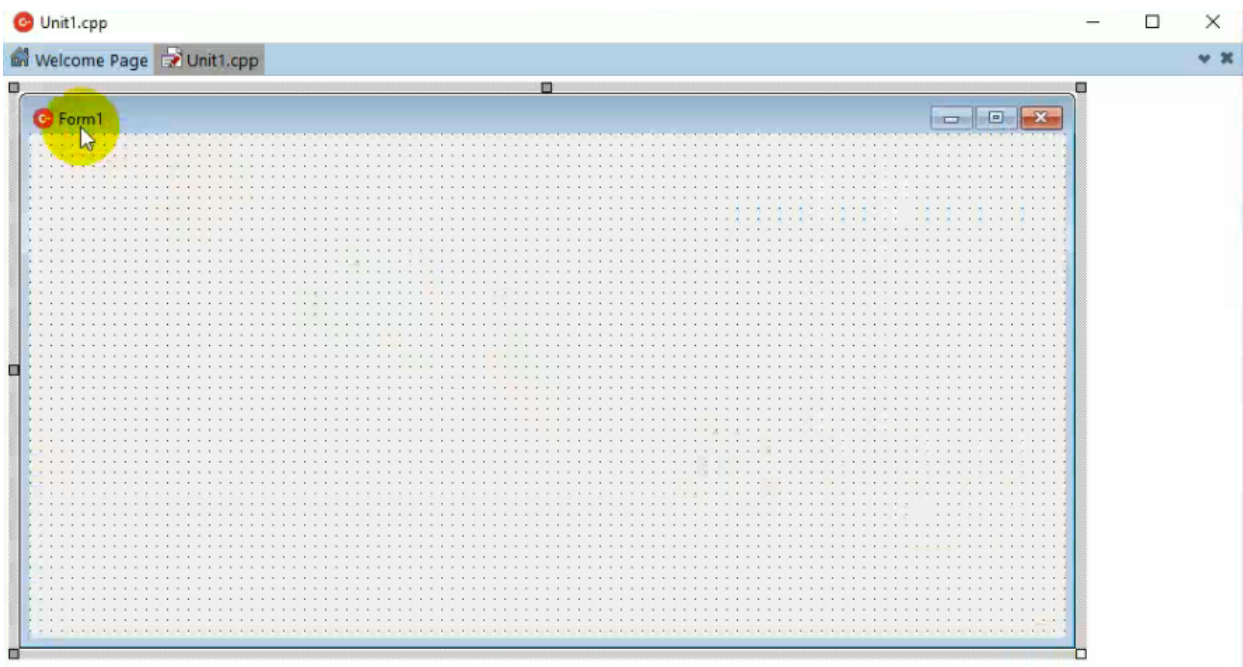
### 2.3. Dasturiy taminot interfeysini ishlab chiqish va amaliyotda qo'llash sohalari.

C++ builder dastur muhitida, Krill va lotin alifbolari orasidagi o'girish jarayonini amalga oshiruvchi dastur yaratish jarayoni hamda dasturiy taminotning interfeysini ishlab chiqish jarayonini rasmlar va izohlar bilan keltiramiz.

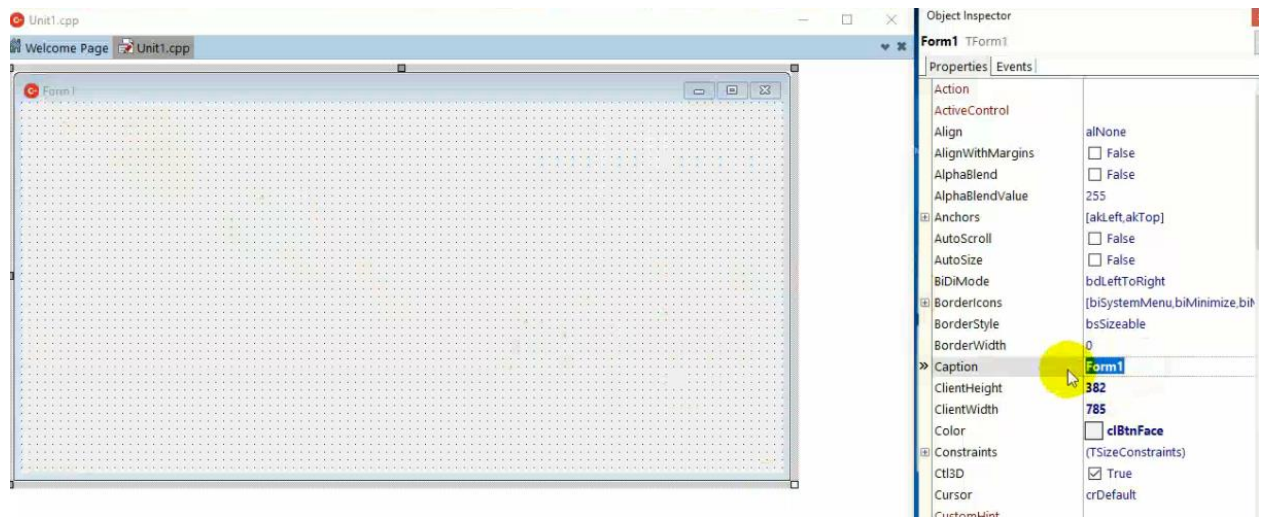
Dastlab amaliy dastur yaratish uchun forma hosil qilamiz:



14-rasm



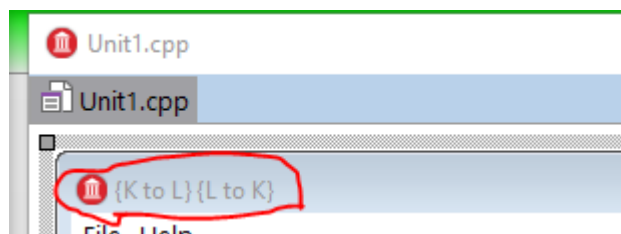
15-rasm



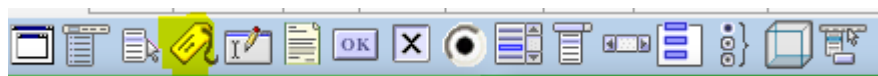
16-rasm. Formaning chap yuqori qismida aks etadigan dastur nommini o'zgartiramiz. Odatiy holatda u Form1 deb nomlanadi.

BorderIcons	[biSystemMenu,biMinimize,bi
BorderStyle	bsSizeable
BorderWidth	0
Caption	{K to L} {L to K}
ClientHeight	242
ClientWidth	659
Color	clBtnFace

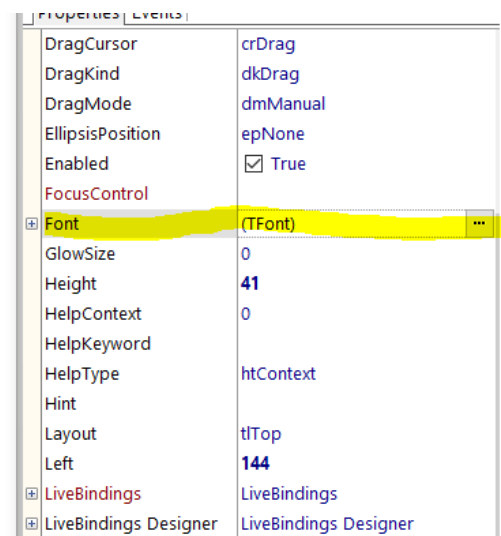
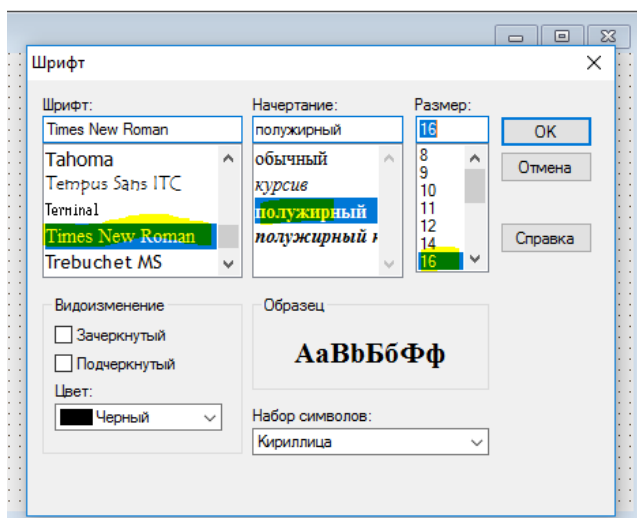
17-rasm



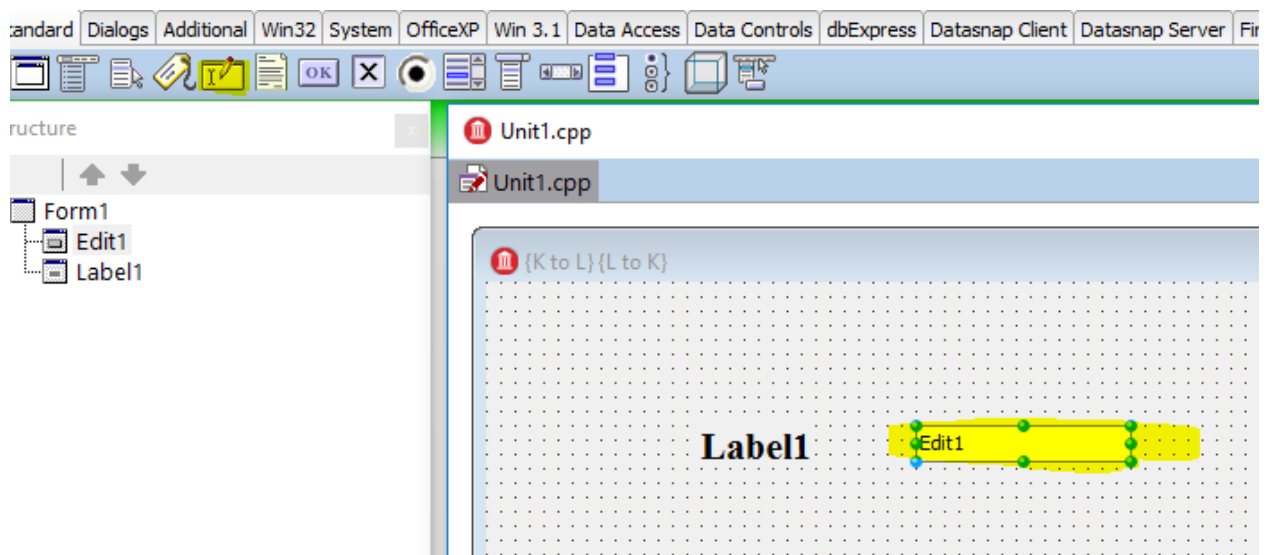
18-rasm



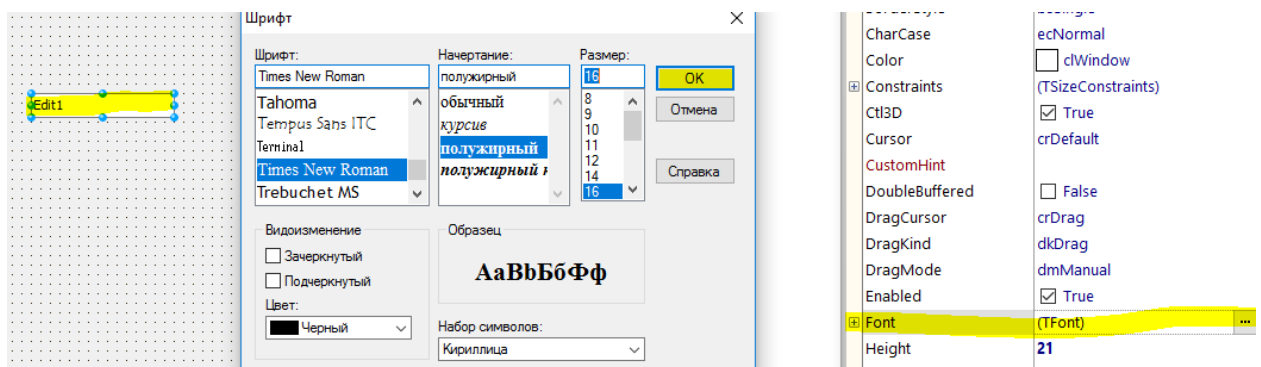
19-rasm. Label komponentasini tanlab formaga joylashtiramiz.



20-rasm. Label komponentasi o'lchovlari va yuzuv shiriftini o'zgartiramiz.



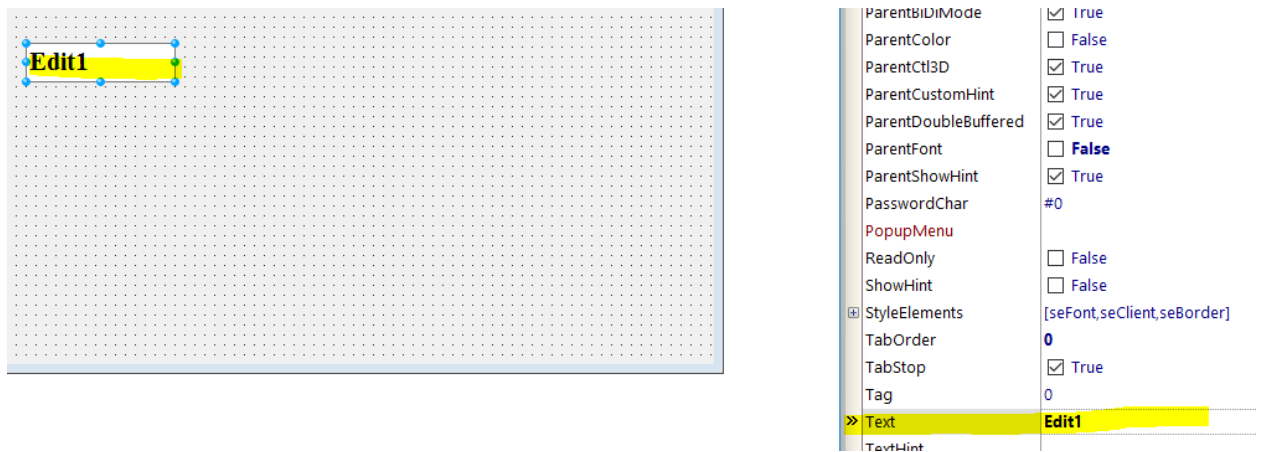
21-rasm. Edit komponentasini tanlab formaga joylaymiz.



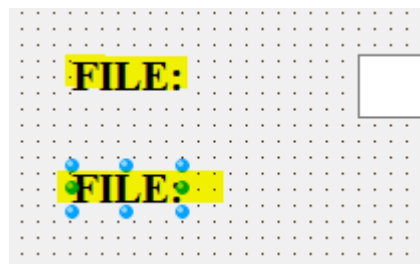
22-rasm. Edit komponentasining o'lchalarini o'zgartiramiz.



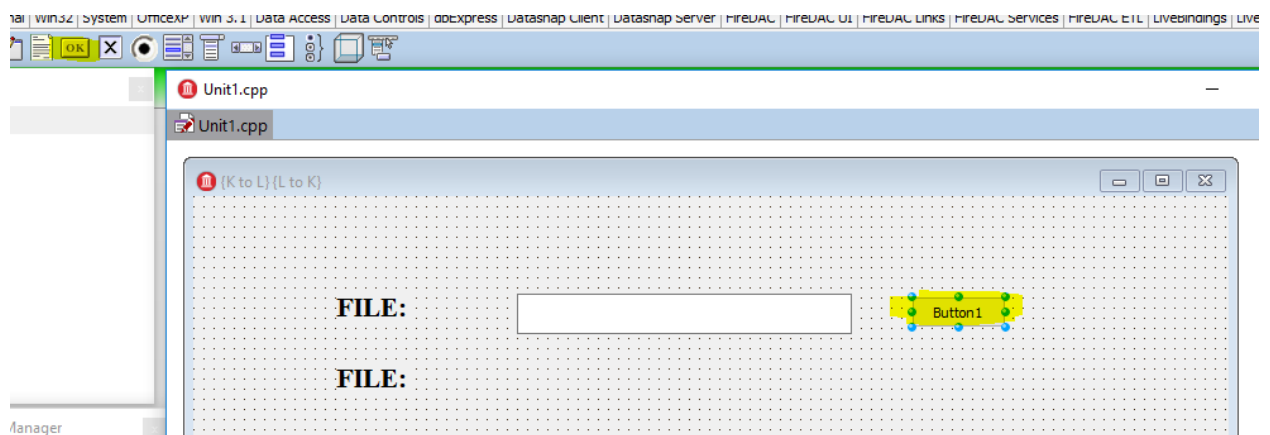
23-rasm. Label komponentasi Caption xususiyatini o'zimizga mos holatga keltiramiz.



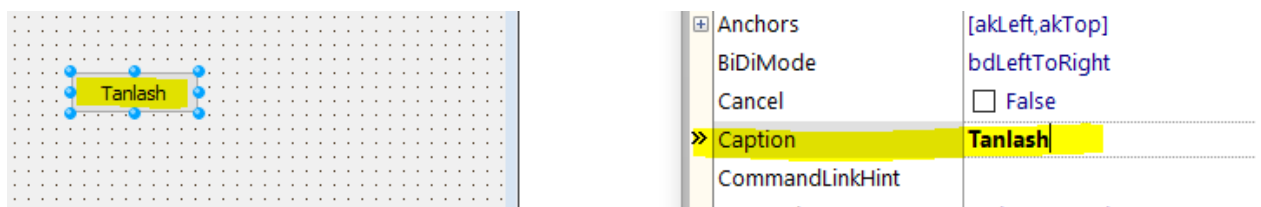
24-rasm. Edit komponentasida Avtomatik saqlangan textni o'zgartiramiz.



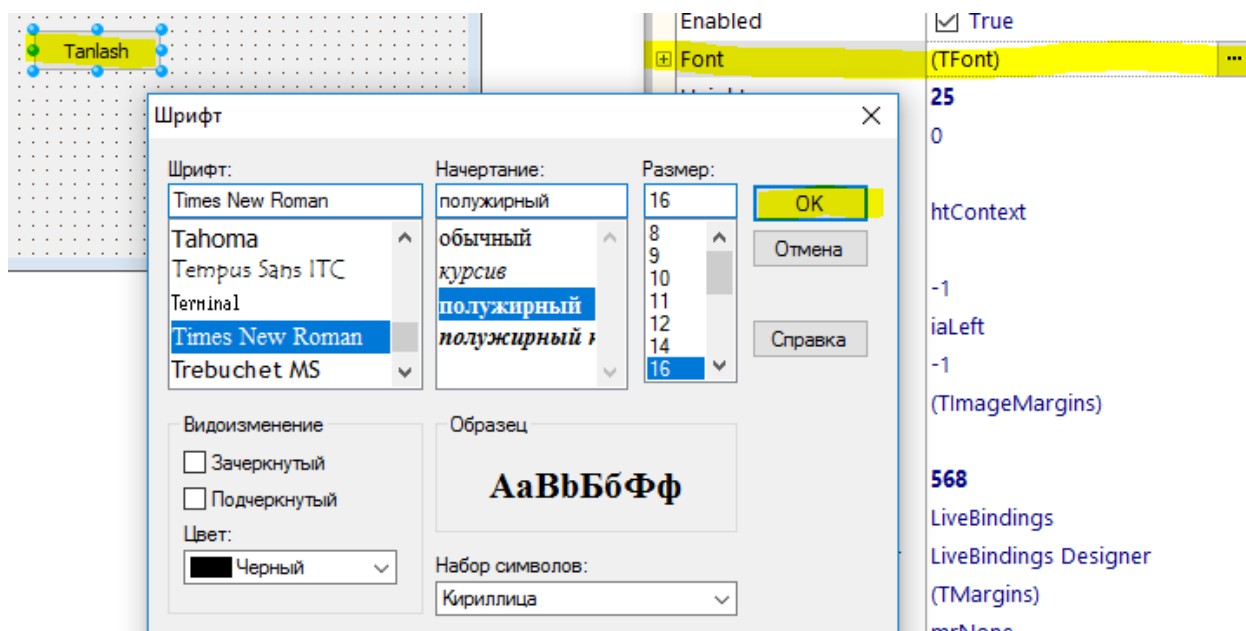
25-rasm. Ctrl+C orqali File nomli komponentadan nusxa olib Ctrl+V oraqli o'lchamlari joriy dasturga to'g'irlangan Edit komponentasiga ega bo'lamiz.



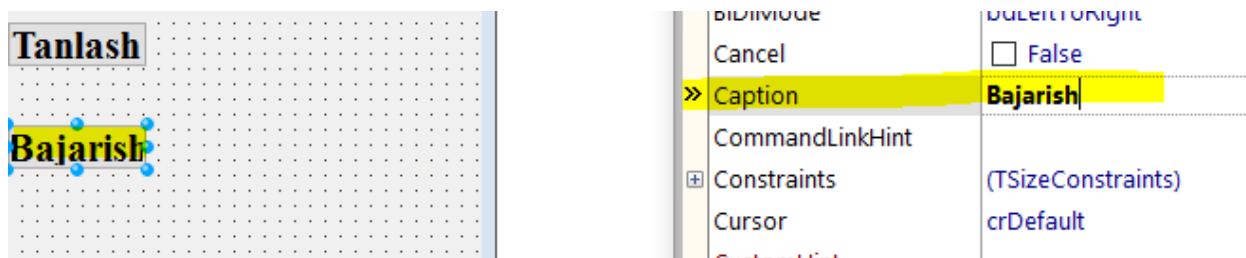
26-rasm. Standart komponentalar ro'yhatidan Button komponentasini formaga qo'yamiz.



27-rasm. Button komponentasi nomini o'zgartiramiz( Caption xususiyati nazarda tutulmoqda!).



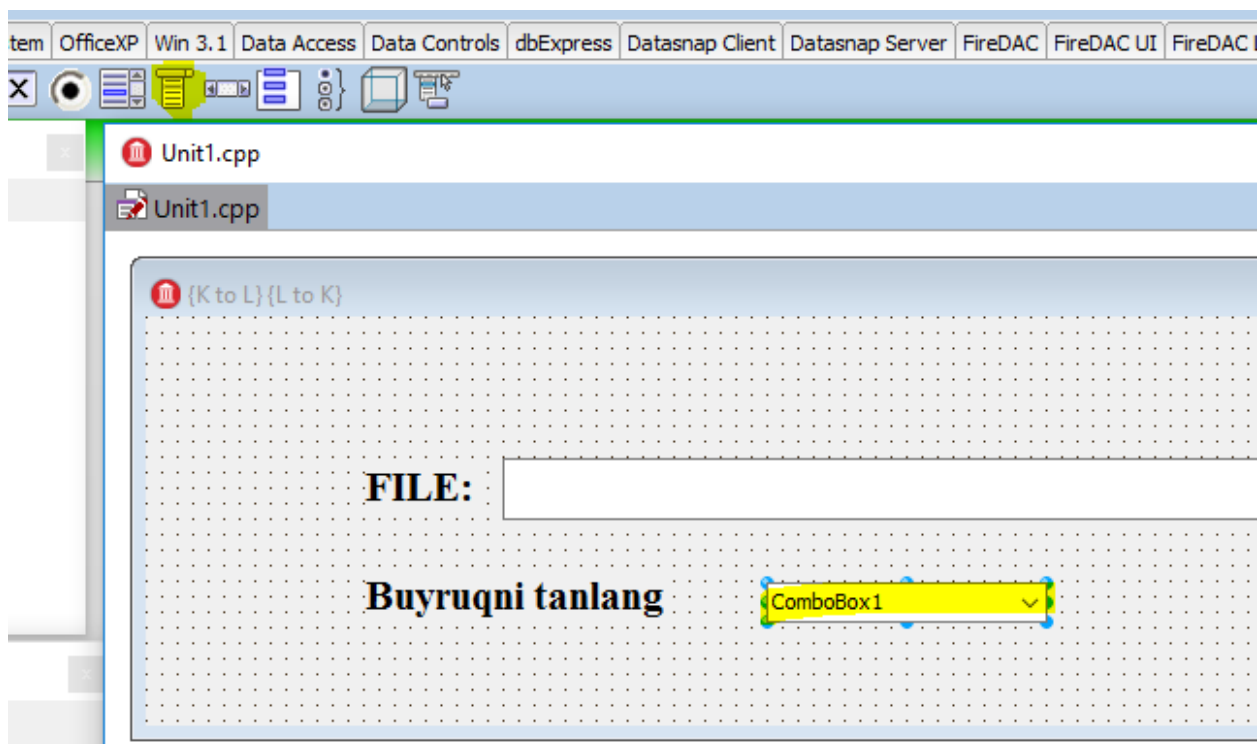
28-rasm. Button komponentasi o'lchamlarini to'g'irlaymiz.



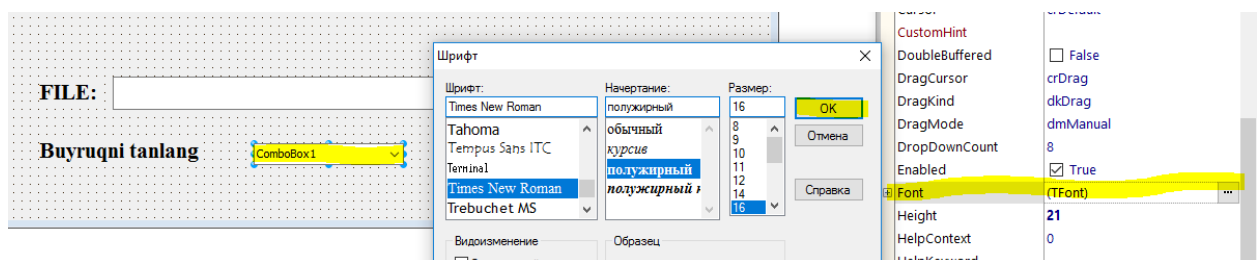
29-rasm. Huddu shunday yo'l bilan "Bajarish" deb nomlangan tugmachani hosil qilamiz.



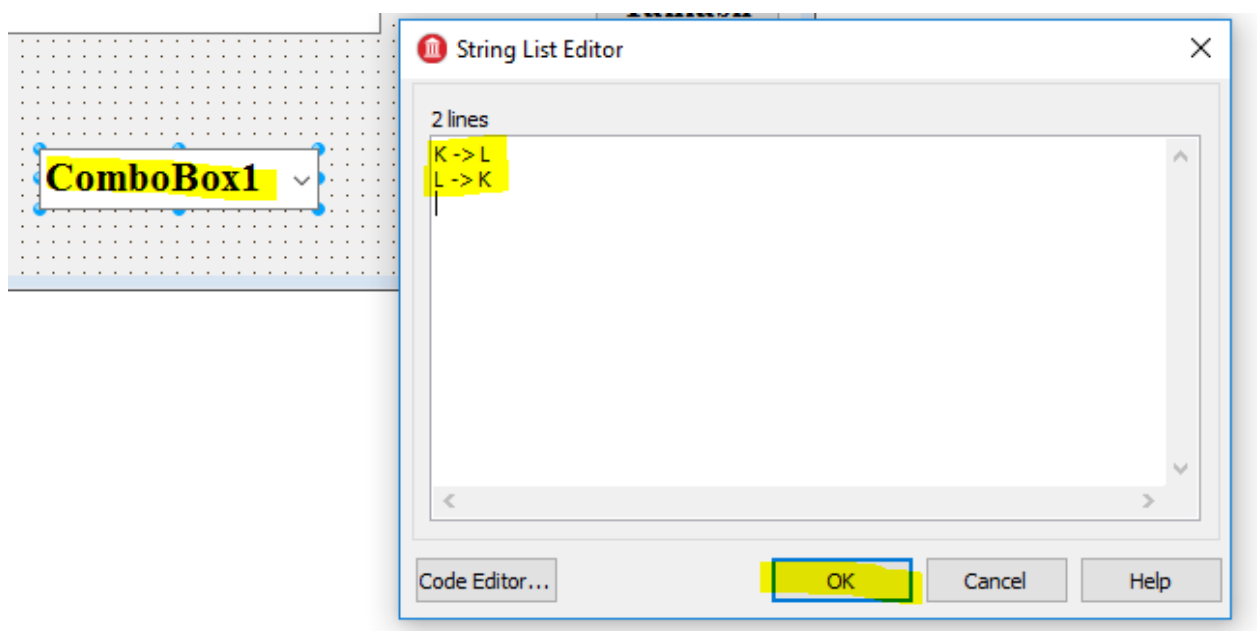
30-rasm. Dastur ishlashi davomida Krill dan lotinga yoki aksincha lotindan krillga o'girilishi uchun tanlash jarayonini tashkil qilish maqsadida "Buyruqni tanlash" xabaruni qoldiramiz.



31-rasm. Formaga ComboBox komponentasini joylaymiz.

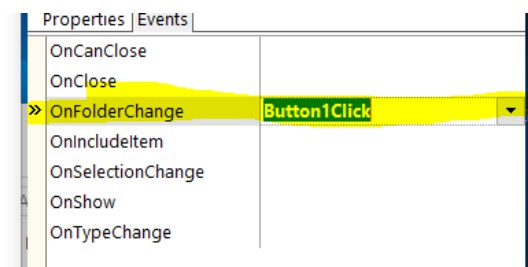
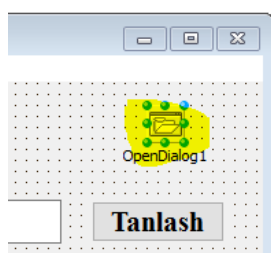


32-rasm. Uning o'lchamlarini to'g'irlaymiz.

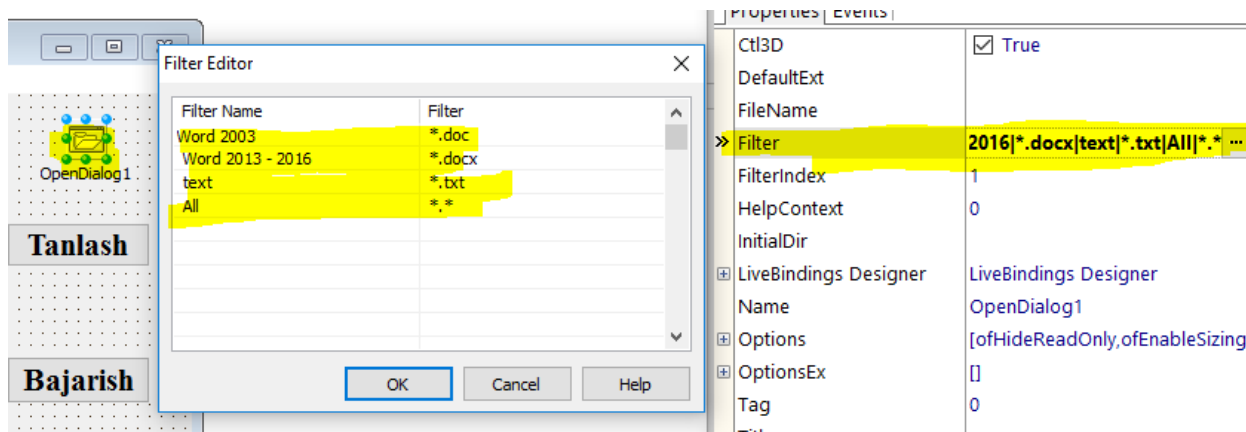


33-rasm. ComboBox komponentasining lines xususiyatiga matnlar kiritamiz.

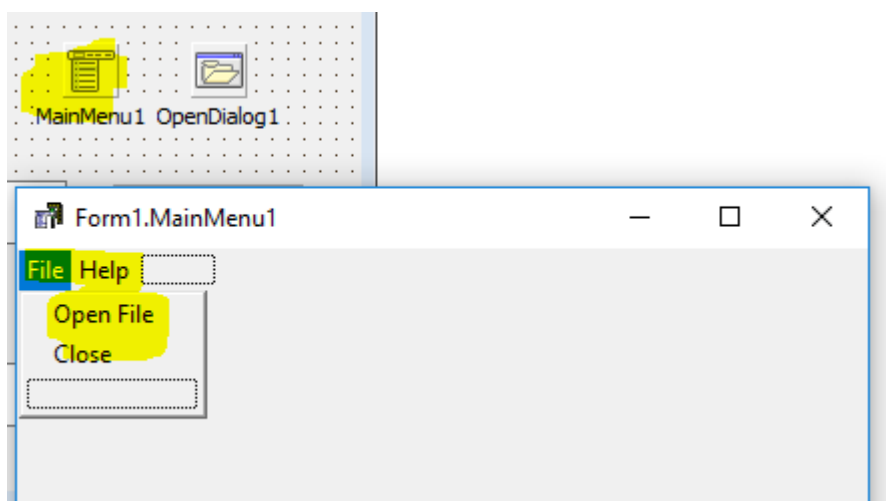




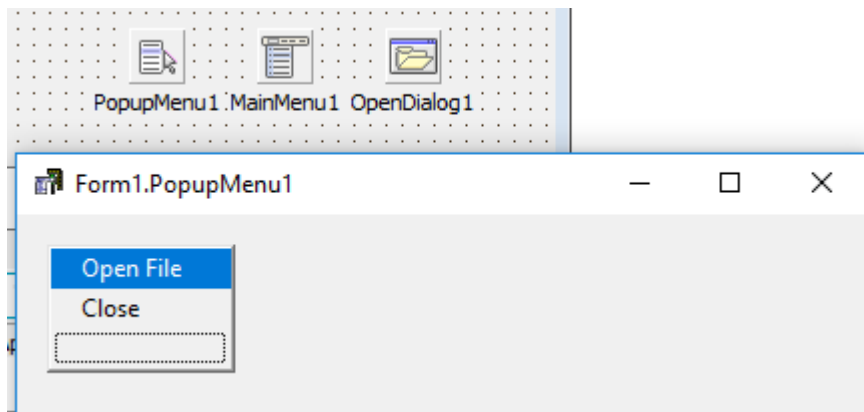
34-rasm. Formamizga OpenFileDialog komponentasini joylashtiramiz. Va uning OnFolderChange xususiyatiga Button1Click xolatini tanlaymiz.(yuqorida tayyorlangan “Tanlash ” tugmachasi boslganda faylni tanlash imkoniyatini berishi uchun)



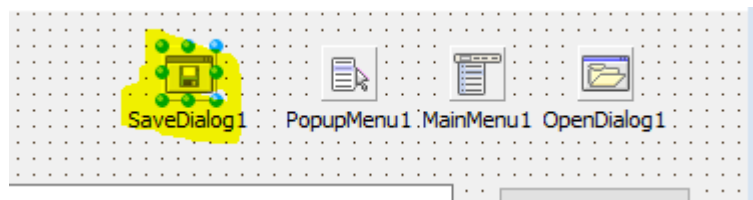
35-rasm. OpenFileDialog komponentasining Filter xususiyatiga matn fayli formatlarini kiritamiz.



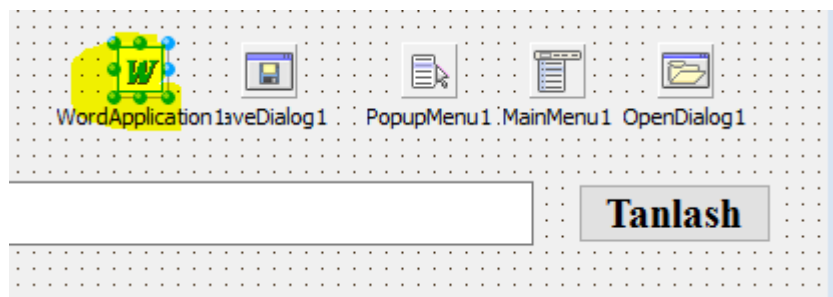
36-rasm. Formaga MainMenu komponentasini foylab. Menyular qatorini hosil qilamiz.



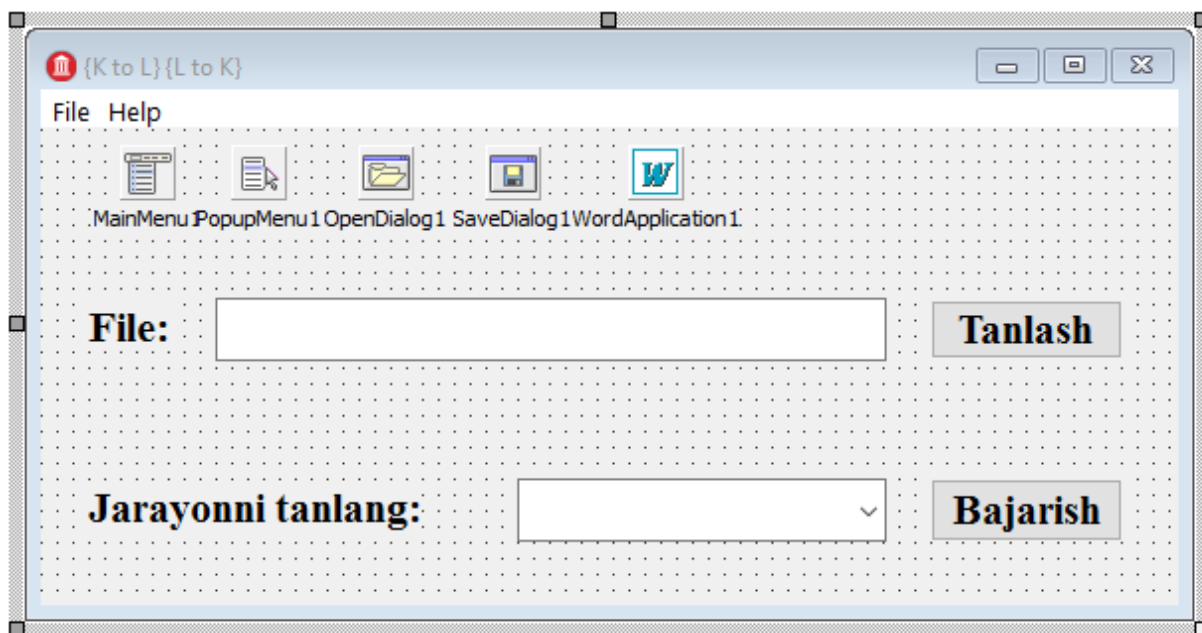
37-rasm. Formaga PopupMenu komponentasini joylab dastur uchun kontekst menyuni hosil qilamiz.



38-Formaga SaveDialog komponentasini joylab qayta ishlangan faylning saqlab qolinishiga erishamiz.



39-Formaga WordApplication komponentasini joylab dasturning word fayllar bilan ishlashini taminlaymiz.



34-rasm. Formani foydalanuvchiga qulay holatda shakillantiramiz. Dasturimizning interfeysi tayyor holda.

Dastur, amaliy matematika va informatika talim yo'nalishi talabalari tomonidan foydalaib tajriba qilindi va yuzaga kelgan muommoli holatlab bartaraf etildi.

## Xulosa

Hozirgi kunda axborot texnologiyalar hamda offis jihozlaridan foydalanmaydigan tashkilotlarning mavjud emasligini hisobga olgan holda offis jihozlaridan foydalanish hamda mehnat samaradorligini oshirish uchun amaliy dasturlarning o'zni beqiyosdir. Tashkilot hodimi mehnatini yengillashtiruvchi amaliy dasturlar sirasiga hujjatlar bilan ishlovchi amaliy dasturlar ham kiradi.

Shu bois biz bitiruv malakaviy ishimizda ishchi hodimlar ko'p duch keladigan muammoli vaziyatlardan biri bo'lgan Krill va lotin alifbosi orasidagi o'girilish munosabatga yechim topish jarayonini ishlab chiqishni maqsad qilib qo'ygan edik. Ushbu bitiruv malakaviy ishni bajarish jarayonida

- Ta'lim muassasalarining (umumiy o'rta ta'lim maktablari, kasb-hunar kollejlari, akademik litseylar) o'quv jarayonida amaliy dasturiy vositalardan foydalanish holati o'rganilib, tahlil qilindi;
  - Amaliy dasturiy vositalarning didaktik imkoniyatlari o'rganib chiqildi;
  - Mavjud amaliy dasturiy vositalar o'rganilib, tahlil qilindi;
  - Amaliy dasturiy vositalarni yaratishda foydalanuvchilarning psixofiziologik xususiyatlari inobatga olinishi o'rganildi;
  - Amaliy dasturiy vositalar yaratish uchun mo'ljallangan dasturiy ta'minotlar o'rganildi va tavsifi keltirildi;
  - Ishlab chiqilgan dasturiy vositalar umumiy o'rta ta'lim maktablarida (2017-2018 o'quv yili davomida Termiz shahar **Qaysi maktabga amaliyotga brogan bo'lsangiz shuni yozing**) sinovdan o'tkazilib, o'quv jarayoni uchun quyidagi samaradorlik berishi aniqlandi:
    - o'quv materialini o'rganish uchun vaqtni tejash;
    - o'quvchiga mustaqil ishlash uchun imkoniyat yaratish;

Ammo shu bilan bir qatorda bu sohada o'z yechimini kutayotgan bir qator muammolar mavjudligi ham aniqlandi. Amaliy dasturiy vositalarni yaratish texnologiyasini egallab olgan mutaxassislarning yetishmasligi, ular jumlasiga mutaxassis bo'lmagan o'qituvchilarning dasturchilar, texnik mutaxassislarsiz amaliy dasturiy vositalarni yaratishlari, muammolarga duch kelishlari, vaqtning ko'proq talab etilishi, dasturiy ta'minot va texnika vositalarini sozlash va foydalanishda vujudga keladigan muammolar, axborotlarni kompyuter ekranidan o'qish jarayonidagi qiyinchiliklar va boshqa jihatlar kiradi.

### **Foydalanilgan adabiyotlar ro'yhati.**

1. Закон Республики Узбекистан «Об образовании» // Ведомости Олий Мажлиса Республики Узбекистан. № 9, ст. 225, 1997 г.
2. O'zbekiston Respublikasi “Kadrlar tayyorlash milliy dasturi” to'g'risidagi qonuni.//O'zbekiston Respublikasi Oliy majlisining IX sessiyasi materiallari. – Toshkent, 29 avgust, 1997 y.
3. O'zbekiston Respublikasining “Axborotlashtirish to'g'risidagi” Qonuni.// O'zbekiston Respublikasi Oliy majlisining materiallari. –Toshkent, 11 dekabr, 2003 y.
4. O'zbekiston Respublikasi Vazirlar Mahkamasining 2001 yil 23 maydagi «2001-2005 yillarda kompyuter va axborot texnologiyalarini rivojlantirish, «Internet»ning xalqaro axborot tizimlariga keng kirib borishini ta'minlash dasturini ishlab chiqishni tashkil etish chora-tadbirlari to'g'risida»gi 230-sonli qarori.// – Toshkent. “Xalq so'zi”. 2001 yil, 24 may. № 101 (2663).
5. O'zbekiston Respublikasi Prezidentining 2002 yil 30 maydagi «Kompyuterlashtirishni yanada rivojlantirish va axborot-kommunikatsiya texnologiyalarini joriy etish to'g'risida»gi PF-3080-son Farmoni.// –Toshkent. “Xalq so'zi”. 2002 yil, 1 iyun. № 116 (2944).
6. O'zbekiston Respublikasi Vazirlar Mahkamasining 2002 yil 6 iyundagi “Kompyuterlashtirishni yanada rivojlantirish va axborot-kommunikatsiya texnologiyalarini joriy etish to'g'risida”gi 200-sonli Qarori.// –Toshkent “O'zbekiston ovozi”. 2002 yil, 8 iyun. № 72.
7. O'zbekiston Respublikasi Vazirlar Mahkamasining “Pedagog kadrlarni qayta tayyorlash va ularning malakasini oshirish tizimini yanada takomillashtirish to'g'risida”gi 25-sonli Qarori.// –Toshkent. 2006 yil 16 fevral.
8. O'zbekiston Respublikasi Oliy va o'rta maxsus ta'lim vazirligi Hay'atining “Oliy ta'lim tizimida o'quv jarayonini va o'quv uslubiy faoliyatni modernizatsiyalash hamda innovatsion ta'lim texnologiyalarini joriy etish to'g'risida”gi Qarori.// –Toshkent. 2010 yil 17 aprel.
9. O'zbekiston Respublikasi Prezidentining “2011-2015 yillar uchun axborot-kommunikatsion texnologiyalar bazasida axborot-kutubxona va axborot-resurs xizmat ko'rsatishni keyinchalik sifatli rivojlantirish bo'yicha chora-tadbirlar to'g'risida”gi PQ-1487-sonli Qarori.// –Toshkent. 2011 yil 23 fevral.
10. Xalq ta'limi Vazirligining “Umumiy o'rta ta'lim maktablarida faoliyat ko'rsatayotgan o'qituvchilarning zamonaviy axborot-kommunikatsiya texnologiyalaridan ta'lim jarayonida foydalanish bo'yicha malakasini oshirishni tashkil etish to'g'risida”gi 125-sonli buyrug'i.// –Toshkent. 2012-yil 21-aprel.
11. Ахборот-коммуникация технологияларини янада ривожлантиришга оид қўшимча чора – тадбирлар тўғрисида : ЎзР Президентини 2005 йил 8

июлдаги ПҚ – 117- сон Қарори // Ўзб.-н. Респ. Қонун хуж. Тўпл.-2005.-№27.- Б.4-5.

12. Ахборот технологиялари соҳасида кадрлар тайёрлаш тизимини такомиллаштириш тўғрисида : ЎзР Президентининг Қарори : 2005 йил 30 май // Ўзб-тон Респ-каси Қонуни хужжатлари тўплами.-2005.-№22.-Б.4-5. // Халқ сўзи.-2005.-3 июнь. // Маърифат.-2005.-июнь.

13. Б. Страуструп. Язык программирования С++. Специальное издание.-М.:ООО «Бином-Пресс», 2006.-1104 с.

14. Павловская Т.А. С++. Программирование на языке высокого уровня – СПб.: Питер. 2005.- 461 с.

15. Подбельский В.В. Язык СИ++.- М.; Финансы и статистика- 2003 562с.

16. Павловская Т.С. Щупак Ю.С. С/С++. Структурное программирование. Практикум.-СПб.: Питер,2002-240с

17. Павловская Т.С. Щупак Ю.С. С++. Объектно- ориентированное программ-ирование. Практикум.-СПб.: Питер,2005-265с

18. Глушаков С.В., Коваль А.В., Смирнов С.В. Язык программирования С++: Учебный курс.- Харьков: Фолио; М.: ООО «Издательство АСТ», 2001.-500с.

19.Юров В., Хорошенко С. Assembler: Учебный курс- СПб, “Питер”,2000.-672с.

20. 1997 йил 27 июлда “Ахборот тизимлари соҳасини қайта ташкил этиш ва бoshқарishни takomillashtirishga oid chora-tadbirlar to‘g‘risida” gi PF-1823-sonli farmoni.

21. 2002-йил 30-майда “Kompyuterlashtirishni yanada rivojlantirish va axborot kommunikatsiya texnologiyalarini joriy etish to‘g‘risida” gi PF – 3080 sonli farmoni.

22. 2005 йил 2 июнда “Ахборот texnologiyalari sohasida kadrlar tayyorlash tizimini takomillashtirish to‘g‘risida”gi PQ-91-sonli qarori.

23. 2005 йил 8 июлда “Ахборот-kommunikatsiya texnologiyalarini yanada rivojlantirishga oid qo‘shimcha chora-tadbirlar to‘g‘risida”gi PQ-117-sonli qarori.

24. 2005 йил 28 sentyabrda “O‘zbekiston respublikasining jamoat ta’lim axborot tarmog‘ini tashkil etish to‘g‘risida” gi PQ-191-sonli qarori.

25. 2007 йил 3 aprelda “O‘zbekiston Respublikasida axborotni kriptografik muhofaza qilishni tashkil etish chora-tadbirlari to‘g‘risida”gi PQ-614-sonli qarori.

26. 2012 йил 21 martda “Zamonaviy axborot-kommunikatsiya texnologiyalarini yanada joriy etish va rivojlantirish chora-tadbirlari to‘g‘risida”gi PQ-1730-sonli qarori.

27. 2013 йил 20 sentyabrda “Mamlakatimizning dasturiy ta’minot vositalari

ishlab chiquvchilarini rag‘batlantirishni yanada kuchaytirish chora-tadbirlari to‘g‘risida”gi PQ-2042-sonli qarori.

28. 2013 yil 25 sentyabda “Milliy geografik axborot tizimini tashkil etish investitsiya loyihasini amalga oshirish chora-tadbirlari to‘g‘risida”gi PQ-2045-sonli qarorilari.

29. Финогенов К.Г. Основы языка Assemblera.-М.: Радио и связь, 2001. - 288 с.

30. Пильшиков В.Н. Упражнения по языку Паскаль-М.: МГУ, 1986.

31. Абель П. Assembler для IBM PC и программирования. 1991. М.: “Высшая школа”, 1992.- 447 с.

32. Скенлон Л. Персональный ЭВМ IBM PC и XT. Программирование на языке Assemblera. -М.: Радио и связь. 1991.- 336 с.

33. Гофман В. Э., Хомоненко А.Д. Delphi 5. - СПб.: БХВ-Санкт-Петербург, 2000. -800с.

34. Немнюгин С.А. Turbo pascal, учебник. Изд. Питер., 2001, -496 с.

35. Поляков Д.Б., Круглов И.Ю. Программирование в среде Турбо-паскаль. (версия 5.5).М.:МАИ,1992.-576с.

36. Абрамов С.А., Гнезделова Капустина Е.Н.и др. Задачи по программированию. - М.: Наука, 1988.

37. Вирт Н. Алгоритмы + структуры данных = программа.-М.:Мир,1985.- 405с.

38. Информатика. Базовой курс. Учебник для Вузов., Санк-Петербург, 2001. под редакцией С.В.Симоновича.

39. Нортон П. Программно-аппаратная организация IBM PC.-М.:Мир,1991.- 327с.

40. Фигурнов В.Э. IBM PC для пользователя. М.: Финансы и статистика. Юнити. 1997.

41. Юров В. Assembler: практикум. -СПб.: Питер, 2002.- 400с.

42. Informatika va programmalsh.O’quv qo’llanma. Mualliflar: A.A.Xaldjigitov, Sh.F.Madraximov, U.E.Adamboev, O’zMU, 2005 yil, 145 bet.

## **Internet resurslari ro'yxati:**

<http://www.xushnudbek.uz>  
<http://www.CyberForum.ru>  
[www.intuit.ru](http://www.intuit.ru)  
[www.book.ru](http://www.book.ru)  
[www.kniga.ru](http://www.kniga.ru)  
[www.gov.uz](http://www.gov.uz)  
[www.ziyonet.uz](http://www.ziyonet.uz)  
[www.ulugov.uz](http://www.ulugov.uz)  
[www.aci.uz](http://www.aci.uz)  
[www.ictp.uz](http://www.ictp.uz)  
[www.terdu.uz](http://www.terdu.uz)  
[www.infoman.uz](http://www.infoman.uz)  
[www.olam.uz](http://www.olam.uz)  
[www.utube.uz](http://www.utube.uz)  
<http://www.uiniti.msk.ru>.  
<http://mcit.aris.ru>  
<http://www.library.fa.ru>.  
<http://www.spb.runnet.ru> <http://sunny.ccas.ru/library.html>.  
<http://www.lrl.da.ru>  
<http://www.infotel.ru>  
<http://www.worldbank.org>  
<http://www.oecdwash.org>  
<http://www.syllabus.com/syllsub.html>  
[www.assist.ru](http://www.assist.ru)  
[www.webmoney.ru](http://www.webmoney.ru)  
<http://sunny.ccas.ru/library.html>  
<http://www.mesi.ru>  
<http://www.CyberForum.ru>  
<http://www.SQL.ru>