



**ЎЗБЕКИСТОН РЕСПУБЛИКАСИ АХБОРОТ ТЕХНОЛОГИЯЛАРИ
ВА КОММУНИКАЦИЯЛАРИНИ РИВОЖЛАНТИРИШ ВАЗИРЛИГИ**
**МУҲАММАД АЛ-ХОРАЗМИЙ НОМИДАГИ ТОШКЕНТ АХБОРОТ
ТЕХНОЛОГИЯЛАРИ УНИВЕРСИТЕТИ**
ФАРҒОНА ФИЛИАЛИ

**ТЕЛЕКОММУНИКАЦИЯ ТЕХНОЛОГИЯЛАРИ ВА
КАСБИЙ ТАЪЛИМ ФАКУЛЬТЕТИ**
ДАСТУРИЙ ИНЖИНИРИНГ КАФЕДРАСИ
“С++ ДА ДАСТУРЛАШ”

Фанидан
амалий машғулотларни ўтказиш учун

УСЛУБИЙ КЎРСАТМА

Таълим йўналишлари

- 5330500 – “Компьютер инжиниринги” (Компьютер инжиниринги,
АКТ сервиси) йўналиши
- 5330600 – “Дастурий инжиниринг” йўналиши
- 5330300 – “Ахборот хавфсизлиги” йўналиши
- 5350400 – “АКТ соҳасида касб таълими” йўналиши

Ушбу услубий кўрсатма С++ да дастурлаш фанидан 5330500 – “Компьютер инжиниринги” (Компьютер инжиниринги, АКТ сервис) йўналиши, 5330600 – “Дастурий инжиниринг” йўналиши, 5330300 – “Ахборот хавфсизлиги” йўналиши ва 5350400 – “АКТ соҳасида касб таълими” йўналишларга амалий машғулотларни олиб бориш учун фойдаланилади.

Ушбу услубий кўрсатма Муҳаммад ал-Хоразмий номидаги Тошкент ахборот технологиялари университети Фарғона филиали томонидан “_____” _____ 2018 йилдаги _____ – сонли баённомаси билан тасдиқланган.

Тузувчилар:

- Х. Ш. Мусаев – ТАТУ ФФ, “Дастурий инжиниринг” кафедраси катта ўқитувчиси
А. А. Хашимов – ТАТУ ФФ, “Дастурий инжиниринг” кафедраси катта ўқитувчиси
А. Қаяюмов – ТАТУ ФФ, “Дастурий инжиниринг” кафедраси ассистент ўқитувчиси

Тақризчилар:

- А. Абдуллаев ТАТУ ФФ, “Дастурий инжиниринг” кафедраси мудирини.ф.д, профессор
Т. Тожиев ФДУ “Ахборот технологияси” кафедраси мудирини, ф-м.ф.н., доцент

Услубий кўрсатма “Дастурий инжиниринг” кафедрасида муҳокама этилган (2018 йил “_____” _____ кунги “_____” – сонли баённома)

**Дастурий инжиниринг
кафедраси мудирини:** _____

А.Абдуллаев

Телекоммуникация технологиялари ва касбий таълим факультетининг Услубий кенгашида тасдиқланган (2018 йил “_____” _____ кунги “_____” – сонли баённома).

**ТТ ва КТ факультети
декани:** _____

О.Қўлдашов

“КЕЛИШИЛДИ”

Ўқув-услубий бўлим бошлиғини: _____

Ш.Умаров

Сўз боши

Компьютернинг дастурий таъминотлари жуда тез суръат билан ўсиб бораётганлиги сабабли дастурий таъминотларни яратувчилари учун жуда кўплаб буйруқларни эсда сақлаб қолиш анча қийинчиликларни келтириб чиқаради. Шу сабабли дастурий таъминотлардан фойдаланиш ва уларни яратишда махсус йўналтирилган, яъни махсуслаштирилган дастурий таъминотларга аҳамият берилмоқда. Амалиёт шуни кўрсатмоқдаки, кичик ўқув курслари кичик доирадаги билимларни бериб, фақат махсус йўналишлар бўйича ўқувчи билимга эга бўлмоқда. Шунингдек, битта йўналиш бўйича чуқур эмас, балки фойдаланувчилар учун зарур буйруқлар ва процедураларни ўрганиш билан чекланиб қолишга мажбур бўлмоқдалар. Бунинг сабаби, жуда кўплаб буйруқларни эсда сақлаб туришнинг имкони йўқлигида.

Баъзи фойдаланувчи яратилган дастурий таъминотлардан қониқиб қолмасдан, балки ўзларининг ҳоҳишларидаги буйруқларни қўшиб ишлатишни ҳамда керак бўлса, баъзи бир буйруқларни янада оптималлаштиришни кўзлайдилар. Шунингдек уларга махсус адабиётлардагидек бир бошдан ҳамма нарсани тушунтиришни кераги бўлмай қолади, чунки улар буйруқларни кўришлари билан гап нима ҳақида ва мақсад нимадан иборатлигини билиб бўладилар.

Ушбу қўлланманинг асосий вазифаси ва келтирилиши қуйидагилардан иборат:

- ана шу тиниб тинчимас изланувчи ўқувчига жуда тез ўрганиш имконини бериш;
- қўлланмада келтириладиган маълумот иш вақтида фойдаланиш кетма-кетлиги билан деярли бир хил;
- бўлимларни келтирилиш кетма-кетлиги жуда ҳам қулай бўлиб, унда махсус адабиётлардан керакли маълумотни излаш вақтини анча камайтиради;
- турли хил ишоратлар қўлланмада келиб чиқадиган турли хил саволларга жуда тез жавоб олиш имконини беради;
- қўлланма бир хил форматда ва мавзулар кетма-кетлиги билан ўқувчининг ўзлаштириши анча тезлашади;
- қўлланманинг хажми унча катта бўлмаган ва қулай форматда ёзилганлиги ўқувчига қўлланмадан фойдаланишни анча осонлаштиради.

Ушбу қўлланмани ўрганишингизда сабр тоқат ва келгусида кучли ва республикамизнинг кўзга кўринган мутахасисларидан бири бўлишингизга ишонган ҳолда омад тилаймиз!

C++ да дастурлаш фани 1, 2 ва 3 семестр давомида ўқитилади.
C++ да дастурлаш фани

5330500 – “Компьютер инжиниринги” (Компьютер инжиниринги, АКТ сервис) йўналиши;

5330600 – “Дастурий инжиниринг” йўналиши;

5350300 – “Ахборот хавфсизлиги” йўналиши;

5350400 – “АКТ соҳасида касб таълими” йўналишларга ўқитилади

Амалий иш №1: C++ Builder интерфейси ва асосий менюлари.

Ишдан мақсад: Ушбу амалий ишда ишида талабалар Borland C++Builder 6 ёки Embarcadero RAD Studio XE2 ни ўрнатишни ўрганадилар.

Керакли жиҳозлар ва ускуналар: Компьютер, видео проектор.

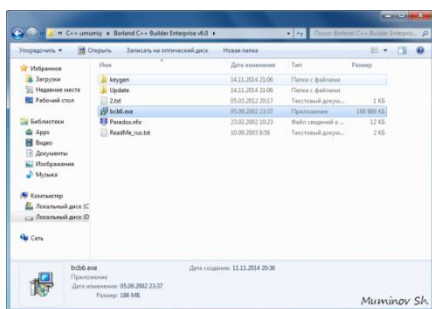
Назарий маълумот.

C++ Builder – Borland фирмасининг янги маҳсулоти бўлиб, C++ тилидаги иловаларни тез ишлаб чиқариш учун мўлжалланган (**RAD-Rapid Application Development**). C++Builder ёрдамида Windows дастурларини C++ тилида тез ва осон тузиш мумкин. Win32 консол иловаларини ва фойдаланувчи учун график интерфейсни яратиш мумкин (**GUI- graphical user interface**). Win32 GUI иловаларини C++Builder ёрдамида яратаётганимизда, биз RAD мухитидаги C++ тилининг ҳамма кучини ишлата оламиз. Бу дегани, биз фойдаланувчи интерфейсини drag-and-drop техникаси ёрдамида ярата оламиз. Биз формаларга ОСХ бошқарув элементларини жойлаштириб, қисқа вақт ичида WEB браузерларга ўхшаган махсус иловаларни яратишимиз мумкин.

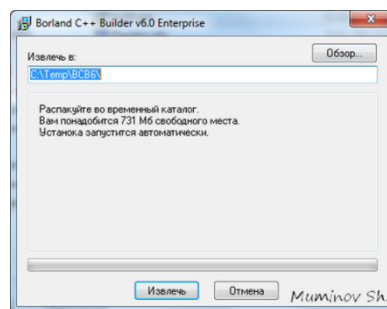
Амалий қисм

C++ Builder 6 ни ўрнатиш.

Borland C++ Builder 6 ни ўрнатиш учун, у сақланган каталог ичидаги **bcb6.exe** файлини ишга туширамиз (1.1.1 – расм).



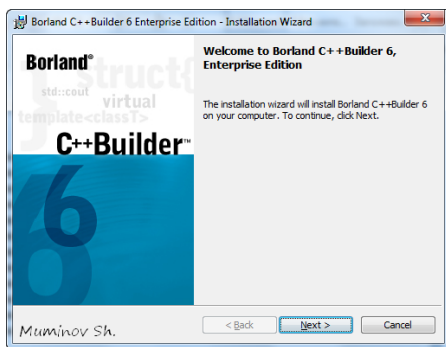
1.1.1 – расм: bcb6.exe файл жойлашган каталог



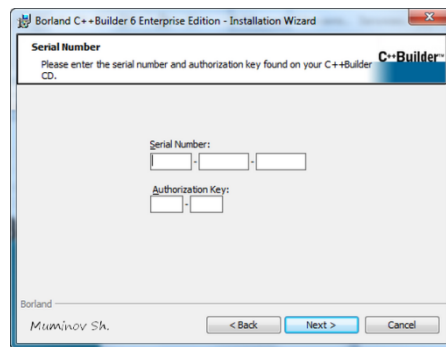
1.1.2 – расм: Маълумотларни архивдан чиқариш ойнаси

Ундан сўнг 1.1.2 – расмдаги ойна ҳосил бўлади. Бу ойнада файли архивдан чиқариш учун жой сўрайди. Биз C:\Temp\BCB6\ деб номланган жойни кўрсатиб у ерга керакли файлларни архивдан

чиқариб оламиз. Архивдан чиқариш жараёни тугаганидан сўнг, Borland C++ Builder 6 ни ўрнатиш жараёни бошланади (1.1.3 – расм).

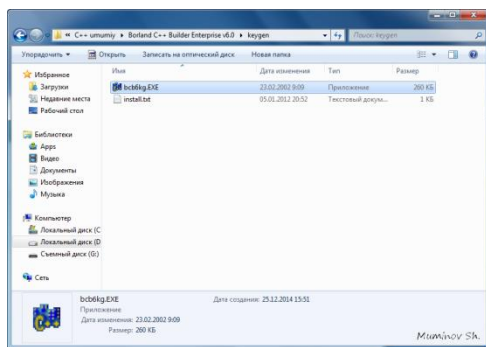


1.1.3 – расм: Borland C++ Builder 6 дастурини ўрнатиш бош ойнаси

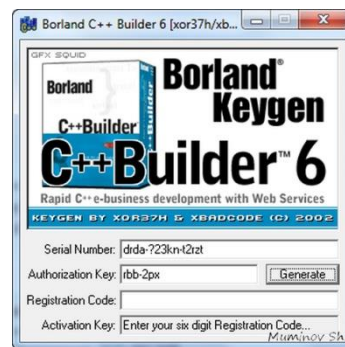


1.1.4 – расм: Дастурни ўрнатишдаги серия рақамини киритиш ойнаси

1.1.4 – расмда дастурнинг серия рақами ва рухсат бериш калитини киритишни сўрайди. Бу икки кодни олиш учун keugen папкасидаги **bc6kg.exe** файлини ишга туширамыз (1.1.5 – расм).

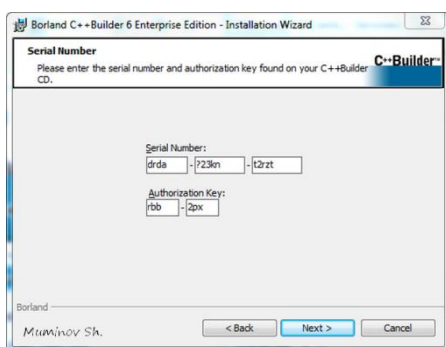


1.1.5 – расм: keugen папкаси ичидаги файллар

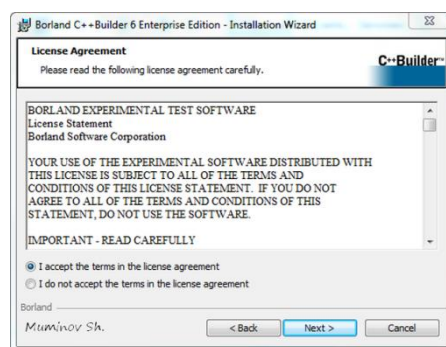


1.1.6 – расм: bc6kg.exe файлини ишга туширишдан ҳосил бўлган ойна

bc6kg.exe файлини ишга туширилгандан сўнг 1.1.6–расмдаги ойна ҳосил бўлади. Бу ойнадан Generate тугмасини босиб керакли кодларни оламиз. Уларни яъни берилган кодларни 1.1.4–расмдаги ойнага бирин–кетинликда жойлаштирамыз ва Next тугмасини босамиз (1.1.7 – расм).



1.1.7 – расм: Дастурни ўрнатишдаги серия рақамини киритиш ойнаси

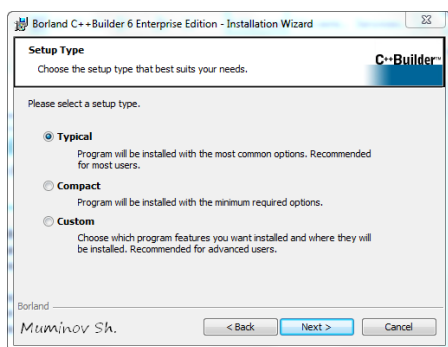


1.1.8 – расм: Лицензия рухсат бериш ойнаси

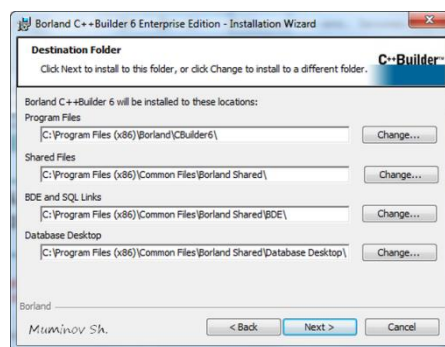
1.1.8–расмда келтирилган ойнасидаги I accept the terms in the license agreementни танлаб Next тугмасини босамиз. Шундан сўнг 1.1.9–расмдаги ойна ҳосил бўлади. Бу ойнада дастурнинг ўрнатиш турини танлашимиз керак. Қуйида дастурнинг ўрнатиш турлари берилган:

- Typical – дастурни умумий ҳолатда ўрнатиш, фойдаланувчилар учун тавсия этилади;
- Compact – дастурни минимал ҳолатда ўрнатиш;
- Custom – дастурни керакли бўлган бўлимларини танлаб ўрнатиш, етакчи фойдаланувчилар учун тавсия этилади.

Ушбу ойнадан кўрсатилган ойналарнинг хаммасидан Next тугмасини босамиз. 1.1.10 – расмда кўрсатилган ойнада дастурни ўрнатилаётган жойни кўрсатиш ойнаси келтирилган, уни ўзгартириб ўзимиз учун қулай жойни танлашимиз мумкин.

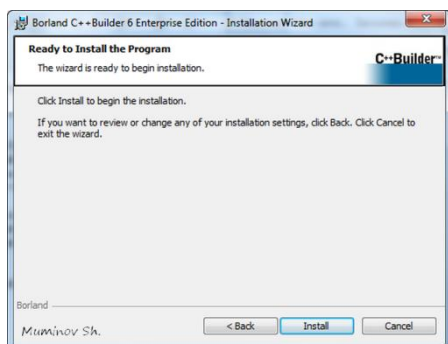


1.1.9 – расм: Дастурни турини танлаш ойнаси

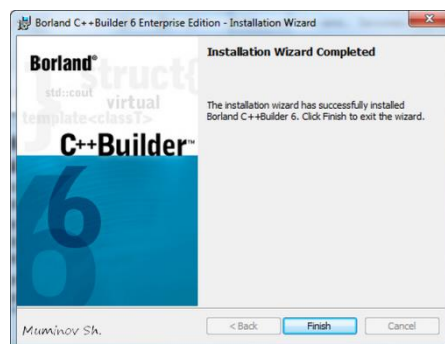


1.1.10 – расм: Дастурни ўрнатиш жойини кўрсатиш ойнаси

Шундан сўнг Next тугмасини босамиз. Шунда 1.1.11 – расмдаги ойна ҳосил бўлади ва Install тугмасини танлаймиз ва ўрнатиш жараёни бошланади.



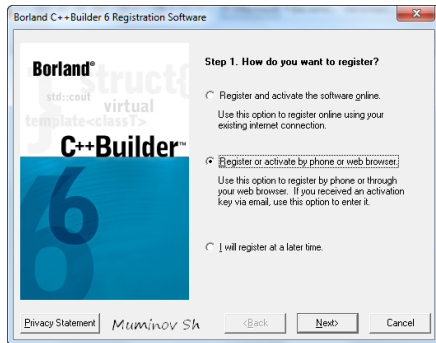
1.1.11 – расм: Дастурни ўрнатиш бошлаш ойнаси



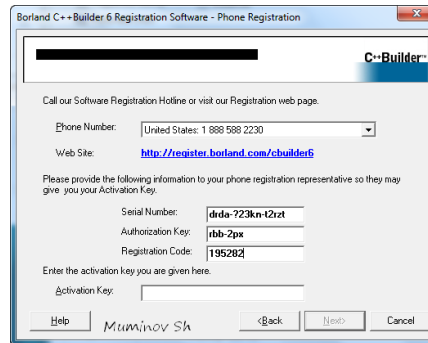
1.1.12 – расм: Дастурни ўрнатиш тугатиш ойнаси

Дастурни ўрнатиш тугагандан сўнг Next тугмаси босилади. Жараён яқунлангандан кейин дастурни ўрнатиб бўлганини

билдирувчи ойна ҳосил бўлади. У ойнадан Finish тугмасини танлаб дастурни ўрнатишни яқунлаймиз. Дастурни ўрнатганимиздан сўнг уни рўйхатдан ўтгазишимиз керак, яъни Регистрация қилишимиз керак. Унинг учун дастурни дастлаб ишга тушираемиз. Шунда 1.1.13 – расмдаги ойна ҳосил бўлади. Бу ойнадан Register or activate by phone or web browser ни танлаб Next ни босамиз.



1.1.13 – расм: Дастурни регистрация қилиш ойнаси

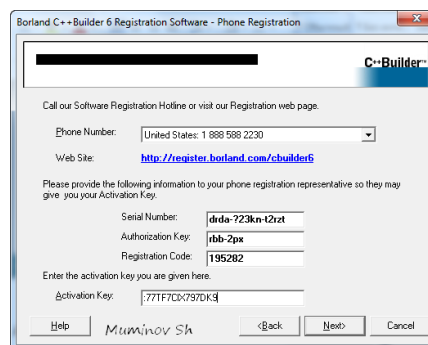


1.1.14 – расм: BCB6 Registration Software – Phone Registration ойнаси

Ундан сўнг экранда 1.1.14 – расмдаги ойна ҳосил бўлади ва биз бу ойнадан Registration Code ни олиб Borland C++ Builder 6 Keygen деб номланган ойнадаги Registration Code майдонига жойлаштирамиз (1.1.15 – расм), кейин бизга Activation Key берилади уни кўчириб олиб Borland C++ Builder 6 Registration Software – Phone Registration деб номланган ойнадаги керакли жойга жойлаштирамиз ва Next ни босамиз (1.1.16 – расм).

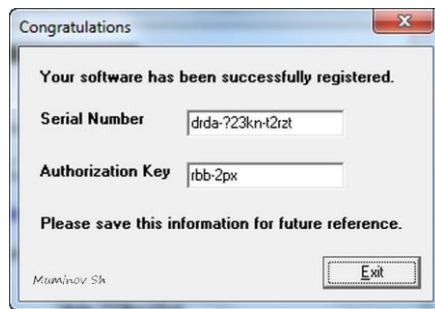


1.1.15 – расм: Borland C++ Builder 6 Keygen ойнаси



1.1.16 – расм: Activation Key ни жойлаштириш ойнаси

Next тугмасини босганимиздан сўнг бизга рўйхатдан ўтиш яқунланганлигини билдирувчи ойна ҳосил бўлади ва ундан Exit тугмасини босиб чиқиб кетишимиз мумкин (1.1.17 – расм).



1.1.17 – расм: Дастурни регистрация қилишни тугатиш ойнаси

Дастурни ишга тушириш Пуск тугмаси ёрдамида амалга оширилади.

Пуск=>Всё программы =>Ворland C++ Builder 6 =>C++ Builder 6

Назорат саволлари:

1. C++ Builder дастурини ўрнатиш тушинтириб беринг?
2. Бош менюнинг бўлимларини гапириб беринг?
3. Файл менюси бандларини тушинтириб беринг?
4. Дастур интерфейсини тушинтириб беринг?

Амалий иш №2: Standard компоненталар палитраси.

Хусусият, услуб ва ходиса

Ишдан мақсад: Ушбу амалий ишида талабалар C++Builder 6 дастурининг стандарт компоненталар палитрасини ўрганадилар.

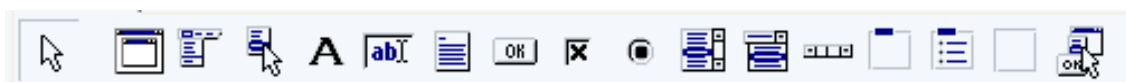
Керакли жиҳозлар ва ускуналар: Компютер, ведио проектер

Назарий маълумот


Объектли дастурлаш – бу таркибида объект тушунчаси бўлган дастурларни қайта ишлаш услубидир. Қўйилган масалалар объектлар ёрдамида ечилса, улар объектли дастурлар деб аталади. Ундаги асосий дастур объектлар кетма-кетлигини ўзида сақлайди ва уларни бир-бири билан боғлайди. Класс – бу мураккаб кўринишга эга бўлиб, бир жойга жамланган функциялар ва маълумотлар ёзувларини ўзида мужассамлаштиради.


Иловалар интерфейси яратиш учун C++Builder визуал компоненталарининг кенг тўпламини тақдим қилади. Уларнинг асосийлари компоненталар палитрасининг Standard, Additional va Win32 варақларида жойлашган.


Standard варағидаги интерфейс компоненталарининг аксарияти Windows операцион тизими бошланғич версияларида ишлатилган объектлардан иборат:



2.2.1 – расм: Standard компоненталар палитраси

 Frames – Фреймлар;

 MainMenu –асосий меню. Бош меню командалари панелини ва уларга мос келадган тушиб қоладиган менюларни яратади. Барча меню буйруқларининг идентификаторлари менюнинг ҳар қандай конкрет буйруғига кириш ҳуқуқига эга бўлган Items хусусияти билан аниқланади, AutoMerge хусусияти Merge ва Unmerge методлари билан биргалиқда турли шаклдаги менюларнинг бирлашиш жараёнини бошқаради;


 PopUpMenu – оқиб чиқувчи меню. Шакл ёки биронта бошқа компонента учун махсус меню яратади. Эътиборга олинг, айнан шу мақсад учун ҳар қандай бошқа компонента PopUpMenu хусусиятига

эга бўлиб, бу хусусиятда сиз унинг билан боғлиқ менюга иқтибос қилишингиз мумкин.


Агар сиз сичқончанинг ўнг тугмасини шаклга ёки берилган компонента мансуб бўлган бирон бошқа элементга босиш билан махсус меню экранда пайдо бўлишини хоҳласангиз, AutoPopup хусусиятининг true қийматини ўрнатинг. Воқеа қайта ишлатгичи – OnPopup ёрдамида бевосита махсус менюнинг пайдо бўлиши олдидан бажариладиган қиймат қайтармайдиган функцияни аниқлаш мумкин.

A Label – фойдаланувчи соҳасидаги бирор жойни белгилаш. Шаклда таҳрир қилиб бўлмайдиган статик матннинг тўртбурчак соҳасини акс эттиради. Одатда матн бошқа компонента номидан иборат бўлади.


Ном матни Caption хусусиятининг қийматидир. Alignment хусусияти матнни текислаш усулини аниқлайди. Шрифт ўлчами автоматик тарзда соҳанинг максимал тўддирилишига мос келиши учун, AutoSize хусусиятининг true қийматини ўрнатинг. Калта соҳа ичида матннинг ҳаммасини кўриш имконига эга бўлиш учун, WordWrap хусусиятининг true қийматини беринг. Transparent хусусиятининг true қийматини ўрнатсангиз, бошқа компонентанинг бир қисмини тўғри унинг устида жойлаштирилган ном орасидан кўришиб турадиган қилишингиз мумкин.

 Edit – бир сатрли матн таҳрирлагич. Ахборот якка сатрини киритишидаги таҳрир қилинадиган тўртбурчак соҳани акс эттиради. Таҳрир соҳасининг ичидаги бошланғич нарсаларни Text хусусиятининг қиймати бўлган сатр аниқлайди.


TEdit компонентаси TCustomEdit синфининг тўғридан-тўғри ҳосиласи бўлиб, унинг барча хусусиятлари, методлари ва фоқеаларига ворислик қилади.

 Memo – кўп сатрли матн таҳрирлагич. Ахборотнинг кўплаб сатрини киритишидаги таҳрир қилинадиган тўртбурчак соҳани акс эттиради. Таҳрир соҳасининг ичидаги бошланғич нарсаларни Lines хусусиятининг қиймати бўлган сатрлар массиви аниқлайди. Ушбу хусусият қиймати устунида тугмачани боссангиз, рўйхат элементлари муҳаррирининг дарчаси очилади.


TMemo компонентаси TCustomMemo синфининг тўғридан-тўғри ҳосиласи бўлиб, унинг барча хусусиятлари, методлари ва воқеаларига ворислик қилади.

 Button – стандарт тугма. Ёзувли тўртбурчак тугмани яратади. Тугмача босилганда, дастурда бирон-бир хатти-ҳаракат бажарилади. Тугмачалар кўпроқ диалогли дарчаларда қўлланади. Default хусусиятининг true қиймати томонидан танлаб олинган яширин тугмача, диалог дарчасида ҳар гал Enter клавишаси босилганда, OnClick ходиса қайта ишлатгичини ишга туширади. Cancel хусусиятининг true қиймати танлаб олган узиш тугмачаси, диалог дарчасида ҳар гал Escape клавишаси босилганда, OnClick воқеа қайта ишлатгичини ишга туширади.

TButton компонентаси TButtonControl синфининг ҳосиласи ҳисобланади.

 CheckBox – бир-бирига боғлиқ бўлмаган танлаш (белгилаш) дарчаси. Иккита ҳолатга ҳамда тавсифий матнга эга бўлган квадрат чекбоксни яратади. Бокс ҳолатини билдирувчи «check» бирон-бир вариантнинг танланишига мос келади (бокс устидан тортилган чизиқ билан белгиланади), «unchecked» ҳолати эса танлов олиб ташланишига мос келади – бунда Checked компонентасининг хусусияти мос равишда ўзгаради ҳамда OnClick воқеаси юзага келади. Тавсифий матн Caption хусусиятида сақланади. AllowGrayed хусусиятининг true қийматини ўрнатиб, боксни тўқроқ рангли (масалан, кулранг) қилиш мумкин. State хусусияти жорий ҳолатни ва бокс рангини акс эттиради.


TCheckBox компонентаси TButtonControl синфининг ҳосиласидир.

 RadioButton – бир соҳадаги бир-бирига ўзаро инкор қилувчи танлаш. Иккита ҳолатга ҳамда тавсифий матнга эга бўлган юмалоқ тугмачани яратади (бунда тавсифий матн юмалоқ тугмачанинг вазифасини спецификация қилади).

Radio-тугмалар бир-бирини истисно қиладиган танлов вариантларининг тўпламидан иборат: яъни ушбу вақт дақиқасида фақат битта тугма танлаб олинishi мумкин (ички қора доирача билан белгиланади), аввал танланган тугмадан эса танлов автоматик тарзда олинади. Radio-тугма босилганда, checked компонентасининг хусусияти ҳам мос равишда ўзгаради ва OnClick воқеаси юзага келади.

Одатда radio-тугмалар аввалдан шаклда ўрнатилган контейнер ичига жойлаштирилади. Агар битта тугма танланган бўлса, ушбу гуруҳга мансуб барча бошқа тугмаларнинг танловлари автоматик тарзда олиб ташланади. Масалан, шаклдаги иккита radio-тугма, агар улар бошқа-бошқа контейнерларда жойлашган бўлсагина, бир пайтнинг ўзида танлаб олиниши мумкин. Агар radio-тугмаларнинг гуруҳланиши очиқ-ойдин берилмаган бўлса, бу ҳолда уларнинг ҳаммаси, яширин ҳолда, контейнер дарчалари (Tform, TGroupBox ёки TPanel) дан бирида гуруҳланади.

TRadioButton компонентаси TButtonControl синфининг ҳосиласидир.


 ListBox – танланадиган сатрлар рўйхати. Танлаш, қўшиш ёки ўчириш учун мўлжалланган матн вариантлари рўйхатининг тўртбурчак соҳасини акс эттиради.

Агар рўйхатдаги барча элементлар ажратилган соҳага сиғмаса, рўйхатни айлантириш линейкаси ёрдамида кўриб чиқиш мумкин. Рўйхат элементлари Items хусусиятининг ичида, дастур бажарилиш вақтида танлаб олинадиган элемент рақами эса ItemIndex хусусиятининг ичида жойлашган бўлади. Рўйхат элементлари матн муҳаррирининг дарчаси Items хусусияти қийматининг графасида тугмача билан очилади. Рўйхат элементларини Items объектининг Add, Append, Delete ва Insert методлари ёрдамида динамик тарзда қўшиш, ўчириш, орасига жойлаш ва ўрнини алмаштириш мумкин. Масалан:

```
1: LisBox1->Items->Add(«Ro'yxatning oxirgi elementi»);
```

Sorted хусусиятининг true қиймати рўйхат элементларини алифбо тартибида навларга ажратиб жойлаштиради.

TListBox компонентаси TcustomListBox синфининг ҳосиласи бўлиб, унинг барча хусусият, метод ва воқеларига ворислик қилади.

 ComboBox – очилувчи рўйхатга эга бўлган танлаш. Таҳрир соҳаси ҳамда матн вариантларининг тушиб қоладиган рўйхати комбинациясини танлаш учун яратади.

Text хусусиятининг қиймати бевосита таҳрир соҳасига киритиб қўйилади. Фойдаланувчи танлаб олиши мумкин бўлган рўйхат элементлари Items хусусиятининг ичида бўлади, дастурнинг


бажарилиш пайтида танлаб олиниши мумкин бўлган элемент рақами ItemIndex хусусиятининг ичида бўлади, танлаб олинган матннинг ўзи эса SelText хусусиятининг ичида бўлади. SelStart ва SelLength хусусиятлари матннинг қайси қисмини танлаб олишни белгилаб бериш ёки матннинг қайси қисми танлаб олинганини билиш имконини беради.

Items объектининг Add, Append, Delete ва Insert методлари ёрдамида рўйхат элементларини динамик тарзад қўшиш, ўчириш орасига қўйиш ва ўрнини алмаштириш мумкин, масалан:

```
1: ComboBox1->Items->Insert(0, «Ro'yxatdagi birinchi element»);
```

Sorted хусусиятининг true элементи рўйхат элементларини алифбо тартибида навларга ажратилишини таъминлайди. TComboBox компонентасининг турини Style хусусиятидан танлаб олиш мумкин.

TComboBox компонентаси TCustomComboBox синфининг ҳосиласи бўлиб, унинг барча хусусиятлари, методлари ва воқеаларига ворислик қилади.


 ScrollBar – соҳа бўйича ҳаракатлантириш йўлаги. Дарча, шакл ёки бошқа компонента ичидагиларини кўриб чиқиш учун, масалан, бирон–бир параметр қийматини берилган интервал ичида ҳаракатланиши учун, югургичли айлантериш линейкасини яратади.

Айлантирилаётган объект хулқ-атворини OnScroll воқеалар қайта ишлатгичи аниқлайди. Фойдаланувчи линейканинг ўзида сичқончани босганда (югургичнинг ҳар иккала томонида), югургич қанчага сурилиши кераклигини LargeChange хусусиятининг қиймати аниқлаб беради. Фойдаланувчи сичқончаним стрелкали тугмачалар (линейка охиридаги) устида босганда ёки позициялаш тугмачаларини босганда, югургич қанчага сурилиши кераклигини SmallChange хусусиятининг қиймати аниқлаб беради.

Min ва Max хусусиятларининг қийматлари югургичнинг йўл қўйилиши мумкин бўлган жой алмашинувлари интервалларини белгилайди. Сизнинг дастурингиз югургични Position хусусиятининг қиймати аниқлаб берадиган керакли позицияга жойлаштириши мумкин. SetParams методи бир пайтнинг ўзида Min, Max ва Position га тегишли барча хусусиятлар қийматларини аниқлаб беради.

 GroupBox – номланадиган контейнер соҳа;


Тўғри бурчакли рамка кўринишидаги контейнер бўлиб, у қандайдир бир интерфейс элементларининг мантиқий боғланган гуруҳини шаклда визуал бирлаштиради. Бу компонента Windows нинг бир номдаги объектнинг инкапсуланишидан иборат.

 RadioGroup – ўзаро бир–бирини инкор қилувчи танлашлар гуруҳи. Тўғри бурчакли рамка кўринишидаги контейнер бўлиб, у бир–бирини мантиқан истисно қиладиган radio-тугмалар гуруҳини шаклда визуал бирлаштиради.

Radio-тугмалар битта контейнерга жойлаштирилганда «гуруҳланади». Бу гуруҳдан фақат битта тугмача танлаб олиниши мумкин. TRadioGroup компонентасига тугмаларни қўшиш учун, Items хусусиятининг таҳрири бажарилиши керак. Items хусусиятининг навбатдаги сатрига ном берилса, шу тугма гуруҳловчи рамкада пайдо бўлади. Ушбу дақиқада қайси тугма танлаб олиниши кераклигини ItemIndex хусусиятининг қиймати аниқлаб беради. Columnsхусусиятининг тегишли қийматини жойлаштириб, сиз radio тугмаларни бир неча устунга гуруҳлашингиз мумкин.

Panel – контейнер соҳа. Бошқа компонентларни ўз ичига олиши мумкин бўлган бўш панелни яратади. Сиз TPanel дан ўз шаклингизда инструментлар панели ёки ҳолатлар сатрларини яратиш учун фойдаланишингиз мумкин.

TPanel панел компонентаси TCustomPanel синфининг ҳосиласи бўлиб, унинг барча хусусиятлар, методлари ва воқеалари тўлиқ ворислик қилади.

 ActionList – амаллар (стандарт буйруқлар, боғланадиган функциялар) рўйхати.

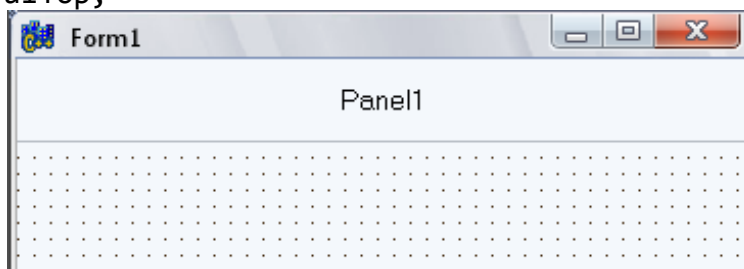
Визуал компоненталарнинг баъзи бир хусусиятларини кўриб чиқамиз. Align хусусияти қуйидаги қийматларнинг бирини қабул қилиши мумкин:

- alNone –тўғрилаш амалга оширилмайди. Компонента иловани яратиш пайтида қаерга жойлаштирилган бўлса, шу жойда қолади;
- alTop –компонента контейнернинг қисмига кўчирилади, компонента баландлиги ўзгармайди, эни эса контейнер энига тенг бўлади;
- alBottom – alTop га ўхшаш, фақат компонента контейнер пастига жойлашади;

- `alLeft` –компонента контейнернинг чап томонига кўчади, эни ўзгармайди, бўйи эса контейнер бўйига тенглашади;
- `alRight` – `alLeft` га ўхшаш, фақат компонента контейнернинг ўнг томонига кўчади;
- `alClient`–компонента контейнерни тўла эгаллайди.

Мисол:Panelни формага нисбатан тўғрилаш.

1: `Panel1->Align=alTop;`



2.2.2 – расм: Panelни юқорига жойлаштириш.

`Color` компонента фонининг рангини аниқлайди. `Color`хусусиятининг қиймати 4 байтли 16 саноқ системасидаги сон бўлиб, унинг катта байти ранг палитрасини аниқлайди (одатда \$00), кичик учта байти қизил, яшил ва кўк рангларнинг RGB интенсивлигини аниқлайди. Бу рангларни турли нисбатдаги аралашмасидан натижавий ранг ҳосил бўлади. Масалан, \$000000 – қора, \$FFFFFF–оқ, \$0000FF–қизил, \$00FF00 –яшил, \$FF0000 –кўкрангни беради. Ранглар қийматини константалар орқали бериш анча қулай. Масалан, `clAqua`–оч кўк (\$FFFFFF00), `clBlack` – қора (\$000000) ваҳоказо.

`Ctrl3D` хусусияти `Boolean` турида бўлиб, визуал компонента кўринишини аниқлайди. Агар `Ctrl3D` қиймати `false` бўлса, компонента икки ўлчамли тасвир кўринишида, агар `true` бўлса–уч ўлчовли тасвирланади (келишилган ҳолдаги кўриниш).

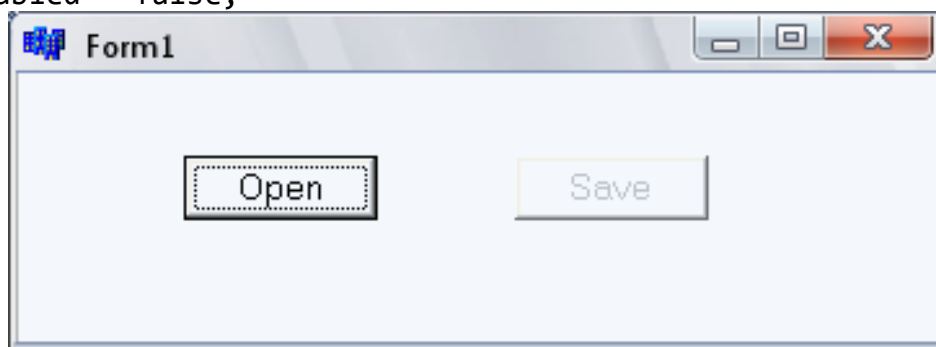
`Cursor` хусусияти сичқонча кўрсатгичининг кўринишини аниқлаб беради. Бу кўринишлар ранг – баранг, улар ичида энг асосийлари қуйидаги ўзгармаслар билан берилади:

- `crDefault`–сичқонча кўрсатгичи келишув бўйича (одатдаги стрелка) ;
- `crNone`– кўрсатгич кўринмайди;
- `crArrow`–кўрсатгич стрелга кўринишида;
- `crCross`–кўрсатгич хоч кўринишида;
- `crHourGlass`–кўрсатгич кутиш режими кўринишида.

`Enabled` хусусияти `Boolean` туридабўлиб, компонентани фаоллигини, яъни сичқончадан ёки клавиатурадан келаётган

сигналларга акс таъсир бера олишини аниқлайди. Агар хусусият true (келишилган ҳолат қиймати) қийматга эга бўлса, компонента фаол ҳисобланади. Компонентанинг фаол бўлмаган ҳолатини сарлавҳанинг ёки матннинг оқарган рангда эканлиги билдиради. Айрим ҳолларда қандайдир компонента ўчирилган (блокировка қилинган) бўлади, агар у билан боғлиқ қандайдир амални бажаришни иложи бўлмаса. Масалан, Save тугмаси ҳужжатни сақлаш амалини бажарадиган бўлсин, агар сақланадиган ҳужжатнинг ўзи бўлмаса, сақлаш амалининг маъноси йўқ. Бундай ҳолларда тугма фаол бўлмаган ҳолатга ўтказилади:

1: Save->Enabled = false;



2.2.3 – расм: Тугманинг Enabled хусусияти.

Font хусусияти визуал компонента акланувчи матн шрифтини аниқлайди. Ўз навбатида TFont синфи шрифт параметрларини бошқаришга имкон берувчи хусусиятларга эга. Буларнинг ичида асосийлари қуйидагилар:

- Name–шрифтни номини аниқлайди;
- Size–пунктларда шрифт ўлчамини беради;
- Style–шрифт стилини беради;
- Color – Matn рангини бошқаради.

Height ва Width хусусиятлари компонентанинг мос равишда вертикал ва горизонтал ўлчамларини пикселларда беради.

Left ва Top хусусияти компонентанинг у жойлашган контейнерга (форма ёки панелга) нисбатан координаталарини аниқлайди. Ўз ўрнида форма ҳам компонента ва унинг координаталари экраннинг чап юқори бурчагига нисбатан аниқланади.

Hint хусусияти курсор компонента соҳасида ва бир неча сония ҳаракатсиз ҳолатда бўлганда экранга ёрдамчи матнни чиқаради. Ёрдамчи матн сариқ фонда чиқади ва компонентанинг ишлатилиш

мақсадини қисқача ёритади. Ёрдамчи матн экранда пайдо бўлиши учун boolean туридаги ShowHint хусусиятининг қийматини true деб аниқлаш зарур.

TabOrder хусусияти контейнердаги компоненталарнинг фокус олиш тартибини аниқлайди (“Tab” тугмаси босилганда), яъни компоненталар "айланиб" чиқиш кетма – кетлигини аниқлайди. Келишув бўйича бу кетма – кетлик формани кўринишини яратишда компоненталарни контейнерга жойлаштириш тартибига мос келади: биринчи компонентанинг TabOrder хусусиятининг қиймати 0, иккинчисиники - 1 ва ҳоказо. Тартибни ўзгартириш учун компонентанинг TabOrder хусусиятига зарур қийматни бериш керак. Ҳар бир контейнер бошқаларига боғлиқ бўлмаган табуляция тартибига эга бўлади. Иккита компонента бир хил табуляция тартибига эга бўлиши мумкин эмас.

TabStop хусусияти TabOrder билан биргаликда ишлатилиб, компонентанинг фокус олиши ёки йўқлигини аниқлайди. Агар TabStop хусусиятининг қиймати true бўлса, компонента фокус олиши мумкин, акс ҳолда йўқ. Визуал компоненталарнинг табуляция тартибини Edit буйруқлар менюсида жойлашган EditTabOrder (Табуляция тартибини ўзгартириш) диалог дарчаси орқали амалга ошириш мумкин.

Амалий қисм.

Standard компоненталар палитрасидан фойдаланиб дастурлар тузамиз.

1 – мисол: Ҳароратни ифодаловчи дастур яратиш.

Ечиш: Ҳарорат асосан 2 хил ўлчовда ўлчанади. Бу икки ўлчовлар қуйидагилар ҳисобланади: цельсий ва фаренгейт. Ҳароратни цельсийда ўлчаб уни фаренгейтга қуйидаги формула бўйича ўтказамиз.

$$\text{Цельсийдан фаренгейтга ўтиш: } t_F = \frac{9}{5}t_C + 32;$$

$$\text{Фаренгейтдан цельсийга ўтиш: } t_C = \frac{5}{9}(t_F - 32).$$

Биз бу дастурни тузиш учун керак бўладиган компоненталар билан танишиб чиқамиз:

Ҳароратни киритиш учун Edit компонентаси;

Фойдаланувчиларга эслатма учун Label компонентаси;

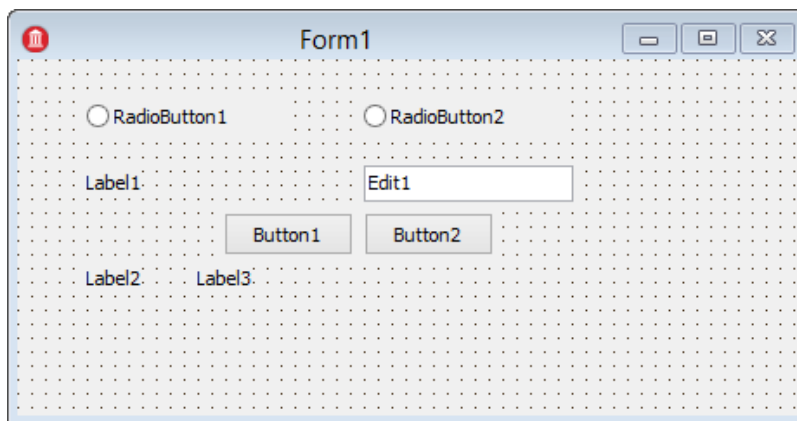
Киритилган ҳароратни тавдалантириш учун RadioButton компонентаси;

Ҳароратни биридан иккинчисига ўтиш учун Button компонентаси.

Юқорида келтирилган компоненталардан фойдаланиб форма яратимиз. Форма яратишдан олдин илова яратишни кўриб чиқайлик.

Биринчи, С++ Builder дастурни ишган туширамыз.

Иккинчи, File менюсининг New бандидаги С++ Builder Application бандини танлаймыз. Шунда илова, илованинг формаси ва формага тегишли бўлган модуллар яратилади. Формага керакли компоненталарни жойлаштирамыз (2.3.1 –расм).



2.3.1 – расм: Формага керакли компоненталарни жойлаштириш.

2.3.1 – расмдаги формани 2.3.2 – расмдаги кўринишга келтирамыз. Бунинг учун форманинг хусусиятларини ва формадаги компоненталарни хусусиятларини ўзгартирамыз.

Form1

Caption	Harorat konvertori...
Height	240
Width	357

RadioButton2

Caption	Farengoyt
Left	200
Top	24
Checked	False

Edit1

RadioButton1

Caption	Selsiy
Left	40
Top	24
Checked	True

Label1

Caption	Haroratni kiriting:
Left	40
Top	64

Button1

Text	0	Caption	Hisoblash
Left	154	Left	96
Top	61	Top	104

Button2

Caption	Tozalash
Left	177
Top	104

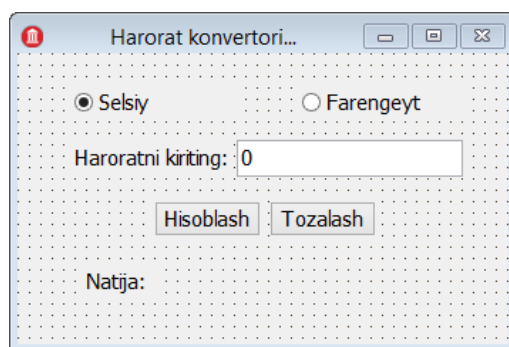
Label2

Caption	Natija:
Left	48
Top	151

Label3

Caption	<тозалаймиз>
Left	96
Top	151

Юқорида келтирилган ишларни бажарганимиздан сўнг 2.3.2 – расмдаги кўринишга келамиз.



2.3.2 – расм: Формадаги объектларни фойдаланувчига тушунарли кўриниши.

Формадаги Hisoblash тугмаси босилганида (OnClick) ходисасига листинг 2.3.1 ни ёзамиз. Бу листингни OnClick ходисасига ёзишимиз учун қуйидаги кетма – кетликни амалга оширамиз. Object Inspector ойнасидаги Events бўлимига ўтамиз, келтирилган ходисаларнинг ичидан OnClick ходисасини топамиз ва сичқончанинг чап тугмасини унинг устида 2 марта чертамиз ёки Hisoblash тугмаси устида сичқончанинг чап тугмасини унинг устида 2 марта чертамиз:

Листинг 2.3.1:

```

1: void __fastcall TForm1::Button1Click(TObject *Sender)
2: {
3:     float ts, tf;
4:     if (RadioButton1->Checked) {
5:         ts = StrToFloat(Edit1->Text);
6:         tf = 9.0/5 * ts + 32;
7:         Label3->Caption = FloatToStr(tf) + " F";
8:     }
9:     if (RadioButton2->Checked) {
10:        tf = StrToFloat(Edit1->Text);
11:        ts = 5.0/9 * (tf - 32);

```

```
12:         Label3->Caption = FloatToStr(ts) + " C";
13:     }
14: }
```

Формадаги Tozalash тугмаси босилганида (OnClick) ходисасига қуйидаги листинг ёзамиз:

Листинг 2.3.2:

```
1: void __fastcall TForm1::Button2Click(TObject *Sender)
2: {
3:     RadioButton1->Checked = true;
4:     Edit1->Text = "0";
5:     Label3->Caption = "";
6:     Button1->SetFocus();
7: }
```

Назорат саволлари:

1. Ходиса нима?
2. Компоненталар палитрасининг асосийларини санаб беринг.
3. Standard компоненталар палитраси компоненталарини тушунтириб беринг.
4. Align хусусиятининг қийматларини санаб беринг.
5. Cursor хусусиятининг қийматларини санаб беринг.

Амалий иш №3: Бир ва бир нечта форма билан ишлаш. Мулоқот ойналари ва уларнинг турлари

Ишдан мақсад: Ушбу амалий ишида талабалар бир ва бир нечта формалар билан ишлашни ўрганадилар.

Керакли жиҳозлар ва ускуналар: Компютер, ведио проектер

Назарий маълумот.

Ихтиёрий илованинг асосий элементи форма (контейнер) ҳисобланади. Формага бошқа кўринадиган ва кўринмайдиган компоненталарни жойлаштириш мумкин. Форма, фойдаланувчи нуқтаи назаридан, у илова билан ишлаётган дарчадир. Иловага киритилган ҳар бир янги форма ўзининг модули (init) га эга. Модулда формада бажариши керак бўлган функция, процедуралар келтирилади. Одатда мураккаб иловалар бир нечта формадан иборат бўлади. Янги илова (программа) тузиш учун File менюсидан NewApplication командаси танланади. Одатда биринчи форма асосий форма ҳисобланади. Иловага янги форма қўшиш учун File менюсидан NewForm танланади. Бир формадан бошқасига ўтиш учун Show ва ShowModal методларидан фойдаланиш мумкин. ShowModal методи жорий формани ёпгандан кейин бошқа формалар билан ишлашга рухсат беради. Бу метод оператив хотиралан унумли фойдаланиш учун ишлатилади. Show ва ShowModal методларини айни вақтда кўринмайдиган формалар учун ишлатиш мумкин. Агар формани актив ёки пассивлиги номаълум бўлса, у ҳолдақуйида келтирилган дастур коди ёзилади.

```
1: If (! Form2->Visible) Form2->ShowModal;
```

Show ва ShowModal методлари бажарилганда форманинг OnShow ходисаси содир бўлади. Hide методи орқали формани кўринмайдиган қилиш мумкин. Формани Close методи орқали ёпиш мумкин. Бир нечта форма билан ишлаганда бир формадан бошқасига ўтиш учун Shift+F12 тугмаси босилади. Форманинг баъзи хусусиятларини кўриб чиқамиз.

BorderIcons хусусиятига тегишли бўлган катталиклар қуйидагилардан иборат:

– biSystemMenu – Формада система тугмаларининг кўриниш ёки кўринмаслигини таъминлайди;

- biMinimize – Формадаги система тугмаларидан «свернуть» тугмасини актив ёки пассивлигини таъминлайди;
- biMaximize – Формадаги система тугмаларидан «развернуть» тугмасини актив ёки пассивлигини таъминлайди;
- biHelp– Формадаги «ёрдам» тугмасини актив ёки пассивлигини таъминлайди.

Форманинг WindowState хусусиятига тегишли бўлган катталиклар:

- wsMaximized – форма фаоллашаётганда тўлиқ экран ҳолатида очилишни таъминлайди;
- wsMinimized – форма фаоллашаётганда йиғилган («свернуть») ҳолатида очилишни таъминлайди;
- wsNormal – форма фаоллашаётганда одатий, яъни яратилгандаги ўлчамда очилишини таъминлайди.

Форманинг баъзи ходисаларини кўриб чиқамиз:

- OnActivate – ходисаси форма фаоллашган вақтда юз беради;
- OnClose – ходисаси форма ёпилаётганда юз беради;
- OnCreate – ходисаси форма яратилганда юз беради;
- OnResize – ходиса форманинг ўлчамлари ўзгарганда (катталашганда ёки кичиклашганда) юз беради;
- OnCloseQuery – форма ёпилишидан олдин савол билан мурожаат қилиш учун ишлатилади.

Мисол: Форма ёпилиши олдидан фойдаланувчига “Формани ёпишни хоҳлайсизми?” деб мурожаат этиш ва Yes тугмаси босилганда, формани ёпиш дастурини кўриб чиқамиз.

Ечилиши: Форманинг OnCloseQuery ходисасига листинг 3.1.1 даги дастурни ёзамиз.

```

1: void __fastcall TForm1::FormCloseQuery(TObject *Sender, bool
   &CanClose)
2: {
3:     if(MessageDlg("Chiqishni
   xohlaysizmi?", mtConfirmation, TMsgDlgButtons() << mbYes << mbNo, 0) ==
   mrYes)
4:         CanClose = true;
5:     else
6:         CanClose = false;
7: }
```

Дастур яратиш давомида турли хилдаги мулоқот ойналари билан ишлашга тўғри келади. Бу ойналарни турли хилдаги номлар билан аташ мумкин. Бу ойналар қуйидагилар:

- Тасдиқлаш ойнаси;
- Хатолик ойнаси;
- Огоҳлангитиш ойнаси;
- Ахборот ойнаси;
- Фойдаланувчи ойнаси.

Юқорида келтирилган мулоқот ойналарининг барчасини бири-биридан фарқи, ойналарнинг иконкасидадир. Мулоқот ойналарини ишга тушириш учун дастурлаш тилининг стандарт функцияларидан бири бўлган `MessageDlg` дан фойдаламиз. Функциянинг параметрлари билан танишамиз:

```
MessageDlg(string str, TMsgDlgType mdt, TMsgDlgButtons btn, int help);
```

Биринчи параметр сатрли типда бўлиб, бу параметрнинг ўрнида келган ўзгарувчи ёки ўзгармас катталиқнинг қиймати изоҳ сифатида экранда гавдаланади.

Иккинчи параметр мулоқот ойнасининг типи бўлиб, бу параметр қуйида келтирилган ўзгармасдан бирини қабул қилади:

- `mtConfirmation` (Тасдиқлаш ойнаси);
- `mtInformation` (Ахборот ойнаси);
- `mtWarning` (Огоҳлантириш ойнаси);
- `mtError` (Хатолик ойнаси);
- `mtCustom` (Фойдаланувчи ойнаси).

Учинчи параметр, бу мулоқот ойнасининг тугмаларидир. Бу тугмалар қуйидагилардир:

```
mbYes, mbNo, mbOK, mbCancel, mbAbort, mbRetry, mbIgnore, mbAll,  
mbNoToAll, mbYesToAll, mbHelp, mbClose
```

Windows операцион тизимида фойдаланиладиган файлларни сақлаш, файлларни очиш, маълумотларни чоп этиш мулоқот ойналари компоненталар палитрасининг алоҳида палитрасида тайёр компонента ҳолатида берилган. Бу палитра `Dialogs` номи билан номлангандир. Бу палитрадаги компоненталар дастур ишга тушганда формада кўринмайди. Бу палитрадаги баъзи бир компонентани кўриб чиқамиз:

- TOpenDialog – бу компонента очиш мулоқот ойнасини ишга туширади ва бу ойна билан ишлаш имконини яратади;
- TSaveDialog – бу компонента сақлаш мулоқот ойнасини ишга туширади ва бу ойна билан ишлаш имконини яратади;
- TFontDialog – бу компонента шрифт мулоқот ойнасини ишга туширади ва бу ойна билан ишлаш имконини яратади;
- TColorDialog – бу компонента ранглар мулоқот ойнасини ишга туширади ва бу ойна билан ишлаш имконини яратади;
- TPrintDialog – бу компонента чоп этиш мулоқот ойнасини ишга туширади ва бу ойна билан ишлаш имконини яратади;
- TFindDialog – бу компонента қидириш мулоқот ойнасини ишга туширади ва бу ойна билан ишлаш имконини яратади;
- TReplaceDialog – бу компонента ўрнига қўйиш мулоқот ойнасини ишга туширади ва бу ойна билан ишлаш имконини яратади.

Юқорида келтирилган компоненталарни чақириш учун, компонентанинг Execute методини ишга туширилади.

Амалий қисм.

Dialog компоненталар палитраси ёрдамида Windows операцион тизимининг стандарт дастурларидан бири бўлган Блокнот дастурининг кичик кўринишини яратамиз. Бу дастурни тузиш учун бизга керакли бўлган компоненталар рўйхатининг кўриб чиқамиз:

Бош менюни яратиш учун – TMainMenu;

Маълумотларни киритиш ва таҳрирлаш майдони учун – TMemo;

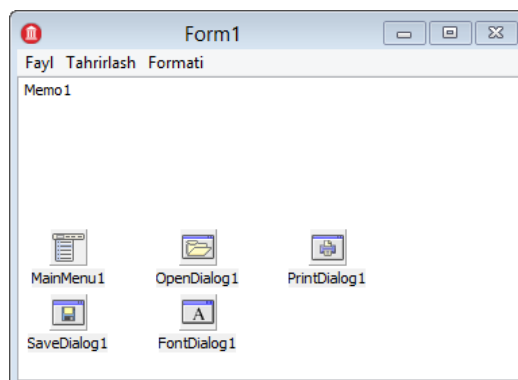
Киритилган маълумотларни сақлаш учун – TSaveDialog;

Олдин яратилган файлларни очиш учун – TOpenDialog;

Киритилган маълумотларни шрифтини ва унинг ўлчамини ўзгартириш учун – TFontDialog;

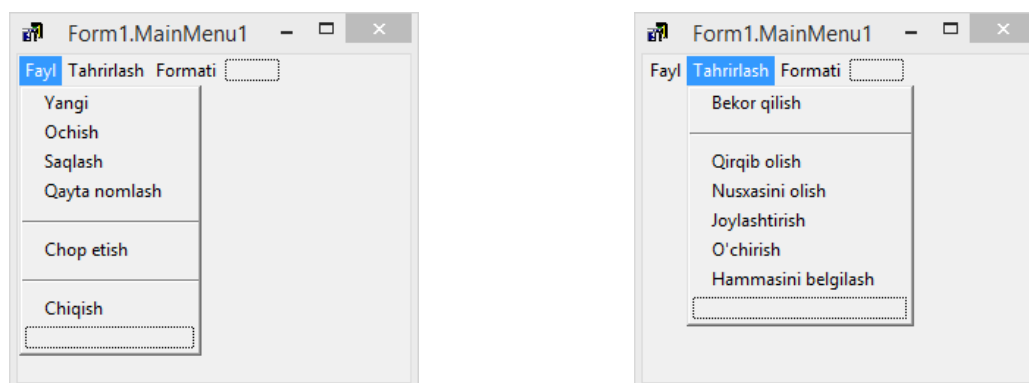
Ойнадаги маълумотларни чоп этиш учун – TPrintDialog.

Юқорида келтирилган компоненталарни формага 3.3.1 – расмда кўрсатилгандай жойлаштирамиз.



3.3.1 – расм: Формага керакли компоненталарни жойлаштириш.

Бош менюдаги менюларни яратиш учун сичқончанинг чапги тугмасини MainMenu1 компонентасининг устида икки марта босилади. Шундан сўнг ойна ҳосил бўлади, бу ойнада асосий менюнинг бандларини шакллантирилади. 3.3.2 – расмда бу бандларнинг кўриниши келтирилган.



3.3.2 – расм: Файл ва Таҳрирлаш менюларининг бандлари.

Блокнот дастурининг кодини ёзишда 3.3.2–расмдаги менюнинг бандларига асосланган ҳолда яратилади. Бу бандларни чертилганида ёки босилганида ходисаларига қуйидаги листингларда келтирилган кодларни ёзишимиз.

Листинг 3.3.1: Формат менюсидаги Сўзларни бўлиш бандининг коди.

```

1: void __fastcall TForm1::Sozlarnibolish1Click(TObject *Sender)
2: {
3:     Memo1->WordWrap = !Memo1->WordWrap;
4:     Sozlarnibolish1 -> Checked = Memo1->WordWrap;
5:     if (Memo1->WordWrap) Memo1->ScrollBars = ssVertical;
6:     else Memo1->ScrollBars = ssBoth;
7: }

```

Листинг 3.3.2: Формат менюсидаги Шрифт бандининг коди.

```

1: void __fastcall TForm1::Shrifti1Click(TObject *Sender)
2: {
3:     if (FontDialog1->Execute()) Memo1->Font = FontDialog1->Font;

```

4: }

Листинг 3.3.3: Тахрирлаш менюмининг бандлари.

```
1: void __fastcall TForm1::Bekorqilish1Click(TObject *Sender)
2: {
3:     // Bekor qilish
4:     Memo1->Undo();
5: }
6:
7: void __fastcall TForm1::Qirqibolish1Click(TObject *Sender)
8: {
9:     //Qirqib olish
10:    Memo1->CutToClipboard();
11: }
12:
13: void __fastcall TForm1::Nusxasiniolish1Click(TObject *Sender)
14: {
15:     //Nusxasini olish
16:     Memo1->CopyToClipboard();
17: }
18:
19: void __fastcall TForm1::Joylashtirish1Click(TObject *Sender)
20: {
21:     //Joylashtirish
22:     Memo1->PasteFromClipboard();
23: }
24:
25: void __fastcall TForm1::Hammasinibelgilash1Click(TObject *Sender)
26: {
27:     // Hammasini belgilash
28:     Memo1->SelectAll();
29: }
```

Назорат саволлари:

1. Форманинг WindowState хусусияти қандан ўзгармаслар қабул қилади ва уларнинг таснифи?
2. Форманинг BorderIcons хусусияти қандан ўзгармаслар қабул қилади ва уларнинг таснифи?
3. Мулоқот ойналарининг турлари ва унинг тугмалари?
4. Dialog компоненталар палитрасикомпоненталари?

Амалий иш №4: Additional компоненталар палитраси компоненталари ёрдамида дастурларни лойиҳалаш

Ишдан мақсад: Additional компоненталар палитрасини ўрганиш.

Керакли жиҳозлар ва ускуналар: Компютер, ведио проектер

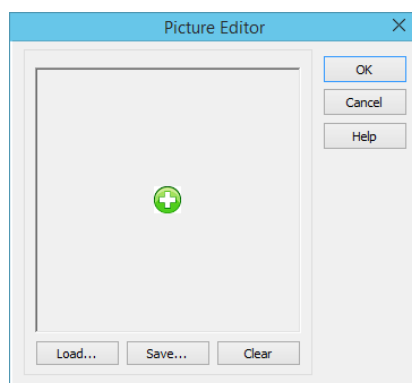
Назарий маълумот

C++ Builder дастурлаш муҳитида Standard компоненталар палитрасида кейинги ўринда турувчи компоненталар палитраси бу Additional компоненталар палитрасидир. Additional компоненталар палитрасидаги компоненталар 4.1.1 – расмда берилган.



4.1.1 – расм: Additional компоненталар палитраси.

Компоненталар палитрасидаги компоненталар билан танишиб чиқамиз:



4.1.2 – расм: Additional компоненталар палитраси.

TBitBtn – Расмли тугма. Биткўринишидагитасвир туширилган тугма яратади. Бундай тугмалар кўпроқ махсус диалогли ойналарда қўлланилади. Бу тугма яратиш учун TButton тугмасига янги хусусиятлар қўшилган. Масалан, Kind –тугманинг стандарт кўринишлари хусусияти, Glyph – битли тасвирларни чақириш хусусияти (4.1.2–расм).

TSpeedButton –Тезкор мурожаат тугмаси. Одатда бу тугма формада расмли тугма яратиш ва фақатгини сичқонча ёрдамида ишлаш учун яратилади.

TMaskEdit –Қолип бўйича маълумотларни киритиш учун бир қаторли тахрирлаш дарчасини яратади.

TStringGrid – Сатрларнинг икки ўлчамли жадвали. Ушбу компонентага тегишли барча хусусиятларнинг номлари ва вазибалари бўлиб, сиз улардан дастурни лойиҳалаш босқичида тўла фойдаланишингиз мумкин.

Символли кетма-кетликлар билан боғлиқ барча объектлар керакли объектга муружаат қилиш имконини берадиган Objects хусусиятида мужассам бўлган. Дастур бажарилиш пайтида символли кетма-кетликлар ва сетка устунининг улар билан боғлиқ объектлари Cols хусусияти билан адресланади. Rows хусусияти сетка сатрлари билан худди шундай иш тутиш имконини беради. Сетканинг барча символли кетма-кетликлари сетканинг керакли уясини адреслайдиган (манзиллайдиган) Cells хусусиятида мужассамдир.

TDrawGrid – Расм жойлаштириш мумкин бўлган жадвал. Тузилма ҳолига келтирилган график маълумотларни сатрлар ёки устунлар бўйича акс эттириш учун мунтазам сетка яратади. RowCount ва ColCount хусусиятлари вертикал бўйича ва горизонтал бўйича сетка уяларининг сонини белгилайди.

Options хусусиятининг қийматлари сетканинг тури (масалан, устунлар орасида ажратувчичизиқларга эга бўлган сетка тури) ва унинг хулқ–атворини (масалан, устундан устунга Tab клавишаси бўйлаб ўтиш) ўзгартириш имконини беради. Сеткадаги ажратиш чизиқларининг энг GridLineWidth хусусиятли томонидан белгиланади, айлангириш чизиқчалари эса ScrollBars хусусияти томонидан қўшилади. FixedCols ва FixedRows хусусиятлари устунлар ва сатрларнинг айлангирилишини таъқиқлаб қўйиш имконини беради, FixedColor хусусияти эса барча устун ва сатрларга маълум ранг беради.

TImage – График шакл жойлаштириладиган соҳа. Шаклда график тасвир контейнерини яратади (бу бит кўринишдаги файллар, пиктограмма ёки метафайллар бўлиши мумкин).

Тасвирлар файллари муҳаррирининг дарсаси Picture хусусияти қийматлари графасидаги тугма билан очилади. Контейнер ўз ўлчамларини тасвирни тўлиқ сиғдирадиган қилиб ўзгартириши учун, AutoSize хусусиятига true га ўрнатинг. Кичикроқ ўлчамдаги дастлабги

тасвир бутун формага чўзилиб кетиши учун, Stretch хусусиятига true қийматини ўрнатиш.

Тасвирлар файлларининг динамик юкланиши ва сақланиши учун, Picture объект хусусиятининг LoadFromFile ва SaveToFile услубларидан қуйида келтирилган дастур кодидан фойдаланиш:

- 1: Image->Picture->LoadFromFile(«<fayl nomi>»);
- 2: Image->Picture->SaveToFile(«<fayl nomi>»);

TShape – Стандарт геометрик шакллар. Айлана ва эллипс, квадрат ва тўғри тўртбурчак (бурчакларини юмалоқлаш мумкин) каби оддий геометрик шаклларнинг расмини чизади. Танлаб олинган геометрик шаклнинг турини Shape хусусияти, ранг ва бўяш усулини Brush компонентасига жойланган иккита Color ва Style хусусиятлари аниқлайди. Шаклларнинг ўлчамларини ҳам тегишли хусусиятлар аниқлайди.

TBevel – Бўрттирилган (ўйилган) шаффоф тўғри тўртбурчак соҳа. Худди искана билан ўйилгандек ҳажмли кўринадиган чизиқлар ёки рамкаларни яратади. Компонента чизаётган объектни Shape хусусияти аниқлайди, Style хусусиятининг қиймати эса объект кўринишини ўзгартириб, уни қавариқ ёки ботиқ ҳолга келтиради. Фойдаланувчи шакл ўлчамларини ўзгартирганда ҳам объектнинг нисбий ҳолатини ўзгармас қолдириш учун, Align хусусиятига true қийматини ўрнатади.

TScrollBar – Ҳаракатланиши мумкин бўлган дарча. Формада ўлчамлари ўзгарувчан контейнерни яратади, бу контейнер автоматик тарзда заруратга кўра айлантириш линейкалари билан таъминланади.

Айлантириб кўриш контейнери ёрдамида дарчанинг айрим соҳаларини айлантириб кўришдан ҳимоялаш мумкин. Масалан, компоненталар панели ва ҳолат панелини ҳимоялаш учун, аввал дарчани айлантириш линейкасини беркитиб қўйинг, кейин эса айлантириш контейнери ишчи соҳасида компоненталар панели ва ҳолат панели ўртасига жойлаштиринг. Контейнерни айлантириб кўриш линейкаси дарчага тегишли бўлиб кўринадиди, бироқ айлантириш фақат контейнер ичида амалга оширилади.

Айлантириб кўриш контейнерларидан яна бошқачасига ҳам фойдаланиш мумкин: улар бирон-бир дарчада кўплаб айлантириб

кўрилатган соҳалар (турлар) яратиш имконини беради. Турлар кўпинча тижорий матн процессорларида, бухгалтерия дастурларида ва лойиҳаларни режалаштириш дастурларида қўлланади. Айлантириб кўриш контейнери бошқа компонентларга, масалан Tbutton ва TCheckBoxга ҳам эга бўлиши мумкин.

TCheckBox – Танлашлар (белгилашлар) рўйхати;

TSplitter – Соҳани ажратувчи чизик;

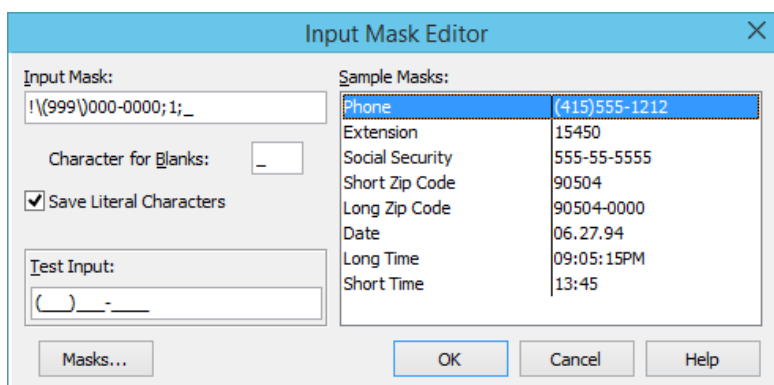
TStaticText – Статик (турғун) сатр;

TControlBar – Воситалар панели учун контейнер;

TApplicationEvents – Илованинг ходисалари;

TChart – Диаграммалар, график чизишнинг воситаси.

TMaskEdit – компонентаси бу юқорида айтиб ўтилганидек қолип бўйича киритиш дарчаси ҳисобланади. Ўзига хос форматдаги маълумотларнинг таҳрир қилинадиган назоратдаги тўртбурчак соҳасини яратади. Киритилаётган матннинг тўғрилиги рухсат этилган форматларни кодловчи ниқоб воситасида текширилади. Бу форматларга матн киритилган ва фойдаланувчига тақдим этилган бўлиши мумкин (сана, вақт, телефон рақами ва ҳ.к.). EditMask хусусияти жорий ниқоб кодини сақлайди. Ниқоблар муҳаррири дарчаси (4.2.1–расм) ушбу хусусият қийматлари графасида тугма билан очилади. TMaskEdit компонентаси TCustomMaskEdit синфининг тўғридан-тўғри ҳосиласидир. У сатрлар ёки устунлар бўйича белгили кетма-кетликларни акс эттириш учун мўлжалланган мунтазам (регуляр) тўрни яратади.



4.2.1 – расм: Ниқоблар муҳаррири дарчаси.

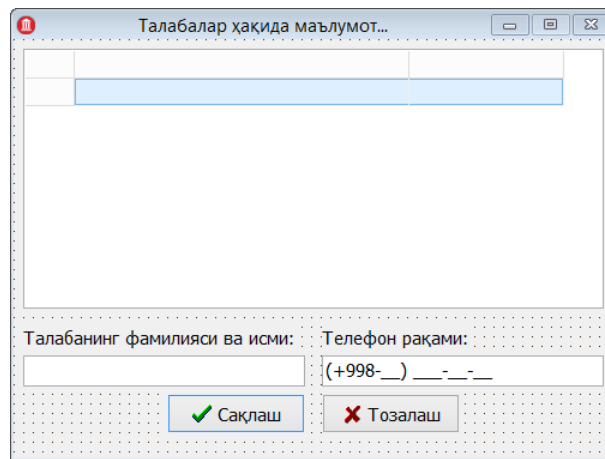
Ниқоблар яратишда қуйидаги белгилардан фойдаланилади:

! – ушбу белги шаблонда керакли бўлган белгиларнинг олдига қўйилади, агарда белги керак бўлмаса шаблондан кейин қўйилади;

> – ушбу белги киритилаётган белгиларни автоматик тарзда юқори регистрга ўтказади;
< – ушбу белги киритилаётган белгиларни автоматик тарзда қуйи регистрга ўтказади;
<> – белгиларни ўтказишни тўхтатиш;
\ – ушбу белги кейинги белгининг ўзи шаблонда акс эттирилади;
L – ушбу белгининг ўрнида харфлар киритилишини белгилайди;
l – ушбу белгининг ўрнида харфлар киритилишини белгилайди, лекин белгилар охиригача киритилиши шарт эмас;
A – ушбу белги харфлар ёки рақамлар киритилиши учун шаблон тайёрлайди;
a – ушбу белги харфлар ёки рақамлар киритилиши учун шаблон тайёрлайди ва яна бўш жой қолдириш имконини берилади;
C – ушбу белги ихтиёрий белгилар кетма–кетлигини киритишни таъминлайди;
0 – ушбу белги фақат рақам киритишни таъминлайди;
9 – ушбу белги фақат рақам киритишни ёки бўш қолдиришни таъминлайди;
– ушбу белги рақамлар, +, - белгилари ва бўш жой киритишни таъминлайди;
: – ушбу белги вақтнинг соат, дақиқа ва секундларни ажратиш учун ишлатилади;
_ – ушбу жойда автоматик тарзда бўш жой қолдиришни таъминлайди.

Мисол: Талабаларнинг фамилияси, исми ва талабанинг телефон рақамини киритиш ва маълумотларни сақлаш дастурни тузиш.

Ечиш: Талабаларни фамилияси ва исмини киритиш учун бизга TEdit компонентаси ва телефон рақамини киритиш учун TMaskEdit компоненталаридан фойдаланамиз. Бу икки компоненталардан ташқари TStringGrid компонентасидан фойдаланамиз. Маълумотларни жадвалга жойлаштириш ва киритиш майдонларини тозалаш учун иккита TBitBtn тугмаларидан фойдаланамиз. Юқорида келтирилган компоненталардан фойдаланиб 4.2.2 – расмдаги формани яратамиз.



4.2.2 – расм: Форманинг кўриниши.

Биринчи ўринда форманинг баъзи бир хусусиятларини ўзгартирамиз:

- `BorderIcons` хусусиятининг `biMaximize` ўзгарувчисининг қийматини `False` га ўзгартирамиз;
- `BorderStyle` хусусиятини `bsSingle` га ўзгартирамиз;
- `ClientHeight` хусусияти қийматини 350 га, `ClientWidth` хусусиятини 500 га ўзгартирамиз;
- `Caption` хусусияти қийматини “Талабалар ҳақида маълумот...” га ўзгартирамиз.

Иккинчи ўринда, `TStringGrid` компонентасининг айрим хусусиятларини ўзгартирамиз. Бу хусусиятлар қуйидагилар:

- `DefaultColWidth` хусусиятининг қийматини 40 га;
- `DefaultRowHeight` хусусиятининг қийматини 22 га;
- `Options` хусусияти `goRowSizing` ўзгарувчиси қийматини `True` га;
- `Options` хусусияти `goRowSelect` ўзгарувчиси қийматини `True` га;
- `TabStop` хусусиятининг қийматини `False` га ўзгартирамиз.

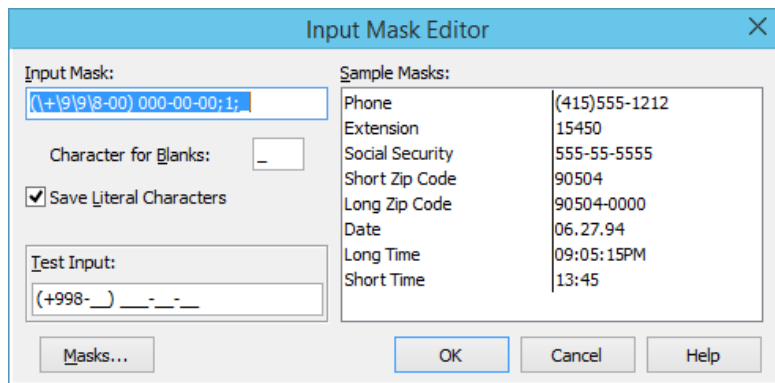
Формадаги жойлашган 2 та `TLabel` компонентасининг `Caption` хусусиятларини мос равишда қуйидаги ча ўзгартирамиз:

`Label1`: “Талабанинг фамилияси ва исми:”;

`Label2`: “Телефон рақами:”.

`Edit1` объектнинг `Text` хусусиятининг қийматини тозалаймиз.

`MaskEdit1` объектнинг `EditMask` хусусиятини ўзгартирамиз. Бунинг учун 4.2.3 – расмда келтирилган ойнадаги `Input Mask` бандини ойнада кўрсатилгандай ўзгартирамиз.



4.2.3 – расм: Input Mask Editor ойнаси.

4.2.2 – расмда берилган 2 та тугманинг Kind ва Caption хусусиятини мос равишда қуйидагича ўзгартирамиз:

BitBtn1 [Caption: “Сақлаш”, Kind: bkOK];

BitBtn2 [Caption: “Тозалаш”, Kind: bkAbort].

Юқорида келтирилган ўзгаришларни тугаганидан сўнг, форманинг OnActivate ходисасига листинг 4.2.1 да келтирилган дастур кодини ёзамиз:

Листинг 4.2.1: Форманинг OnActivate ходисаси.

```

1: void __fastcall TForm1::FormActivate(TObject *Sender)
2: {
3:     StringGrid1->Cells[1][0] = "Талабалар фамилияси ва исми";
4:     StringGrid1->Cells[2][0] = "Телефон раками";
5:     i = 0;
6: }

```

Формадаги Сақлаш тугмасининг босилиш, яъни OnClick ходисасига листинг 4.2.2 да келтирилган дастурнинг кодини ёзамиз:

Листинг 4.2.2: Сақлаш (BitBtn1) тугмасининг OnClick ходисаси

```

1: void __fastcall TForm1::BitBtn1Click(TObject *Sender)
2: {
3:     i++;
4:     StringGrid1->Cells[0][i] = IntToStr(i);
5:     StringGrid1->Cells[1][i] = Edit1->Text;
6:     StringGrid1->Cells[2][i] = MaskEdit1->Text;
7:     Edit1->Clear();
8:     MaskEdit1->Clear();
9:     if (i == StringGrid1->RowCount) {
10:         StringGrid1->RowCount = i + 1;
11:     }
12: }

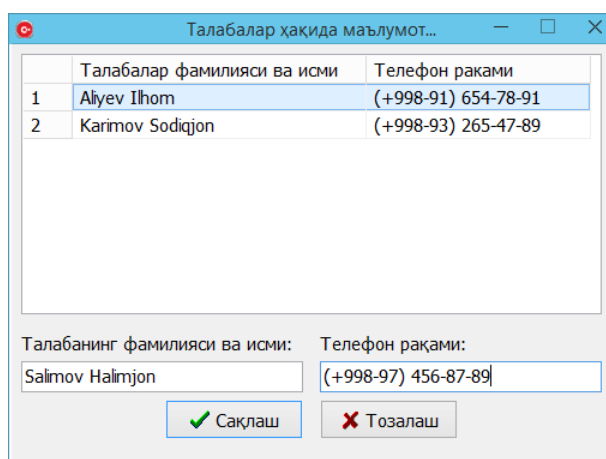
```

Формадаги Тозалаш тугмасининг босилиш, яъни OnClickходисасига листинг 4.2.3 да келтирилган дастурнинг кодини ёзамиз:

Листинг 4.2.3: Тозалаш (BitBtn2) тугмасининг OnClick ходисаси

```
1: void __fastcall TForm1::BitBtn2Click(TObject *Sender)
2: {
3:     Edit1->Clear();
4:     MaskEdit1->Clear();
5: }
```

Юқоридаги барча листингларни ёзилганидан сўнг, дастурни ишга туширамиз. Бунинг учун клавиатурадан F9 тугмасини босамиз. Шунда экранда 4.2.4–расмдаги ойна ҳосил бўлади. Бу ойнани иккита маълумот киритилган ҳолда кўрамиз.



	Талабалар фамилияси ва исми	Телефон рақами
1	Alyev Ithom	(+998-91) 654-78-91
2	Karimov Sodiqjon	(+998-93) 265-47-89

Талабанинг фамилияси ва исми: Salimov Halimjon

Телефон рақами: (+998-97) 456-87-89

✓ Сақлаш ✗ Тозалаш

4.2.4 – расм: Дастурни ишга туширилган ойнаси.

Амалий қисм.

TActionManager компонентаси Additional компоненталар палитрасида жойлашган бўлиб, фойдаланувчи буйруқларини ёзиш имконини яратувчи объект ҳисобланади. Бу компонентани ишлатиш учун қўшимча компонентлар мавжуд. Бу компоненталар қуйидагилардир:

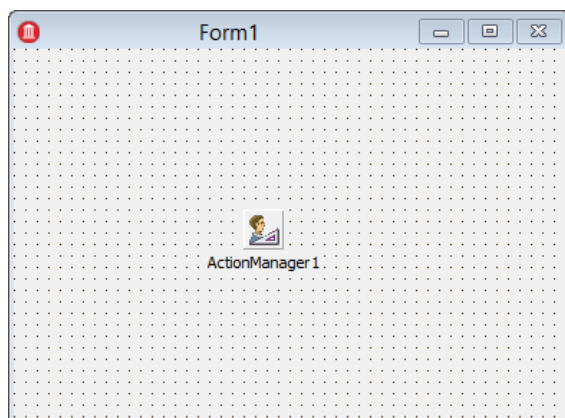
TActionMainMenuBar – TActionManager компонентаси ёрдамида яратилган буйруқлардан ташкил топган асосий меню яратиш имконини берувчи компонента ҳисобланади.

TPopupMenu – TActionManager компонентаси ёрдамида яратилган буйруқлардан ташкил топган сузувчи меню яратиш имконини берувчи компонента ҳисобланади.

TActionToolBar – TActionManager компонентаси ёрдамида яратилган буйруқлардан ташкил топган буйруқлар панели яратиш имконини берувчи компонента ҳисобланади.

TRibbon – TActionManager компонентаси ёрдамида яратилган буйруқлардан ташкил топган асосий меню яратиш имконини берувчи компонента ҳисобланади. Бу компонента Embarcadero компаниясининг RAD Studio 2010 ва ундан юқори версиялардан бошлаб фойдаланишга тақдим этилган.

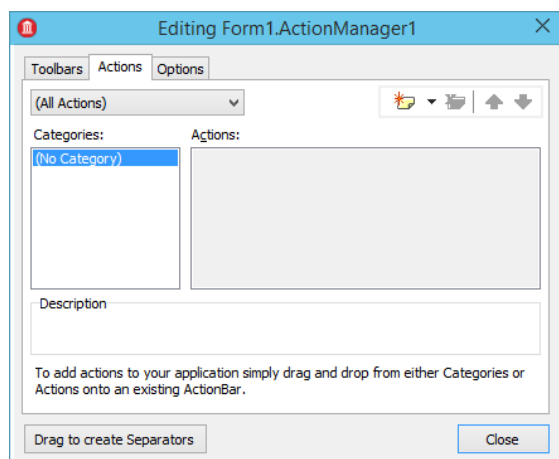
TActionManager компонентаси формага жойлаштиришни кўриб чиқамиз (4.3.1–расм).



4.3.1 – расм: TActionManagerкомпонентасининг формадаги кўриниши.

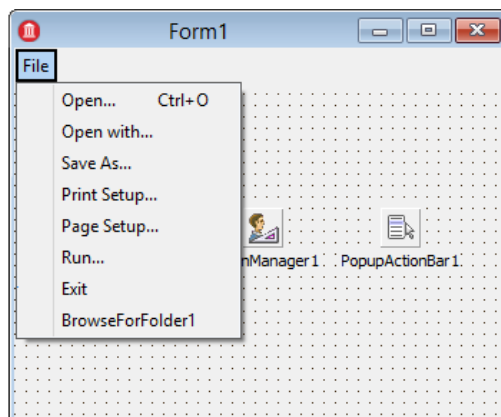
Формада жойлаштирилган компонента дастур ишга туширилганидан сўнг экранда кўринмайди. Бу компонента ёрдамида фойдаланувчи буйруқларини яратишни кўриб чиқамиз.

Компонента ёрдамида яратилган объект устида сичқончанинг чап тугмасини икки марта тезда чертамиз. Бунда 4.3.2–расмдаги ойна ҳосил бўлади. Бу ойна фойдаланувчи буйруқларини яратиш имконини яратади.



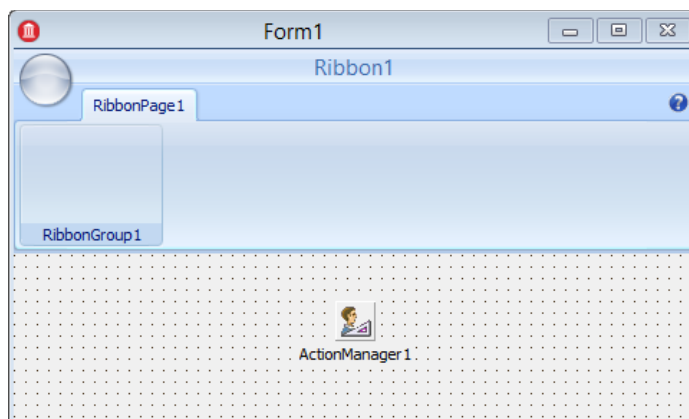
4.3.2 – расм: EditingActionManager ойнаси.

Фойдаланувчилар томонидан яратилган барча буйруқлар Actions бўлимида яратилади. Фойдаланувчи бу бўлимда ўзининг буйруқларини яратиши ёки стандарт буйруқлар фойдаланиши мумкин. Categories бўлими ёрдамида яратилган буйруқларни туркумларга бўлиш мумкин (4.3.3–расм).



4.3.3 – расм: TActionMainMenuBar компонентаси учун Categories бандидан фойдаланиб асосий меню яратиш.

RAD Studio 2010 ва ундан юқори версияларида ишлаб чиқилган TRibbon компонентасидан фойдаланишни кўриб чиқамиз. Бунинг учун формага TRibbon компонентаси ва TActionManager буйруқлар менежери компоненталарини жойлаштирамиз (4.3.4–расм).



4.3.3 – расм: TRibbon компонентаси ёрдамида вкладкали менюлар яратиш.

TRibbon компонентасининг ActionManager хусусияти ёрдамида TActionManager компонентасини бир–бирига боғлаймиз. Шундан сўнг буйруқлар менежеридаги буйруқлардан TRibbon компонентасида ва унинг асосий менюсидан фойдаланилса бўлади.

Назорат саволлари:

1. Additional компоненталар палитрасидаги компоненталарни санаб беринг.

2. TactionManager компонентаси, у билан ишлайдиган компоненталар ва уларнинг вазифаси.
3. TMaskEdit компонентасининг ниқобларда ишлатиладиган белгилар ва уларнинг вазифаси.
4. TButton, TBitBtn ва TSpeedButton компоненталарининг бир-биридан фарқини тушунтириб беринг.

Амалий иш №5: C++ Builder нинг график имкониятлари

Ишдан мақсад: C++ Builder нинг график имкониятларидан фойдаланиб формада чизмалар ҳосил қилиш.

Керакли жиҳозлар ва ускуналар: Компютер, ведио проектер

Назарий маълумот

C++ Builder муҳитида чизиш сирти – TCanvas класс объекти дастур билан ишлаш даврида тасвири маълумотларни ҳосил қилиш имконини яратади. Бу синф объекти сирт бўйича кўриш, график тасвирлар чизиш, расмларни ва сиртнинг бирор қисмини нусхалаш, ҳамда матнни чоп қилиш имконини берувчи хосса ва методларини ўз ичига олади.

Ҳар бир Canvas хоссасига эга компонента ўз навбатида қалам, мўйқалам ва шрифт объектларини ўз ичига олади ва мос равишда Pen, Brush ва Font хоссаларига эга бўлади. Canvas хоссасини ўз ичига олган компонента ва объектлар қуйидагилар: TForm, TPanel, TPaintBox, TImage

Canvas хоссасининг Pen объекти ўз навбатида қуйидаги хоссаларга бўлинади:

Canvas->Pen->Color – қаламнинг рангини ўзгартириш ёки қаламнинг рангини аниқлаш;

Canvas->Pen->Width – қаламнинг қалинлигини ўзгартириш ёки қаламнинг қалинлигини аниқлаш;

Canvas->Pen->Style – чизилаётган чизиқнинг тоифаси. Чизиқ тоифаси қуйидаги қийматларни қабул қилиши мумкин:

psSolid – узлуксиз чизиқ (келишув бўйича);

psDash – тире белгиларидан ҳосил бўлган чизиқ;

psDot – нуқталардан ташкил топган чизиқ;

psDashDot – нуқта ва тире кетма-кетликларидан иборат чизиқ;

psDashDotDot – тире ва нуқталар кетма-кетликларидан иборат чизиқ;

psClear – кўринмас чизиқ;

psInsideFrame – чизиш сиртини чегараловчи тўғри тўртбурчак ичидаги чизиқ.

Brush хоссаси геометрик шакллар ва уларнинг ички соҳасини бўйаш шаклини аниқлайди. У қуйидаги хоссаларга эга:

Canvas->Brush->Color – мўйқалам ранги;

Canvas->Brush->Style – мўйқаламтоифаси аниқлайди ва қуйидаги қийматларни қабул қилиши мумкин:

bsSolid – берилган ранг билан шакл юзасини тўлиқ бўйлади;

bsClear – шакл юзаси бўйялмайди;

bsHorizontal –шакл юзаси параллел чизиқлар билан тўлдирилади;

bsVertical –шакл юзаси вертикал чизиқлар билан тўлдирилади;

bsFDDiagonal – шакл юзаси юқоридан пастга қараган чизиқлар билан тўлдирилади;

bsBDiagonal –шакл юзаси пастдан юқорига қараган чизиқлар билан тўлдирилади;

bsCross –шакл юзаси тўр билан тўлдирилади;

bsDiagCross –шакл юзаси эгри чизиқлардан ҳосил бўлган тўр билан тўлдирилади.

Canvas объектнинг муҳим хоссаларидан бири бу нуқталар ҳосил қилиш хоссасидар. Бу хосса Canvas->Pixels[x][y] ном билан ёзилади, координата ёрдамида нуқтанинг рангини аниқлаш ёки унга бошқа ранг қийматини бериш мумкин.

График объект сиртида матнни ҳосил қилиш учун TextOut дан фойдаланилади. TextOut нинг ёзилиш формати қуйидагича:

1: Объект->Canvas->TextOut(x, y, Text);

Буерда, Объект – матн ҳосил бўлувчи объект номи; x, y – матн бошланувчи координата; Text – ҳосил бўлувчи белги катталиқдаги матн ёки сатрли ўзгарувчи.

Ҳосил бўлувчи матн белгилари Canvas объектига мувофиқ келувчи Font хусусияти орқали ифодаланади. Font хусусияти Tfont объектига тегишли бўлиб, 5.3.1-жадвалда белги характеристикалари ва қўлланилувчи услублари келтирилган.

5.3.1-жадвал

Хусусият

Аниқланиши

Name	Фойдаланилувчи шрифт. Қиймат сифатида шрифт номи ёзилади, масалан, ArialСyr
Size	пунктларда ифодаланилувчи шрифт ўлчами. Пункт-полиграфияда қўлланилувчи ўлчов бир-лиги бўлиб, у тахминан 1/72 дюйм ¹ га тенг
Style	белгини ёзиш усули, қуйидагича бўлиши мумкин: оддий, қалин, курсив, остига чизилган, устига чизилган. Булар қуйидаги константалар ёрдамида амалга оширилади: fsBold(қалин), fsItalic(курсив), fsUnderline(остига чизилган), fsStrikeOut(устига чизилган).Style бир нечта усулларни комбинация қилиши мумкин. Масалан, қалин курсив холатини ифодалаш: Объект->Canvas->Font->Style = TFontStyles() << fsBold << fsItalic
Color	Белги ранги. Қиймат сифатида TColor константа-ларидан фойдаланиш мумкин

Қуйидаги дастур қисми TextOut функциясини қўллаш учун мисол бўла олади:

```
1: Form1->Canvas->Brush->Color =Form1->Color;
2: Form1->Canvas->Font->Size=14;
3: Form1->Canvas->Font->Style=TFontStyles() << fsBold << fsItalic;
4: Form1->Canvas->TextOut(10,10, "Salom, Talabalar!");
```

Амалий қисм.

Тўғри чизиқ ва фигуралар ҳосил қилиш

Геометрик шакллар чизиш учун қуйида келтирилган функциялардан фойдаланиш мумкин:

MoveTo – чизишнинг бошланғич нуқтаси;

LineTo – тўғри чизиқ чизиш;

Ellipse – Айлана ва эллипс чизиш;

Rectangle – тўғри тўртбурчак чизиш;

RoundRect – бурчаклари юмалоқ (силлик) бўлган тўртбурчак;

Arc – айлана ёки эллипс ёйини чизиш;

Polygon – кўпбурчак чизиш;

¹ Дюйм – тахминан 2,54 см тенг

Pie – айлана ёки эллипсининг секторини чизиш. Юқорида келтирилган буйруқ ҳақида тўла маълумотни оламиз.

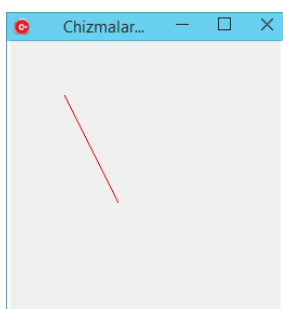
MoveTo(int X, int Y) – чизишнинг бошланғич нуқтасини белгилаш. Бу ерда (X,Y) – нуқтанинг координатаси.

LineTo(int X, int Y) – Тўғри чизиқ чизиш. Бу ерда (X,Y) – нуқтанинг координатаси.

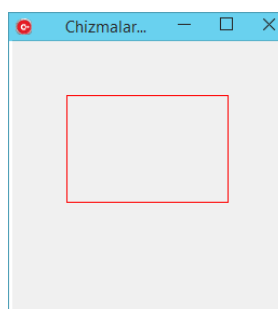
Мисол: Бошланғич координатаси (50,50) нуқтада, охириги нуқтаси координатаси (100,150) нуқтада бўлган қизил рангдаги тўғри чизиқ чизиш.

Форманинг OnPaint ходисасига қуйидаги келтирилган кодни ёзамиз. Натижа 5.4.1 – расмда келтирилган.

```
1: void __fastcall TForm1::FormPaint(TObject *Sender)
2: {
3:     Canvas->Pen->Color = clRed;
4:     Canvas->MoveTo(50,50);
5:     Canvas->LineTo(100,150);
6: }
```



5.4.1 – расм: Тўғри чизиқ чизиш.



5.4.2 – расм: Тўғри тўртбурчак

Rectangle(int X1, int Y1, int X2, int Y2) – диагналининг координаталари (X1,Y1) ва (X2,Y2) нуқталарда бўлган тўғри тўртбурчак чизади. Мисол сифатида координаталари (50,50) ва (200,150) нуқталарда бўлган тўғри тўртбурчак чизамиз.

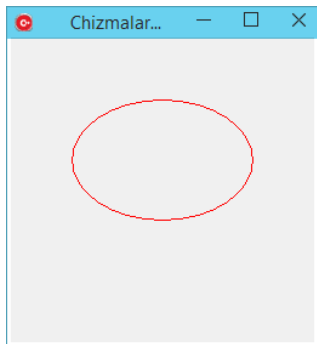
```
1: void __fastcall TForm1::FormPaint(TObject *Sender)
2: {
3:     Canvas->Pen->Color = clRed;
4:     Canvas->Rectangle(50,50,200,150);
5: }
```

Юқорида келтирилган коднинг натижаси 5.4.2 – расмда келтирилган.

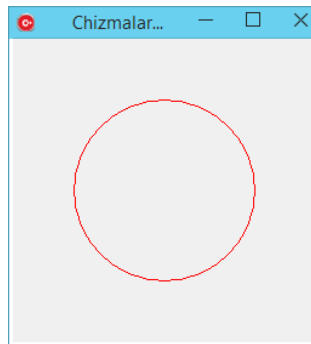
Ellipse(int X1, int Y1, int X2, int Y2) – диагналининг координаталари (X1,Y1) ва (X2,Y2) нуқталарда бўлган тўғри тўртбурчак чизига эллипс ёки айлана чизади. Мисол сифатида диагналининг

координаталари (50,50) ва (200,150) нуқталарда бўлган эллипс чизамиз. Натижа 5.4.3 – расмда келтирилган.

```
1: void __fastcall TForm1::FormPaint(TObject *Sender)
2: {
3:     Canvas->Pen->Color = clRed;
4:     Canvas->Ellipse(50,50,200,150);
5: }
```



5.4.3 – расм: Эллипс чизиш.



5.4.4 – расм: Айлана чизиш

5.4.4 – расмда келтирилган айлана чизиш учун қуйидаги кодни ёзамиз.

```
1: void __fastcall TForm1::FormPaint(TObject *Sender)
2: {
3:     Canvas->Pen->Color = clRed;
4:     Canvas->Ellipse(50,50,200,200);
5: }
```

Arc(int X1, int Y1, int X2, int Y2, int X3, int Y3, int X4, int Y4) – ёй чизиш. Бу ерда (X1,Y1) ва (X2,Y2) – мос равишда ёй чизиладиган тўғри тўртбурчак соҳанинг чап юқори ва ўнг паст учлари координатаси. (X3,Y3) ва (X4,Y4) нуқталар мос равишда ёйнинг бошланғич ва охириги координаталари.

```
1: void __fastcall TForm1::FormPaint(TObject *Sender)
2: {
3:     Canvas->Pen->Color = clRed;
4:     Canvas->Arc(50,50,200,200,150,70,70,150);
5: }
```

Юқорида келтирилган дастур коднинг натижаси 5.4.5 – расмда келтирилган.

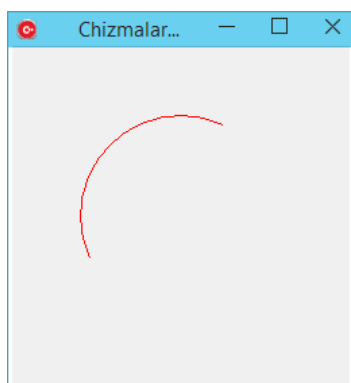
Polygon дан фойдаланиб кўпбурчак чизиш мумкин. Polygon TPoint типли массивни параметр сифатида қабул қилади. Ҳар бир массив элементи ўзида кўпбурчакнинг битта бурчаги координатаси (x,y) ни сақлайди. Polygon эса шу нуқталарни кетма-кет тўғри чизиқлар билан туташтириб чиқади. Чизиқнинг кўриниши (ранги, қалинлиги ва тури) Pen объекти билан ифодаланади.

```

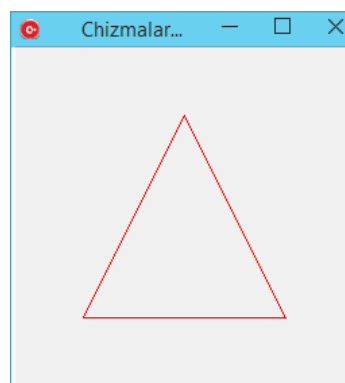
1: void __fastcall TForm1::FormPaint(TObject *Sender)
2: {
3:     TPoint p[3];
4:     p[0] = Point(125,50);
5:     p[1] = Point(50,200);
6:     p[2] = Point(200,200);
7:     Canvas->Pen->Color = clRed;
8:     Canvas->Polygon(p,2);
9: }

```

Юқорида келтирилган дастурнинг коди 5.4.6 – расмда келтирилган.



5.4.5 – расм: Айлана ёйи.



5.4.6 – расм: Учбурчак

Назарий саволлар:

1. Чизиш соҳаси ҳақида нималарни биласиз?
2. Чизиш соҳаси мавжуд бўлган коомпоненталарни санаб беринг?
3. Айлана ёки эллипс чизиш учун қайси буйруқдан фойдаланамиз?
4. Кўпбурчак тасвирини чизиш қайси буйруқ ёрдамида амалга оширилади?
5. Қаламнинг хусусиятларини ўзгартириш қандай амалга оширилади?
6. Мўйқалам хусусиятларини санаб беринг.

Амалий иш №6: DataAccess, DataControls, dbGo ёки ADO компоненталари палитраси

Ишдан мақсад: DataAccess, DataControls, dbGo ёки ADO компоненталар палитраси билан ишлаш.

Керакли жиҳозлар ва ускуналар: Компютер, ведио проектер

Назарий маълумот

Маълумотлар омбори билан ишлаш учун DataAccess компоненталар палитрасидаги компоненталар билан ишлашни ва уларнинг хусусиятларини ўрганишимиз лозим. Бу компоненталар палитрасидаги компоненталар қуйидагилардан иборат.



DataSource – маълумотларни ёки компоненталарни бир–бири билан боғлаш.



Table – маълумотлар омборини (файлни) боғлаш.



Query – маълумотлар омборидаги ёзувларни бошқариш.



StoredProc – маълумотлар омборидаги процедуралар ва функцияларни боғлаш.



Database – ягона маълумотлар омборини боғлаш.

TDataSource – компонентаси ёрдамида маълумотлар омборидаги маълумотларни маълумотлар базаси билан ишлай оладиган компоненталарга боғлашни таъминлайди. Бу компонентанинг хусусиятлари ва ходисалари билан танишиб чиқамиз:

6.1.1 – жадвал: TDataSource компонентасининг хусусиятлари ва ходисалари

AutoEdit	– автоматик равишда таҳрирлашни ёқиш ёки ўчириш
DataSet	– маълумотлар тўплами билан боғлаш
Enabled	– компонентани фаоллигини белгилаш
Name	– компонентанинг номи
OnDataChange	– маълумотлар танланганда
OnStateChange	– ҳолатлардан бири танланганида
OnUpdateChange	– маълумотларни таҳрирланганда

TTable – компонентаси ёрдамида маълумотлар омборидаги жадвални яратилаётган дастурга боғлашни таъминлайди. Бу компонентанинг хусусиятлари ва ходисалари баъзилари билан танишиб чиқамиз:

6.1.2 – жадвал: TTable компонентасининг хусусиятлари ва ходисалари

Active	– компонентани фаоллаштириш
--------	-----------------------------

AutoCalcFields	– автоматик ҳисоблаш майдонларини фаоллаштириш
AutoRefresh	– автоматик қайтаюклашни фаоллаштириш
DatabaseName	– маълумотлар омбори номи
Filter	– филтър кўриниши
Filtered	– филтърни фаоллаштириш
MasterSource	– асосий жадвални кўрсатиш
MasterFields	– асосий жадвални майдонини кўрсатиш
Name	– Компонентанинг номи
TableName	– маълумотлар омборидаги жадвалнинг номи

TQuery – компонентаси ёрдамида маълумотлар омборидаги жадвални яратилаётган дастурга сўровлар ёрдамида боғлашни таъминлайди. Бу компонентанинг хусусиятлари ва ходисалари TTable компонентасининг хусусиятлари билан бир хил ва қуйидаги хусусиятлар қўшилган:

6.1.3 – жадвал: TQuery компонентасининг хусусиятлари ва ходисалари

DataSource	– маълумотлар базасини кўрсатиш
Params	– параметрлар рўйхати
SQL	– SQL сўровларини ёзиш

TStoredProc – компонентаси маълумотлар базасидаги тайёр яратилган процедуралар ва функцияларни дастурга боғлашни таъминлайди.

TDatabase – ягона маълумотлар омборини боғлаш учун ишлатилади. Бу компонента ёрдамида маълумотлар омбори боғланган вақтда юқорида келтирилган компоненталар ёрдамида маълумотлар омборига эмас, балки шу компонентага боғланиш кўрсатилади. Бу компонентанинг баъзи бир хусусиятлари ва ходисалари билан танишиб чиқамиз:

6.1.4 – жадвал: TDatabase компонентасининг хусусиятлари ва ходисалари

AliasName	– маълумотлар омборини махсус номини кўрсатиш
Connected	– маълумотлар омборига уланишни фаоллаштириш
DatabaseName	– маълумотлар омбори номини кўрсатиш
DriverName	– маълумотлар омбори типини кўрсатиш
LoginPrompt	– базага боғланганда кириш ҳуқуқини белгилаш

Юқорида келтирилган компоненталарнинг ёрдамида маълумотлар омборига боғланиш амалга оширилади. Бу компоненталар C++Builderнинг янги версияларида бошқа ном билан аталган ёки баъзилари бошқа палитраларга ўтган бўлиши мумкин.

DataControls компоненталар палитраси ёрдамида маълумотлар омборидаги маълумотларни тахрирлаш, янги маълумот қўшиш, ўчириш ишлари амалга оширилади. Бу компоненталар палитрасидаги компоненталар билан танишиб чиқамиз.



DBGrid – маълумотлар омборидаги маълумотларни жадвал кўринишда гавдалантиради;



DBNavigator – маълумотлар омборидаги ёзувларни устида турли амалларни бажаришни таъминлайди. Бу компонентада янги ёзув қўшиш, маълумотларни ўзгатириш, ўчириш ва ҳоказо ишларни бажариш мумкин.



DBText – маълумотлар омборидаги матнли майдон маълумотини гавдалантирувчи компонента ҳисобланади.



DBEdit – маълумотлар омборидаги бирор майдонни тахрирловчи компонента



DBMemo – маълумотлар омборидаги матнли майдонларни тахрирлаш компонентаси.



DBImage – маълумотлар омборида жойлашган тасвирларни гавдалантириш компонентаси.



DBListBox – маълумотлар омборидаги маълумотларни рўйхатсимон кўринишда чиқарувчи компонента.



DBComboBox – маълумотларни комбинацияли танлаш компонентаси.













DBRichEdit – маълумотлар омборидаги матнли майдонларни кенгайтирилган кўринишда тахрирлаш компонентаси.

Юқорида келтирилган компоненталарнинг номларига эътибор берадиган бўлсак, бу компоненталарнинг номлари олдида DB олд қўшимчаси мавжудлигини кўрамыз. Бу олд қўшимча DataBase сўзининг қисқартирилган шакли бўлиб, маълумотлар базаси деган номни англатади. DB олд қўшимчали компоненталар маълумотлар омборидаги маълумотлар билан ишлаш маъносини англатади. Бу компоненталарнинг хусусиятлари ва ходисалари билан танишиб чиқамиз.

TDBGrid – компонентаси маълумотлар омборидаги маълумотларни жадвал кўринишда фойдаланувчига тушунарли қилиб, алоҳида майдонлар кўринишида кўрсатиб беради. Бу компонентанинг хусусиятлари ёрдамида фойдаланувчи ўзига хос

бўлган кўринишни яратиши мумкин. Масалан, жадвалнинг бирор – бир ёзув танланганида майдонларини алоҳида кўринишда кўриш ёки қаторни бир марта белгилаш кўриниши, майдонлар орасидаги устунларни ажратувчи чизиқнинг кўриниш ёки кўринмаслиги, бирта ёки бир нечта ёзувларни танлаш ҳуқуқи ва ҳоказо.

TDBNavigator – компонентаси маълумотлар омборидаги маълумотлар устида амаллар бажаришга ёрдам беради. Бу компонента тайёр буйруқлардан ташкил топган. Бу компонентанинг буйруқлар тўплами қуйидагилардан ташкил топган:

	nbFirst	Жадвалдаги биринчи маълумотга ўтиш
	nbPrior	Фаол ёзувдан олдинги ёзувга ўтиш
	nbNext	Фаол ёзувдан кейинги ёзувга ўтиш
	nbLast	Жадвалдаги охириги маълумотга ўтиш
	nbInsert	Жадвалга маълумот қўшиш
	nbDelete	Фаол ёзувни ўчириш
	nbEdit	Фаол ёзувни тахрирлаш
	nbPost	Фаол ёзувни сақлаш
	nbCancel	Ўзгартиришларни бекор қилиш
	nbRefresh	Жадвални қайта юклаш

Юқорида келтирилган тугмалардан кераксизларини хусусият ёрдамида кўринмаслик кўринишига келтириш мумкин. Бу хусусият VisibleButtons хусусияти ҳисобланади. Юқорида келтирилган тугмаларнинг ҳар бирига изоҳ бериш ҳам мумкин, бунинг учун Hints хусусиятини ўзгартириш кифоя. Бу компонентани маълумотлар омборига боғлаш учун DataSource хусусиятини TDataSource компонентасига боғлаш зарур бўлади. Шунда компонента маълумотлар омборига боғланиб, уни бошқариш амалга оширилади.

TDBText – компонентаси маълумотлар омборининг жадвалидаги майдонга боғланиб, ундаги ёзувчи фойдаланувчига тахрирлаш мумкин бўлмаган ҳолда кўрсатиб беради. Бу компонента TCustomLabel классининг ҳосиласидир. Бу компонентанинг маълумотлар омбори билан ишлайдиган TLabel компонентаси деб ҳисоблаш мумкин.

TDBEdit – компонентаси маълумотлар омборидаги маълумотларни ҳар бир майдонини фойдаланувчига тушунарли ҳолда дастурга улайди. Бу компонента ёрдамида маълумотлар омборидаги жадвал майдонини тахрирлаш ва ўзгартириш ёки маълумотлар киритиш мумкин.

C++Builder дастурлаш тилини маълумотлар омбори билан боғлаш турлари кўп бўлиб, биз ADO компоненталар тўпламидир. Бу компоненталар палитрасида қуйидаги компоненталар жойлашган:

TADOConnection	бу компонента маълумотлар омборини C++ Builder дастурлаш тили билан боғлайди.
TADOTable	бу компонента маълумотлар омборидаги жадвални C++ Builder дастур билан боғлайди.
TADOQuery	бу компонента маълумотлар омборидаги жадвалга сўровлар ёрдамида боғлайди
TADODataset	бу компонента маълумотлар омборига сўров, жадвал ёки процедурар шаклида уланиш.
TADOCommand	бу компонента маълумотлар омборига сўровлар ёрдамида боғланиш.
TADOStoredProc	бу компонента ёрдамида маълумотлар омборидаги процедураларни дастурга боғлаш учун хизмат қилади.
TRDSCONNECTION	бу компонента ёрдамида тармоқдаги маълумотлар омборига боғланишни амалга оширади.

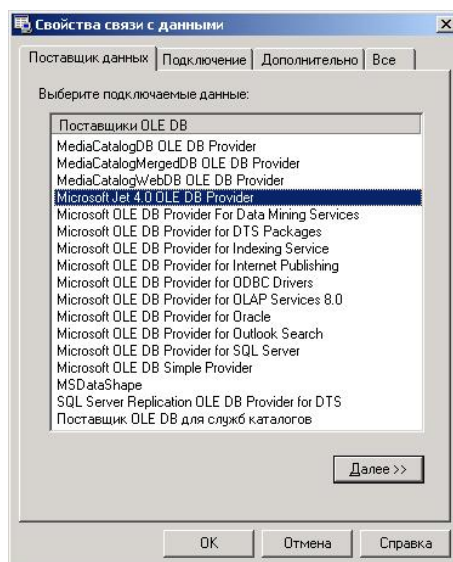
Амалий қисм.

Маълумотлар омборини дастурга боғлаш учун C++ Builder дастурлаш тили ёрдамида яратилаётган илованинг формасига ADOConnection компонентасини ўрнатилади. Компонентани танлаб Object Inspector ойнасидаги Properties (Хусусият) бўлимидаги ConnectionString бандини танланади. Бу бандаги(...)тугмани чертилади. Шунда қуйидаги ойна ҳосил бўлади.(6.3.1 – расм)



6.3.1 – расм. Базани C++Builder дастурлаш тили билан боғлаш ойнаси.

Бу ойнадаги Build... тугмасини чертилади. Шунда қуйидаги ойна ҳосил бўлади. (6.3.2 – расм)



6.3.2 – расм. Базани C++Builder дастурлаш тили билан боғлаш учун базани кўрсатиш ойнаси.

Бу ойнадаги Подключение ойнасига ўтиб, базани қаерда турганини кўрсатилади ва ОК тугмасини босилади.

Шундан сўнг ADOTable компонентасини фаоллаштирилади. Унинг хусусиятлар ойнасидан Connection хусусиятини танлаб компонентасини ADOConnection компонентаси билан боғланилади. DataSource компонентаси хусусиятлар ойнасидан DataSet хусусиятини танланиладива ADOTable компонентасини бир – бирига боғланилади.

Назарий саволлар:

1. DataAccess компоненталар палитрасидаги компоненталарни санаб беринг?
2. ADO компоненталари палитрасидаги компоненталарнинг вазифаларини нималардан иборат?
3. Маълумотлар омборидаги жадвалнинг майдонларини қандай компоненталар билан боғлаш мумкин?
4. DataSource қандай компонента ҳисобланади ва унинг дастурдаги вазифаси нималардан иборат?
5. Тармоқдаги маълумотлар омбори билан боғланишда қандай компонентанинг хизматидан фойдаланилади?

Амалий иш №7: Маълумотлар базаси билан ишлаш

Ишдан мақсад: Ушбу амалий ишида талабалар маълумотлар баъзаси билан дастурни боғлашни ўрганадилар.

Керакли жиҳозлар ва ускуналар: Компютер, ведио проектер

Назарий маълумот

Маълумотлар омбори асосан Microsoft фирмасининг Office пакетида жойлашган Access дастури, SQL Server дастури, MySQL маълумотлар омбори ёрдамида яратилади. Қуйида маълумотлар омборини яратиш учун Access дастуридан фойдаланилади.

Access дастури ёрдамида маълумотлар омбори яратишда авваламбор Access дастури ишга туширилади. Бунинг учун Пуск менюсидан Microsoft Office пакетини топилади. Бу пакетнинг ичидаги Microsoft Access дастурини ишга туширилади. Қуйида яратилаётган маълумотлар омбори Microsoft Office Access 2007 да яратилган. Шундан сўнг янги маълумотлар омбори яратилади. Янги маълумотлар омборини яратиш учун Новая база данных тугмаси босилади. Янги маълумотлар омборининг номи киритилганидан сўнг Создать тугмаси босилади. Янги маълумотлар омбори учун жадвал яратилади. Бунинг учун Создание вкладкасида Конструктор таблиц тугмаси босилади. Ҳосил бўлган майдонлар тўлдирилади. Масалан, яратилаётган жадвалда талабалар ҳақидаги маълумотлар акс эттирилган бўлсин. Талабалар жадвали учун қуйида тахминий майдонлар келтирилган.

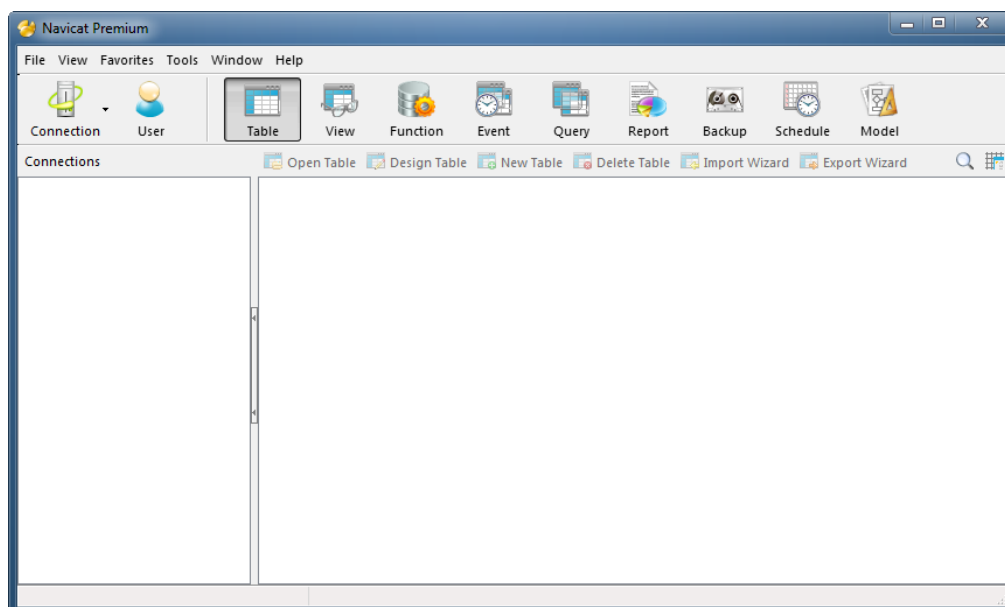
- surname – талабанинг фамилияси;
- given_name – талабанинг исми;
- father_name – талабанинг отасининг исми;
- birthdate – талабанинг туғилган вақти;
- telephone – талабанинг телефон рақами;
- address – талабанинг яшаш манзили.

Бу майдонлардан фойдаланиб жадвал кўриниши 7.1.1 – расмда келтирилган.

Имя поля	Тип данных	Описание
id	Счетчик	
surname	Текстовый	
given_name	Текстовый	
father_name	Текстовый	
birthdate	Дата/время	
telephone	Текстовый	
address	Текстовый	

7.1.1–расм: Student номли жадвалнинг MSAccess 2007 да кўриниши

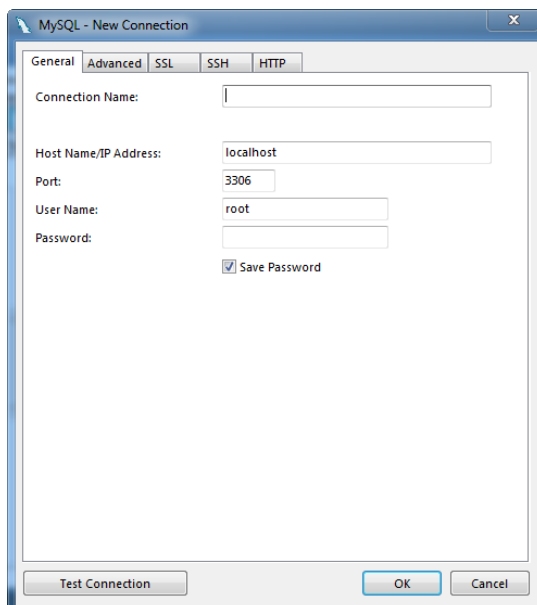
MySQL маълумотлар омбори билан ишлаш учун MySQLServer маълумотлар омборини ўрнатилади. Бу маълумотлар омбори билан ишлаш учун Navicat Premium дастуридан фойдаланилади. Бу дастур MySQL маълумотлар омбори билан ишлашда фойдаланувчи интерфейси ҳамда, SQL сўровларни билмаган фойдаланувчилар учун қулайликлар яратади. Navicat Premium дастурининг асосий ойнаси 7.1.2–расмда келтирилган.



7.1.2–расм: Navicat Premium дастурининг асосий ойнаси

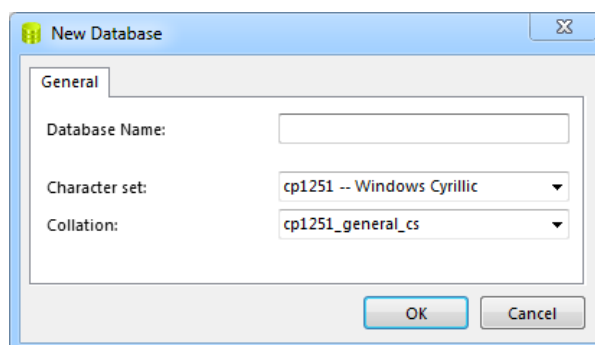
Navicat Premium дастурини MySQL дастурига улаш учун File менюсидан New connection бандидан MySQL бандини танланади. Бу буйруқлар кетма–кетлик билан бажарилганда MySQL маълумотлар омборига улаш ойнаси ҳосил бўлади. Бу ойна 7.1.3–расмда келтирилган. Бу ойнадаги Connection name майдонига маълумотлар омборини боғлашдаги номни кўрсатилади. Масалан, mysql_connect. Қолган майдонлар маълумотлар омбори жойлашган жой ёки компьютер номи кўрсатилади. Агар маълумотлар омбори локал компьютерда жойлашган бўлса, localhost сўзини Host name/ IP address майдонига киритилади, агарда маълумотлар омбори тармоқ

компьютерда бўлса, у ҳолда тармоқдаги компьютернинг IP манзили киритилади. Маълумотлар омбори ўрнатилаётгандаги портнинг рақами Port майдонига киритилади. Маълумотлар омборига кириш ҳуқуқини берувчи User name ва Password лар шу ном билан аталган майдонларга киритилади.



7.1.3–расм: Navicat Premium дастурини MySQL МБга улаш

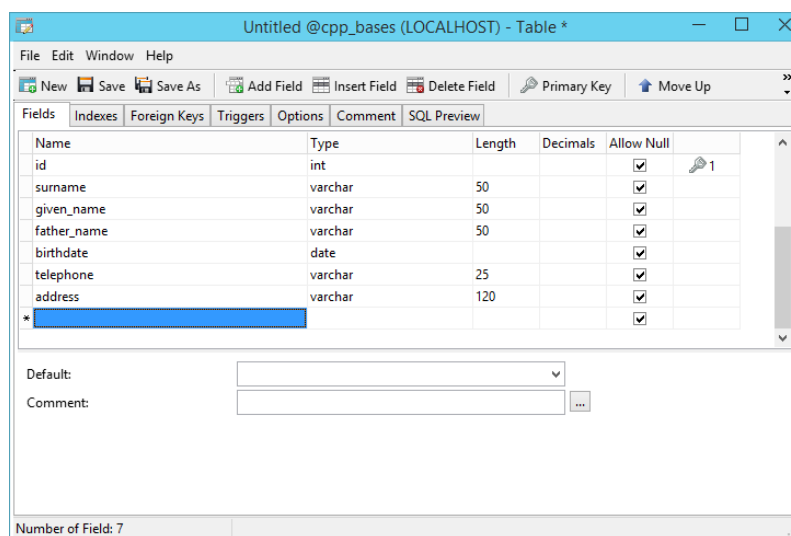
Юқорида келтирилган маълумотлар киритилганидан сўнг, 7.1.3 – расмдаги Test Connection тугмаси орқали уланиш текширилади. Агарда уланиш тўғри бўлса, ОК тугмаси босилади. Navicat Premium дастури ёрдамида янги маълумотларлар базасини яратиш учун ойнанинг ўнг томонидаги mysql_connect нинг устида сичқончанинг ўнг томонини чертилади. Ҳосил бўлган менюдан New Database... бандини танлаймиз. Шунда 7.1.4–расмда келтирилган ойна ҳосил бўлади.



7.1.4–расм: New Database ойнаси

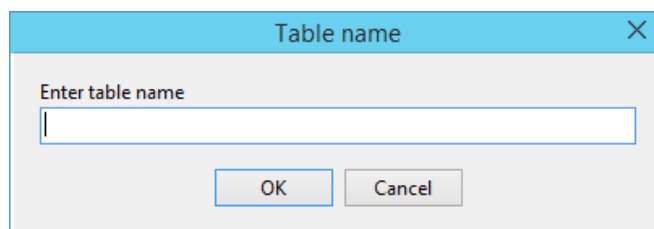
Database name майдонига янги яратилаётган маълумотлар омборининг номи киритилади. Маълумотлар омборининг номи майдонга киритилганидан сўнг ОК тугмаси босилади. Масалан, маълумотлар омборининг номи crr_bases номланг. Шунда

маълумотлар омбори яратилади. Бу маълумотлар омборининг ичига жадвал қўшиш учун биринчи навбатда маълумотлар омборини фаоллаштирилади. Бунинг учун маълумотлар омборининг номининг устида сичқончанинг чап тугмасини икки марта босиш билан амалга оширилади. Маълумотлар омбори очилганидан сўнг ундаги Tables бандини танланади ва унинг устида сичқончанинг ўнг тугмасини босамиз. Ҳосил бўлган менюдан New Table бандини танлаймиз. Ҳосил бўлган ойнада берилган жадвални 7.1.5–расмда берилган каби тўлдирилади.



7.1.5–расм: Янги жадвалнинг майдонларини яратиш ойнаси

Жадвалнинг майдонларини киритиб бўлганидан сўнг, жадвални сақлаш учун ойнадаги File менюсидан Save бандини танланади. Бу банд жадвални сақлаш учун ишлатилади. Агар жадвал олдин сақланмаган бўлса, у ҳолда дастури биринчи навбатда жадвални сақлаш учун ойна ҳосил қилади, бу ойнада жадвалнинг номи киритилиши сўралади (7.1.6–расм).



7.1.6–расм: Яратилаётган жадвалнинг номи киритиш ойнаси

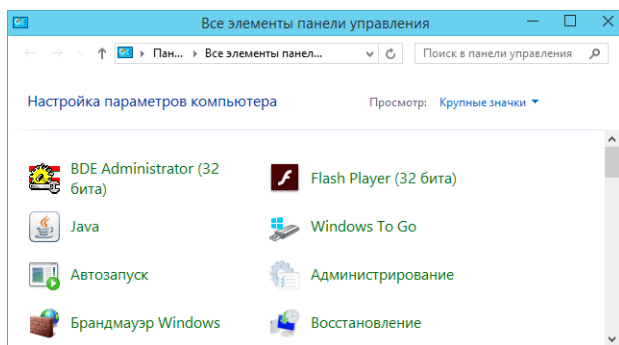
Бу ойнага жадвалнинг номи киритилади. Масалан, student номи билан сақланади.

Амалий иш.

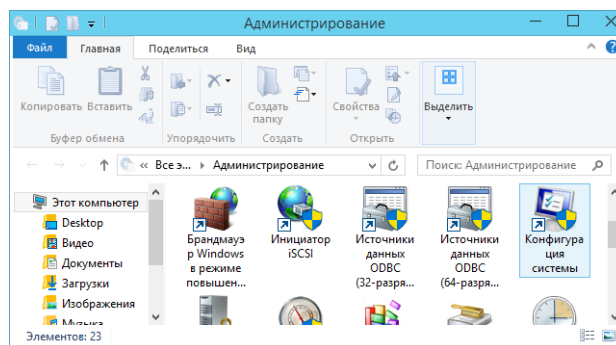
Олдинги маърузада MS Access да яратилган маълумотлар омборини яратилаётган иловага улаш кўрсатиб ўтилган эди. Энди MySQL маълумотлар базасини бошқариш тизимида яратилган маълумотлар омборини фойдаланувчи томонидан яратилаётган иловага улашни тушинтирилади. MySQL маълумотлар базасини бошқариш тизимида яратилган маълумотлар омборини фойдаланувчи томонидан яратилаётган иловага улаш учун MySQL Connection қўшимча дастури керак бўлади. Бу дастур ёрдамида MySQL МББТ яратилган МБ дастурга осонгина уланади.

MySQL МББТни иловага боғлашнинг 2 хил усули мавжуд. Биринчи усулида фойдаланувчи томонидан яратилган дастурнинг ўзи маълумотлар базасига боғланиб ишлайди. Бу вақтда дастур яратилган компьютернинг ўзида ишлайди, лекин бошқа компьютерга олинганда бу дастур ишлаши ноаниқ бўлади. Иккинчи усул қуйидагича амалга оширилади.

– Компьютернинг Панель управления ойнасига кирилади (7.2.1–расм);



7.2.1–расм: Панель управления ойнаси

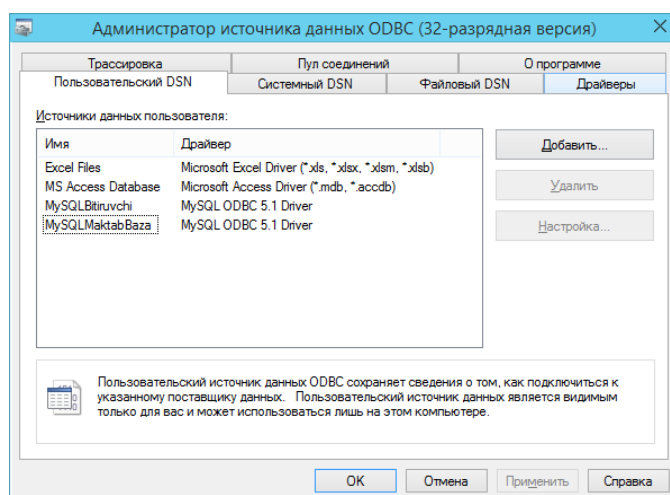


7.2.2–расм: Администрирование ойнаси

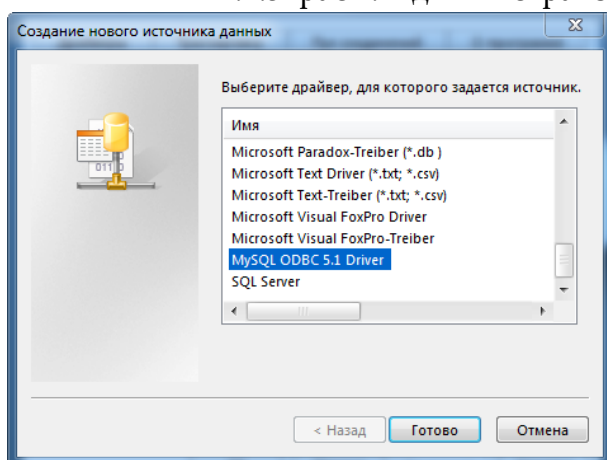
– 7.2.1–расмда кўрсатилган Панель управления ойнасидаги Администрирование бўлимини танланади. Шунда 7.2.2–расмдаги ойна ҳосил бўлади;

– 7.2.2–расмда кўрсатилган ойнадан Источники данных ODBC банди танланади, шунда 7.2.3–расмда кўрсатилган ойна ҳосил бўлади;

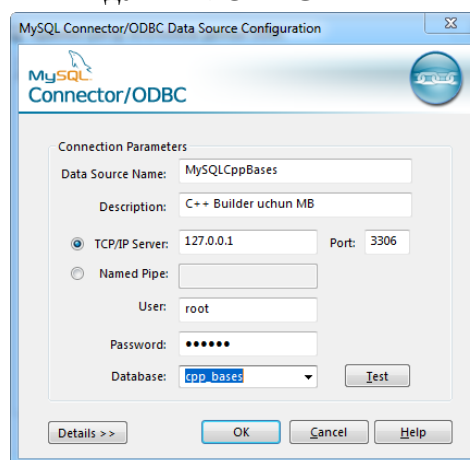
– 7.2.3–расмда кўрсатилган ойнадаги Добавить тугмаси босилади. Бу тугма орқали янги маълумотлар манбаи яратилади. Яратилган маълумотлар манбаига маълумотлар омбори ва MySQL МББТ га кириш ҳуқуқлари ёзилади ва алоҳида ном билан сақланади. Бу кетма-кетлик 7.2.4– ва 7.2.5–расмларда кўрсатилган.



7.2.3–расм: Администратор источника данных ODBC

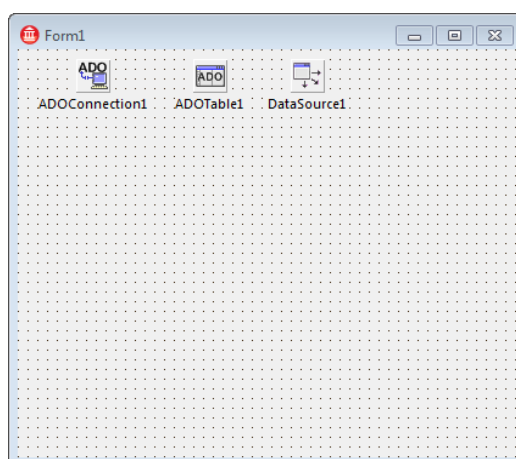


7.2.4–расм: MySQL ODBC ни танлаш



7.2.5–расм: MySQL МБга уланиш

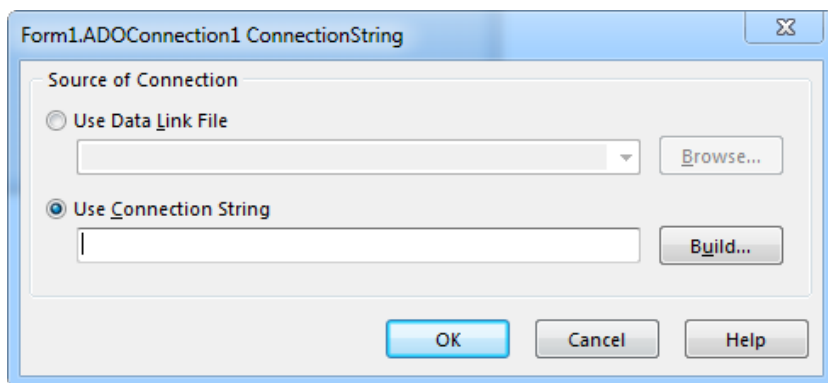
7.2.5–расмда келтирилган ойнадаги ОК тугмаси босилганида янги маълумотлар манбаи ҳосил бўлади. Янги маълумотлар манбаидан фойдаланиб C++Builder дастури ёрдамида маълумотлар омборига уланишни қуйида кўриб чиқилади.



7.2.6–расм: Формадаги компоненталар кўриниши

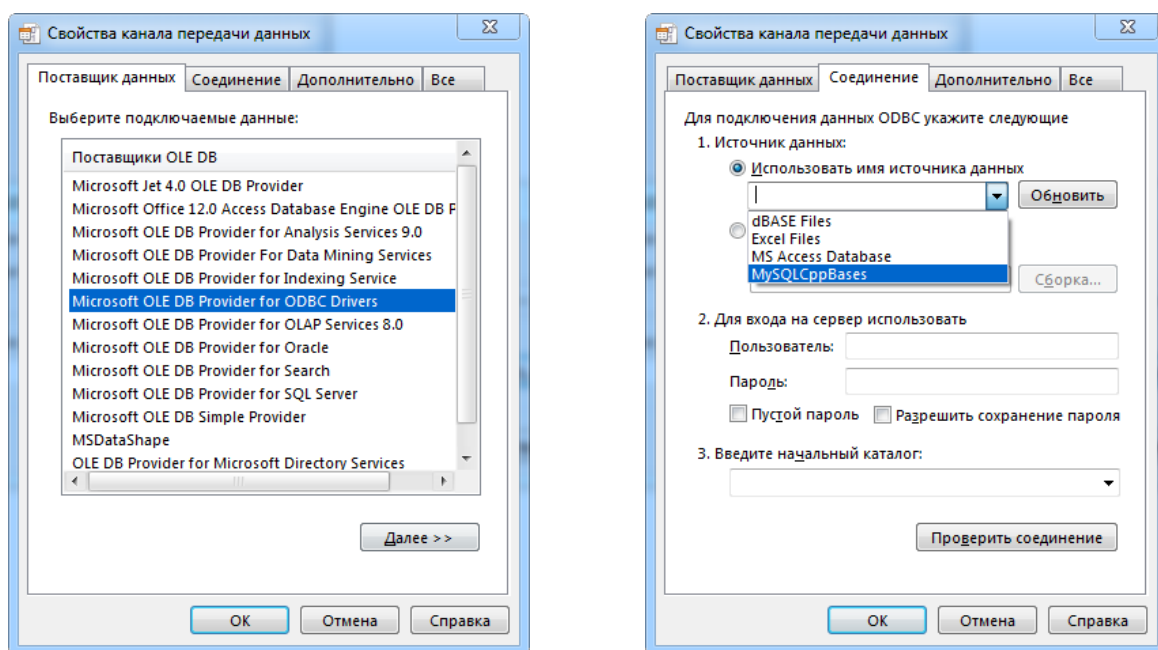
C++Builder муҳитида янги илова яратилади. ADO (C++Builder 6) ёки dbGo (C++Builder XE-XE8) компоненталар палитрасидан ADOConnection, ADOTable ва Data Access компоненталар палитра-

сидан DataSource компоненталарини формага жойлаштирилади. Бу форманинг кўриниши 7.2.6–расмда кўрсатилган. ADOConnection компонентасини танланади, Object Inspector ойнасидаги Properties бандидан connectionString хусусияти танланади. Бу хусусиятга ўзгартириш киритилади. Бунинг учун хусусиятнинг ўнг томонида жойлашган 3 нуқта тугмаси босилади. Шунда 7.2.7–расмда келтирилган ойна ҳосил бўлади.



7.2.7–расм: ConnectionString хусусиятини ўзгартириш ойнаси

7.2.7–расмда кўрсатилган ойнадан Build тугмаси босилади. Бу тугма босилганда 7.2.8–расмдаги ойна ҳосил бўлади. Бу ойнада келтирилган катталиклардан Microsoft OLE DB Provider for ODBC Drivers банди танланиб Далее тугмаси босилади. Шунда 7.2.8–расмда келтирилган ойнанинг Соединение бўлимига ўтади, бу бўлимдан Использовать имя источника данных бандини танланади.



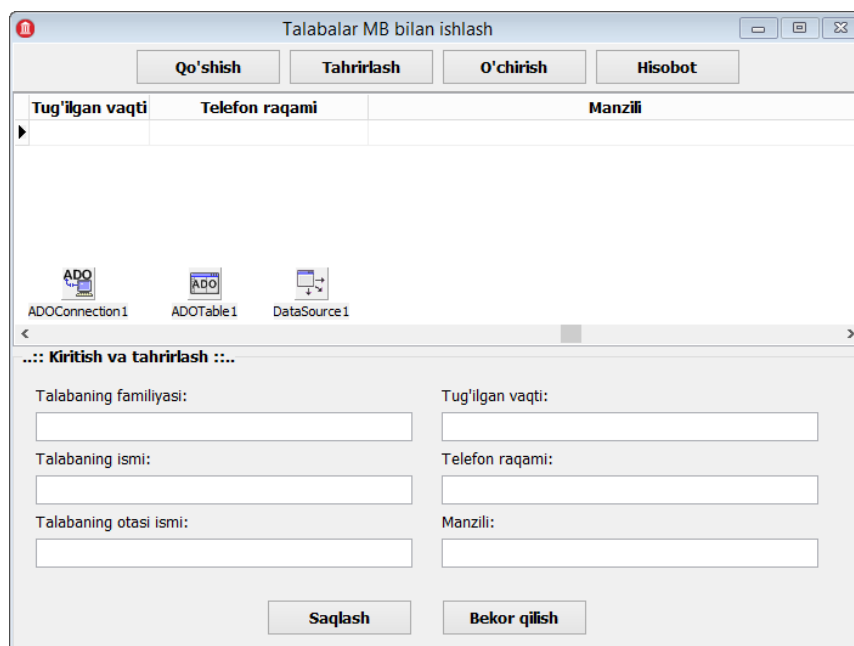
7.2.8–расм: Свойства канала передачи данных ойнаси

Бу бўлимда жойлашган танлай майдонидан MySQLCppBases танланади ва ОК тугмаси босилади. Шунда 7.2.7–расмда келтирилган

ойнадаги майдонга танланган маълумотларга мос равишдаги буйруқлар тўплами Use Connection String майдонига гавдаланади. Бу ойнадаги ОК тугмаси босилади. ADOConnection компонентасининг LoginPrompt хусусиятига false қиймат берилади. ADOTable компонентаси танланади. ADOTable компонентаси Connection хусусиятига ADOConnection1 номини берилади ёки сичқончанинг чап тугмаси икки марта чертилади. TableName хусусиятига маълумотлар омборидаги керакли бўлган жадвал номи танланади. Active хусусияти қиймати true билан алмаштирилади. DataSource компонентасининг DataSet хусусиятига ADOTable1 қиймати берилади. Ююқорида келтирилган кетма–кетликлар бажарилганидан сўнг, MySQL маълумотлар омборияратиётган илова билан боғланади.

Маълумотлар базасидаги жадвални дастур ёрдамида тахрирлаш

Маълумотлар базаси фойдаланувчи томонидан яратилаётган дастурга улангандан сўнг фойдаланувчи интерфейси яратилади. Фойдаланувчи интерфейсини яратишда маълумотлар омбори билан ишловчи компоненталар, буйруқ берувчи тугмалардан фойдаланилади (7.3.1–расм).



7.3.1–расм: Фойдаланувчи интерфейси

Юқорида келтирилган ойнадаги тугмаларнинг кодлари қуйидаги листингларда берилган.

Листинг 7.3.1: “Қўшиш” тугмасининг OnClick ҳодисаси

```
1: void __fastcall TForm1::Button1Click(TObject *Sender)
2: {
```

```

3:     // Qo'shish tugmasi bosilganda
4:     GroupBox1->Visible = True;
5:     ADOTable1->Insert();
6:     DBEdit1->SetFocus();
7: }

```

GroupBox1 бу компонента Форманинг қуйи қисмини эгаллаган бўлиб, компоненталар гуруҳидан иборат. Бу гуруҳда TLabel, TDBEdit ва TButton класслари ёрдамида объектлар яратилган. GroupBox1 объектининг Visible хусусияти true қиймат берилиб, бу объект кўриниши таъминланмоқда. ADOTable1 объектининг Insert методи ёрдамида жадвалга янги ёзув қўшиш учун рухсат берилмоқда. DBEdit1 объектининг SetFocus методи ёрдамида курсорни DBEdit1 объектга келтириш таъминланмоқда.

Листинг 7.3.2: “Тахрирлаш” тугмасининг OnClick ҳодисаси

```

1: void __fastcall TForm1::Button2Click(TObject *Sender)
2: {
3:     // Tahirlash tugmasi bosilganda
4:     GroupBox1->Visible = True;
5:     ADOTable1->Edit();
6:     DBEdit1->SetFocus();
7: }

```

Фаол маълумотлар тахрирлаш учун ADOTable1 объектининг Edit методи ёрдамида амалга оширилади. Бунда DBEdit1 объектларининг ҳар бири фаол турган ёзувнинг майдонларини қабул қилади.

Листинг 7.3.3: “Ўчириш” тугмасининг OnClick ҳодисаси

```

1: void __fastcall TForm1::Button3Click(TObject *Sender)
2: {
3:     // O'chirish tugmasi bosilganda
4:     int res = MessageDlg(
5:         "Yozuvni o'chirmoqchimisiz!",
6:         mtInformation,
7:         TMsgDlgButtons() << mbYes << mbNo, 0
8:     );
9:     if (res == mrYes) {
10:        ADOTable1->Delete();
11:    }
12: }

```

Жадвалдаги маълумотларни ўчириш учун фойдаланувчидан ўчириш рухсатномаси ёрдамида сўраш амалга оширилади. Агарда фойдаланувчи ўчириш учун рухсат берса, дастур томонидан фаол турган ёзув ўчирилади. Агарда фойдаланувчи томонидан ўчиришга

рухсат берилмаса, у ҳолда дастур ўчириш буйруғини инобатга олмайди.

Листинг 7.3.4: “Сақлаш” тугмасининг OnClick ҳодисаси

```
1: void __fastcall TForm1::Button5Click(TObject *Sender)
2: {
3:     // Saqlash tugmasi bosilganda
4:     ADOTable1->Post();
5:     GroupBox1->Visible = False;
6: }
```

Дастур жадвалга янги маълумот қўшиш ёки мавжуд маълумотларни тахрирлашдан сўнг сақлаш буйруғини амалга оширади. Бу буйруқ ADOTable1 объектининг Post методи ёрдамида амалга оширилади. GroupBox1 объектининг Visible хусусияти false қиймат берилиб, бу объект қўринмаслини таъминланмоқда.

Листинг 7.3.5: “Бекор қилиш” тугмасининг OnClick ҳодисаси

```
1: void __fastcall TForm1::Button6Click(TObject *Sender)
2: {
3:     // Bekor qilish tugmasi bosilganda
4:     ADOTable1->Cancel();
5:     GroupBox1->Visible = False;
6: }
```

Дастур жадвалга янги маълумот қўшиш ёки мавжуд маълумотларни тахрирлаш фойдаланувчи томонидан бекор қилиниши мумкин. Бу вақтда амалга оширилган ўзгаришларни бекор қилинади. Бу ўзгаришларни бекор қилиш учун ADOTable1 объектининг Cancel методидан фойдаланилади.

Назорат саволлари:

1. MySQL маълумотлар базаси билан ишлаш учун қандай дастур ёрдамида амалга оширилади?
2. Маълумотлар омборига уланиш қандай турлари мавжуд?
3. DBControl компоненталар палитрасидаги компоненталардан фойдаланмасдан маълумотлар омборига ёзув қўшиш мумкинми?
4. DbGrid компонентасининг Options хусусиятини тушунтириб беринг?

Амалий иш №8: Fast Report ва Rave Report компоненталар палитраси

Ишдан мақсад: Ушбу амалий ишида талабалар Fast Report компоненталар палитраси ёрдамида ҳисобот тайёрлашни ўрганадилар.

Керакли жиҳозлар ва ускуналар: Компютер, ведио проектер

Назарий маълумот

Маълумотлар омбори ёрдамида ишлайдиган дастурларнинг барчасида ҳисобот бериш имконияти мавжуд бўлади. Ҳисобот олишнинг турлари кўп бўлиб, уларга қуйидагаларни келтириш мумкин:

- QReport компоненталар палитраси (C++ Builder 6 учун);
- FastReport компоненталари (барча C++ Builder версиялари учун);
- RaveReport компоненталари (барча C++ Builder версиялари учун);
- Microsoft Excel дастурига юклаш.

Юқорида келтирилган ҳисобот шакллари компоненталар, махсус дастурлар ёки буйруқлар ёрдамида олинади. QReport компоненталар палитраси ёрдамида олинadиган ҳисоботлар асосан ўрнатилган компоненталар ёрдамида яратилади. Бу компонента C++ Builder 6 дастурида тайёр ўрнатилган бўлиб, бу компоненталар тўпламидан ихтиёрий марта фойдаланиш мумкин.

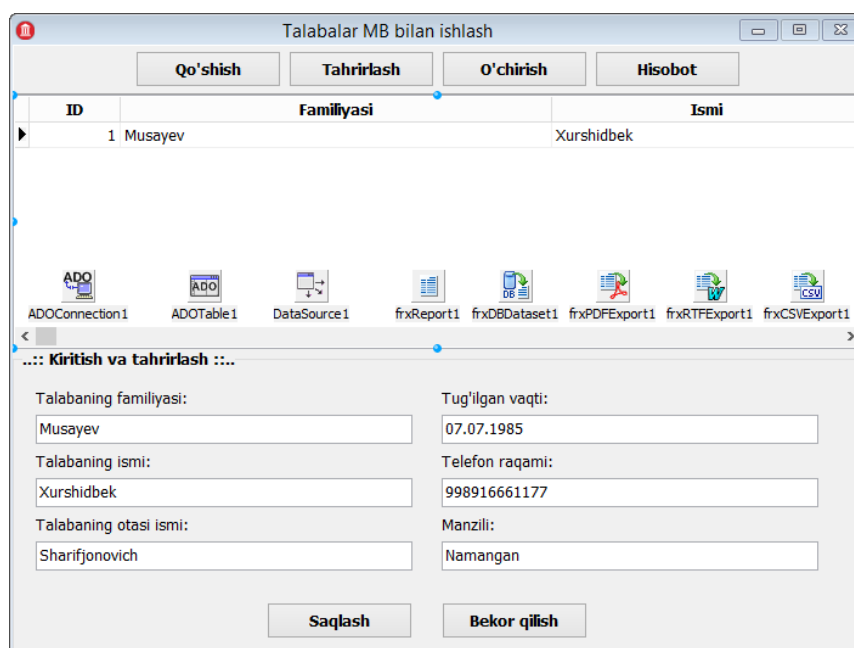
Амалий қисм.

FastReport компоненталари C++Builder барча версиялари учун ишлаб чиқилган. C++ Builder 6 учун бу компоненталар тўпламини қўшимча ўрнатиш керак бўлди. RAD Studio XE2 тўпламида бу комопенталар тўплами тайёр ўрнатилган. Маълумотлар базасидаги ёзувларнинг ҳисоботини олиш учун бу компоненталар тўпламида фойдаланишни кўриб чиқайлик:

- TfrxReport – бу компонента ёрдамида ҳисоботлар олиш файллари яратилади;
- TfrxDBDataset – бу компонента ёрдамида ҳисобот файлига маълумотлар омборини улаш;

- TfrxPDFExport – бу компонента ёрдамида олинган ҳисобот файлини PDF кўринишга ўтказиш;
- TfrxRTFExport – бу компонента ёрдамида олинган ҳисобот файлини RTF кўринишга ўтказиш;
- TfrxCSVExport – бу компонента ёрдамида олинган ҳисобот файлини CSV кўринишга ўтказиш.

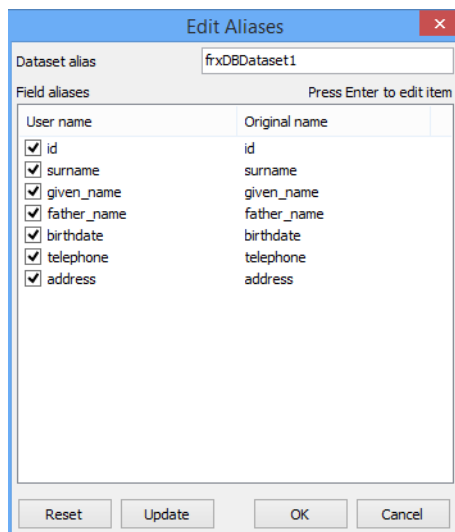
Юқорида келтирилган компоненталарни 7.3.1 – расмда келтирилган формага жойлаштирилади. Шунда 8.2.1–расмда келтирилган форма кўринишига келади.



8.2.1–расм: Фойдаланувчи интерфейси

8.2.1–расмда келтирилган ойнадаги frxReport1 ва frxDBDataset1 компоненталарни бир – бирига улади, яъни frxReport1 компонентасининг DataSet хусусиятига frxDBDataset1ни ёзилади. frxDBDataset1 компонента маълумотлар базаси билан ишлаш имконини берадиган ADOTable1 ёки DataSource1 компоненталаридан бирига уланади. Агар frxDBDataset1 компонентасига ADOTable1 компонентасини улаш керак бўлса, у ҳолда frxDBDataset1 компонентасининг DataSet хусусияти ёрдамида уланади. Агар frxDBDataset1 компонентасига DataSource1 компонентасини улаш керак бўлса, у ҳолда frxDBDataset1 компонентасининг DataSource хусусияти ёрдамида уланади. Уланиш амалга оширилганидан сўнг frxDBDataset1 компонентасининг устида сичқончанинг ўнг тугмаси чертилади, ҳосил бўлган контекст менюдан Edit Fields Aliases... банди танланади шунда 8.2.2–расмдаги ойна ҳосил бўлади. Бу ойнада

ҳисобот яратиш учун керак бўлган майдонлар номлари ва ҳисоботга маълумотлар базасини жўнатишдаги номи кўрсатилади. Маълумотлар базасини ҳисоботга жўнатишда киритиладиган ном Dataset alias майдонига киритилади. Ҳисоботга жўнатилаётган майдонларнинг номларига фойдаланувчи томонидан ўчириб қўйиш ёки ҳисоботга жўнатилмаслигини таъминлаш мумкин. Бу ўзгартиришлар ойнадаги жадвал кўринишда келтирилган қисмида амалга оширилади.



8.2.2–расм: Ҳисобот учун майдонларни тахрирлаш

8.2.1–расмда келтирилган ойнадаги тугмалардан **Hisobot** тугмаси ёрдамида ҳисобот олинади. Бунинг учун бу тугмага қуйидаги листингга келтирилган кодни ёзамиз.

Листинг 8.2.1: **Hisobot** тугмасига ёзиладиган код.

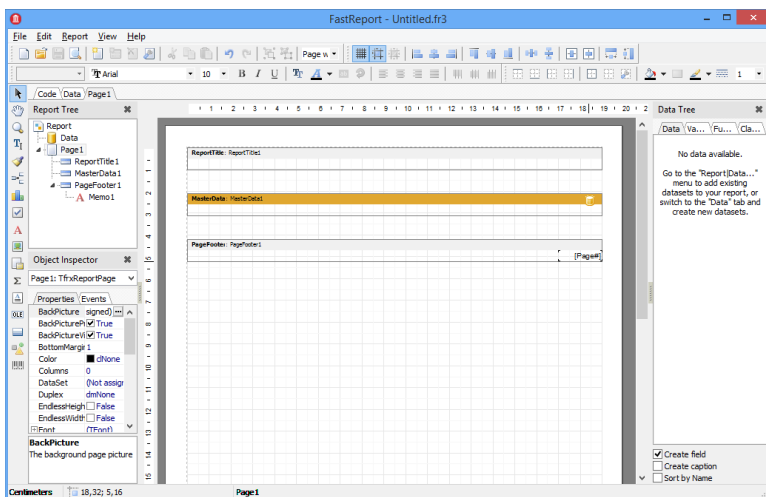
```

1: void __fastcall TForm1::Button4Click(TObject *Sender)
2: {
3:     // Hisobot tugmasi bosilganda
4:     String str;
5:     str = ExtractFilePath(Application->ExeName)+"\\report.fr3";
6:     frxReport1->LoadFromFile(str);
7:     frxReport1->ShowReport();
8: }

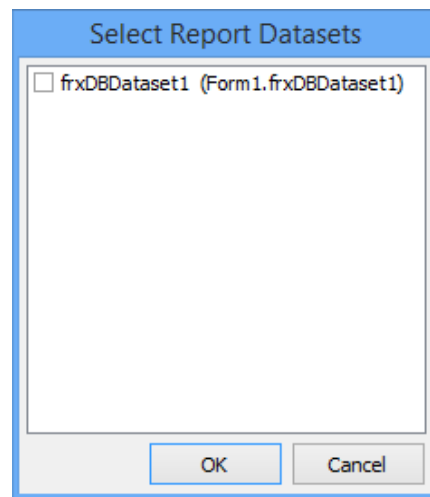
```

Бу кодни ёзишдан олдин ҳисобот файлини яратиш лозим. Бу файлини яратиш учун, frxReport1 компонентаси устида сичқончанинг ўнг тугмасини чертилади, ҳосил бўлган менюдан Edit Report банди танланади. Шунда ҳисобот файлини ҳосил қилувчи ойна ҳосил бўлади (8.2.3–расм). Бу ойна ёрдамида ҳисобот файли яратилади. Бунинг учун ойна ёрдамида файлини фойдаланувчи томонидан яратилган маълумотлар база билан боғлаш учун менюлар сатридаги Report

бандидан Data бандини танланади. Шунда 8.2.4–расмда келтирилган ойна ҳосил бўлади. Бу ойнадан керакли банднинг олдига белги қўйилади ва ОК тугмаси босилади. Шунда 8.2.3–расмда келтирилган ойнанинг ўнг қисмида маълумотлар базасининг майдонлари билан боғланган ҳолда, ҳисобот яратиш учун маълумотлар базасига боғланган майдонлар ҳосил бўлади.

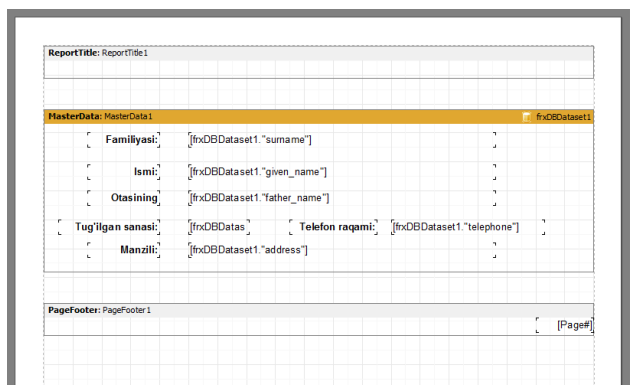


8.2.3–расм: FastReport ҳисоботлар яратиш ойнаси



8.2.4–расм: МБ танлаш

Ҳосил бўлган майдонларни ишчи соҳадаги MasterData бандига жойлаштирилади. MasterData бандини танлаиб, унинг хусусиятларидан DataSet бандига frxDBDataset1 ёки фойдаланувчи томонидан ўзгартириш киритилган номи берилади. MasterData бандига жойлаштирилган майдонларни турли кўринишда жойлаштириш мумкин (8.2.5–расм).



8.2.5–расм: MasterData бандига майдонларни жойлаштириш

8.2.5–расмда келтирилган MasterData майдонидаги изоҳларни ёзиш учун 8.2.3–расмда келтирилган ойнанинг чап томонида келтирилган компонентлардан Text object компонентасидан фойдалинилади. Бундан ташқари ҳисоботларни жадвал шаклида олиш мумкин. Бунинг учун ҳисоботни тайёрлашда ҳар бир майдоннинг хошиясидаги чизиқларни кўринишини таъминлаш керак

бўлади. Бу ҳисоботни тайёрлашда ReportTitle бандини қўлланилади. Бу бандга жадвал кўринишининг юқори сарлавҳа қисмини, жадвалнинг керакли бўлган изоҳлар келтириш мумкин (8.2.6–расм). Ҳисобот яратишда маълумотларни бир – бирига қўшиш ёки бирлаштириш, стандарт функциялардан фойдаланиш ёки фойдаланувчи томонидан яратилган функциялардан фойдаланиш мумкин.

ReportTitle: ReportTitle1				
Talabalar haqida ma'lumot				
№	Talabani FIOsi	Tug'ilgan sanasi	Telefon raqami	Manzili
1	[fxDBDataset1."given_name"]	[fxDBDataset1."birthdate"]	[fxDBDataset1."telephone"]	[fxDBDataset1."address"]

PageFooter: PageFooter1 [Page#]

8.2.6–расм: Жадвал кўринишдаги ҳисоботлар тайёрлаш.

Жадвал кўринишда тайёрланган ҳисоботни ҳам листинг 8.2.1 да кўрсатилганидек чақириш мумкин. Агар икки ва ундан ортиқ кўринишдаги ҳисоботлар мавжуд бўлса, у ҳолда листинг 8.2.1 га ўзгартиришлар киритилади. Бу ўзгартиришлар листинг 8.2.2 да келтирилган.

Листинг 8.2.2: **Hisobot** тугмаси босилганда сўров ойнаси ҳосил қилиш.

```

1: void __fastcall TForm1::Button4Click(TObject *Sender)
2: {
3:     // Hisobot tugmasi bosilganda
4:     String str;
5:     TForm *frm;
6:     frm = new TForm(Application);
7:     frm->BorderStyle = bsDialog;
8:     frm->Height = 200;
9:     frm->Width = 300;
10:    frm->Position = poScreenCenter;
11:    frm->Caption = "Hisobot turini tanlash...";
12:    frm->Font->Size = 14;
13:    TRadioButton *rbPage = new TRadioButton(frm);
14:    rbPage->Caption = "Alohida ko'rinishi";
15:    rbPage->Parent = frm;
16:    rbPage->Left = 50;
17:    rbPage->Top = 30;
18:    rbPage->Width = 200;
19:    TRadioButton *rbTable = new TRadioButton(frm);

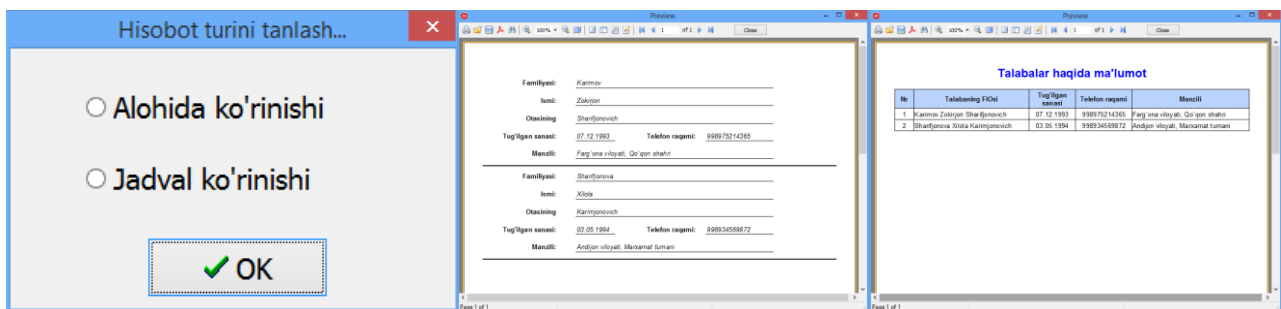
```

```

20:   rbTable->Caption = "Jadval ko'rinishi";
21:   rbTable->Parent = frm;
22:   rbTable->Left = 50;
23:   rbTable->Top = 30 + rbPage->Height + 30;
24:   rbTable->Width = 200;
25:   TBitBtn *myBtnOK = new TBitBtn(frm);
26:   myBtnOK->Parent = frm;
27:   myBtnOK->Kind = bkOK;
28:   myBtnOK->Top = rbTable->Top + rbTable->Height+30;
29:   myBtnOK->Width = 120;
30:   myBtnOK->Height = 40;
31:   myBtnOK->Left = 90;
32:   if(frm->ShowModal() == mrOk)
33:   {
34:       str = ExtractFilePath(Application->ExeName);
35:       if (rbPage->Checked)
36:           str = str+"\\report.fr3";
37:       else
38:           str = str+"\\report1.fr3";
39:       frxReport1->LoadFromFile(str);
40:       frxReport1->ShowReport();
41:   }
42:   delete myBtnOK;
43:   delete rbPage;
44:   delete rbTable;
45:   delete frm;
46: }

```

Юқоридаги листингнинг коди бажарилиши натижаси 8.2.7–8.2.9 – расмларда келтирилган.



8.2.7–расм: Ҳисобот турини танлаш ойнаси

8.2.8–расм: Ҳисоботнинг оддий кўриниши

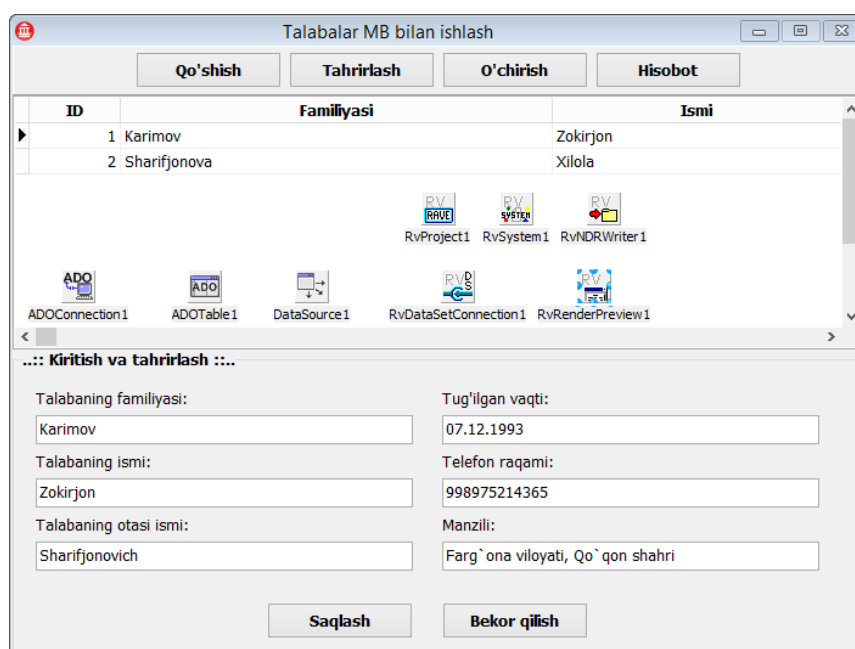
8.2.9–расм: Ҳисоботнинг жадвал кўриниши

RaveReport компоненталар палитраси компоненталари C++Builder дастурининг барча версияларида мавжуд бўлиб, у билан ишлаш бошқа ҳисобот компоненталари билан ишлаш анчагина қийинроқ ҳисобланади, аммо QReport компоненталарида қила олинмайдиган ишларни қилса бўлади. Фойдаланувчи томонидан яратилган дастурга ҳисобот тайёрлашда RaveReport компоненталаридан фойдаланиш учун дастурлаш муҳитининг

компоненталар палитрасидаги Rave бўлимидан фойдаланамиз. Қуйида Rave компоненталар тўпламининг асосий компоненталари келтирилган.

- TRvProject – бу компонента ёрдамида ҳисоботлар олиш файллари яратилади;
- TRvSystem – бу компонента ёрдамида ҳисобот файли билан фойдаланувчи томонидан яратилган дастурни ўртасидаги алоқани тиклаш учун хизмат қилади;
- TRvDataSetConnection – бу компонента ёрдамида ҳисобот файлига маълумотлар омборини DataSet ёрдамида улаш;
- TRvRenderPDF – бу компонента ёрдамида олинган ҳисобот файлини PDF кўринишга ўтказиш;
- TRvRenderRTF – бу компонента ёрдамида олинган ҳисобот файлини RTF кўринишга ўтказиш;
- TRvRenderHTML – бу компонента ёрдамида олинган ҳисобот файлини HTML кўринишга ўтказиш.

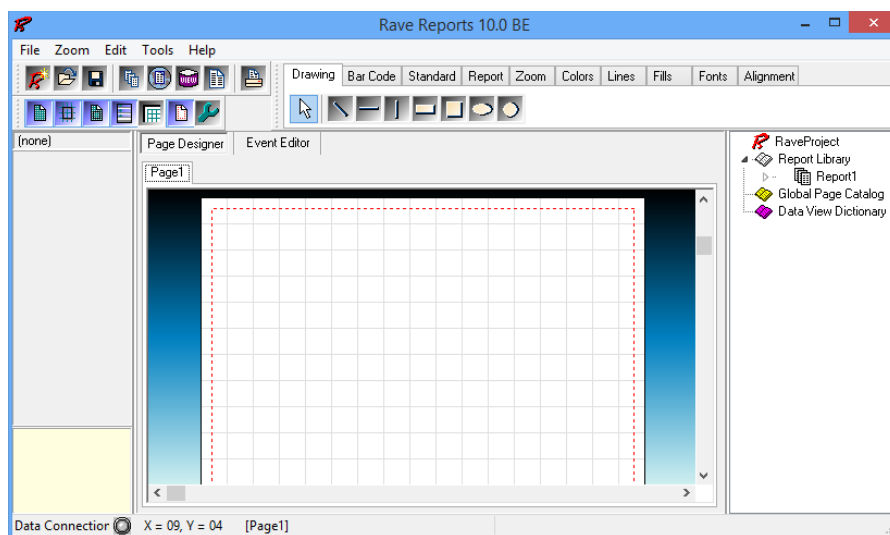
RaveReport компоненталар палитрасидаги компоненталарни 7.3.1–расмда келтирилган ойнадаги формага ўрнатамиз.



8.3.1–расм: Фойдаланувчи интерфейси (RaveReport компоненталари)

RaveReport компонентаси ёрдамида ҳисобот олиш учун компоненталар бир–бирига боғланиши керак бўлади. Биринчи навбатда ҳисобот олишда маълумотлар базасини ҳисобот компонентасига уланади. Бу компоненталар фойдаланувчи томонидан яратилаётган формадаги ADOTable1 ва

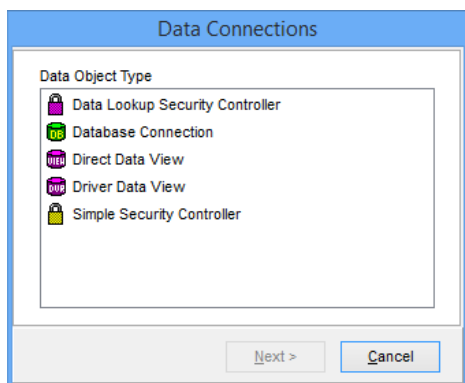
RvDataSetConnection1 компоненталари ҳисобланади. RvDataSetConnection1 компонентасининг DataSet хусусиятига ADOTable1 компонентаси номи ёзилади. Бу икки компоненталар боғлангандан сўнг RvProject1 ва RvSystem1 компоненталари ўзаро боғланади, яъни RvProject1 компонентасининг Engine хусусиятига RvSystem1 номи ёзилади. Бу боғланишлар амалга оширилганидан сўнг ҳисоботнинг кўринишини тайёрлаб олишимиз керак бўлади. Ҳисоботнинг кўринишини тайёрлаш учун 8.3.1–расмда кўрсатилган формадаги RvProject1 компонентасининг устида сичқончанинг ўнг тугмасини чертамыз, ҳосил бўлган оқиб чиқувчи менюдан Rave Visual Designer... бандини танлаймыз. Шунда 8.3.2–расмдаги ойна ҳосил бўлади.



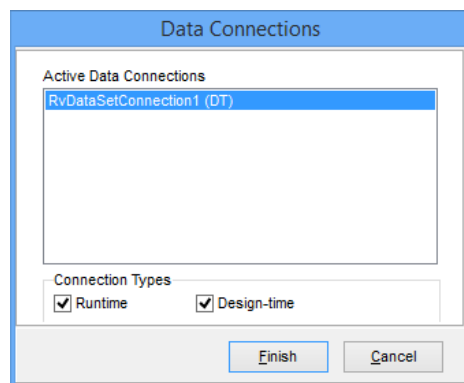
8.3.2–расм: RaveReport ҳисоботларни яратиш ойнаси

8.3.2–расмда келтирилган ойнадаги Page1 бўлимига юқорида жойлашган Report компоненталар палитрасидаги Region компонентасини жойлаштирилади. Region компонентасининг ичига Report компоненталар палитрасидаги DataBand компонентасини жойлаштирилади. Дастурдаги маълумотлар омбори билан боғлаш учун File менюсидан New Data Object банди танланади, шунда 8.3.3–расмдаги ойна ҳосил бўлади. Бу ойнадан Direct Data View банди танланади ва Next тугмаси босилади. Ҳосил бўлган ойнадаги Active Data Connections бўлиmidан керакли уланиш танланади ва Finish тугмаси босилади (8.3.4–расм). Маълумотлар омбори ҳисоботга боғланганидан сўнг жойлаштирилган компоненталар билан маълумотлар омбори боғланади. DataBand компонентасининг DataView хусусиятига DataView1 деб ёзилади. Бу жараён ҳар бир ёзувни алоҳида – алоҳида кўрсатиб бериш учун қўлланилади. Ҳар бир

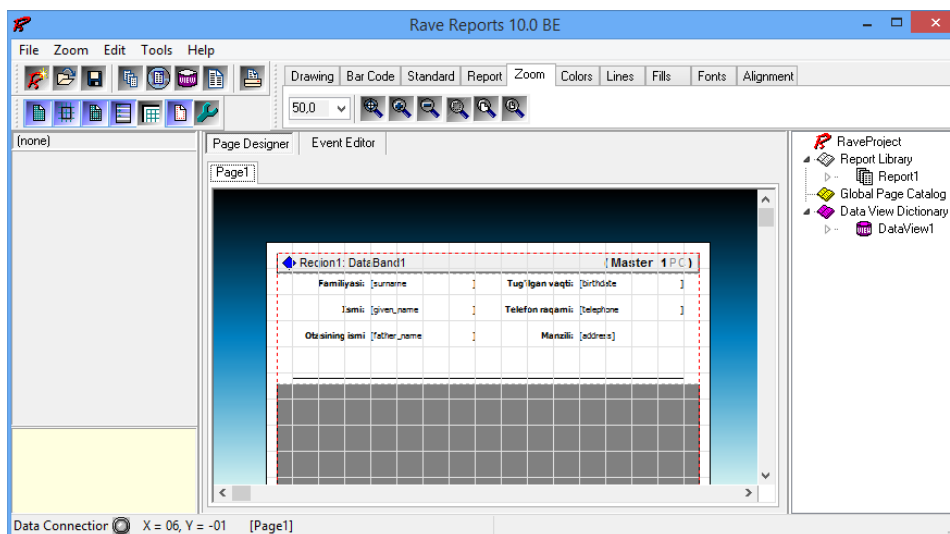
ёзувдаги майдонларни кўрсатиб бериш учун Standard ва Report компоненталар палитрасидаги Text ва DataText компоненталаридан фойдаланилади. Бу компоненталарни DataBand1 компонентасининг ичига 8.3.5–расмда кўрсатилганидек жойлаштирилади. DataText компоненталарнинг ҳар бирини DataView ҳусусиятига DataView1 ни ёзилади ва ҳар бир компонентанинг DataField ҳусусиятига маълумотлар базасидаги ёзувларнинг майдонлари танланади.



8.3.3–расм: Маълумотлар оқими турини ойнаси



8.3.4–расм: Фаол маълумотлар омборини танлаш ойнаси



8.3.5–расм: Ҳисобот файлини ҳосил қилиш

Ҳисобот файли сақланади. Ҳисоботни кўриш учун 8.3.1–расмдаги Hisobot тугмасига қуйидаги код ёзилади.

```

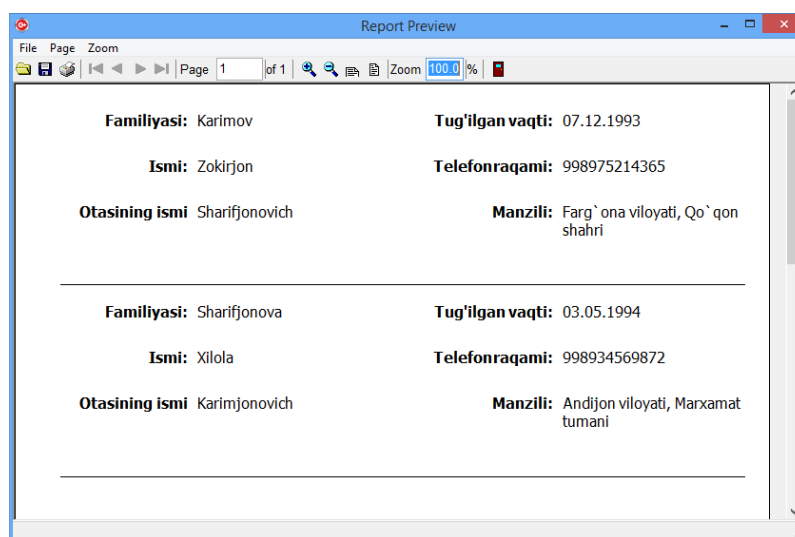
1: void __fastcall TForm1::Button4Click(TObject *Sender)
2: {
3:     String str;
4:     str = ExtractFilePath(Application->ExeName);
5:     str = str+"\\oddiy.rav";
6:     RvProject1->ProjectFile = str;
7:     RvSystem1->DefaultDest = rdPreview;
8:     RvProject1->Execute();
9: }

```

Бу коддан ташқари RaveReport билан ишлаётган вақтда форманинг яратилиш жараёнида RaveReport компоненталарининг баъзи бир буйруқлари ёзилиши керак бўлади. Бу буйруқлар форманинг OnCreate ходисасига ёзилади.

```
1: void __fastcall TForm1::FormCreate(TObject *Sender)
2: {
3:     RvProject1->SelectReport("Report1", true);
4:     RvProject1->Open();
5: }
```

Hisobot tugmasi bosilganida 8.3.6–расмдаги ойна ҳосил бўлади.



8.3.6–расм: Ҳисобот файлини ҳосил қилиш

Ҳисобот файлини сақлаш тугмаси орқали фойдаланувчи ихтиёрий форматда сақлаши мумкин. Бу форматларга мисол сифатида HTML, PDFларни келтиришимиз мумкин.

Маълум бир дастурлар фойдаланишда фойдаланувчиларга турли хилдаги қулайликлар яратилади. Бу қулайликлар фойдаланувчига турли хилдаги ҳисоботлар олиш учун хизмат қилади. Шундай дастурлар тузишда ундаги маълумотларни Microsoft Excel дастурига экспорт қилиш ёки MS Excelдаги маълумотларни импорт қилиш мумкин. Бу жараён дастурчи томонидан турли хилдаги йўллар билан амалга оширилади. MS Excel маълумотлари билан ишлаш учун OLE серверларидан фойдаланиш мақсадга мувофиқ. OLE сервери билан ишлаш учун қуйидаги код фойдаланувчи томонидан кодлар муҳарририга ёзилади.

```
1: #include <ComObj.hpp>
2: #include <utilcls.h>
```

Бу икки буйруқ OLE серверни фойдаланувчи томонидан яратилаётган дастурга улайди. OLE сервери ёрдамида MS Excel, MS Word ва WordPad дастурлардан фойдаланмаган ҳолда маълумотлар ёзиш мумкин. Ҳозир MS Excel билан ишлашни кўрайлик.

OLE серверини MS Excel билан ишлаш принципи қуйидагича:

1: Илова => Китоблар тўплами => Китобнинг номи => Варақлар тўплами => Варақ номи => Катакчалар тўплами (Катакча номи)

Юқорида келтирилган принципдан фойдаланиб ўзгарувчилар яратилади. Бу ўзгарувчиларнинг типи ихтиёрий вариантда бўлишини таъминлаш мақсадида Variant типи танланади.

```
1: Variant vVarApp; // илова учун
2: Variant vVarBooks; // Китоблар тўплами учун
3: Variant vVarBook; // Китоб номи учун
4: Variant vVarSheets; // Варақлар тўплами учун
5: Variant vVarSheet; //Варақ номи учун
6: Variant vVarCells; //Катакчалар тўплами учун
7: Variant vVarCell; //Катакча номи учун
```

Юқоридаги ўзгарувчилардан фойдаланилган ҳолда, MS Excelни ишга туширилади. Бу буйруқ қуйидагича:

```
1: vVarApp = CreateOleObject("Excel.Application");
```

Бу буйруқ натижасида MS Excel дастури ишга туширилади, аммо экранда кўринмайди. Дастур экранда кўриниши учун унинг баъзи бир хусусиятларига қийматлар ўрнатиш керак бўлади. OLE ёрдамида яратилган объектларга қиймат бериш ёки унинг қийматини аниқлаш оддий қиймат узатиш ёки қабул қилишдан фарқланади, яъни қийматлар объектнинг хусусиятига ўрнатилади ёки объектнинг хусусиятининг қиймати олинади.

Объект хусусиятининг қиймати ўрнатилаётган бўлса, у ҳолда қуйидаги кодга мурожаат қилинади.

```
1: vVarApp.OlePropertySet("хусусият_номи", қиймати);
```

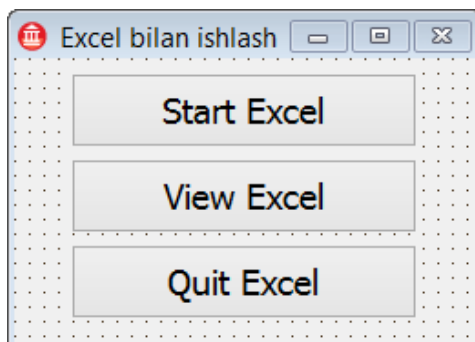
Агар хусусиятининг қиймати аниқланаётган бўлса, у ҳолда қуйидаги кодга мурожаат қилинади.

```
1: ўзгарувчи_номи = vVarApp.OlePropertyGet("хусусият_номи");
```

Агар объектнинг бирон бир методига мурожаат қилинаётган бўлса, у ҳолда бу буйруқ қуйидагича амалга оширилади.

```
1: vVarApp.OleProcedure("методнинг_номи", методнинг_параметрлар_рўйхати);
```

Юқорида келтирилган буйруқлар ёрдамида MS Excel дастурини ишга туширишни кўриб чиқайлик. Бунинг учун C++ Builder дастурида янги илова яратамиз. Илованинг асосий формасига 3 та тугма жойлаштирамиз. Бу тугмаларнинг сарлавҳасига Excelни ишга тушириш (Start Excel), ишчи столда Excelни кўрсатиш (View Excel) ва Excelни ёпиш (Quit Excel) деб ўзгартиришлар киритамиз (8.4.1–расм).



8.4.1–расм: Excel билан ишлаш ойнаси

8.4.1–расмда келтирилган ойнадаги Start Excel тугмасининг OnClick ҳодисасига қуйидагича код ёзамиз.

```
1: void __fastcall TForm1::Button1Click(TObject *Sender)
2: {
3:     /*
4:     *     Start Excel тугмаси босилганда
5:     */
6:     if (!fStart) {
7:         vVarApp = CreateOleObject("Excel.Application");
8:         fStart = true;
9:     }
10:    else
11:    {
12:        MessageBoxA(0, "Excel dasturi oldin ishga tushirilgan",
13:        "Natolik!", MB_OK);
14:    }
```

Бу буйруқни таҳлил қиладиган бўлсак, 6 қатордаги fStart ўзгарувчисини глобал ўзгарувчи сифатида эълон қилиниб, литерал false қиймат узатишган. Еттинчи қатордаги vVarApp ўзгарувчиси ҳам, глобал ўзгарувчи сифатида эълон қилинган бўлиб, унинг типи Variant типидадир. Дастур агар fStart ўзгарувчисининг қиймати рост бўлмаса, яъни Excel дастури олдин ишга туширилмаган бўлса, у ҳолда буйруқ Excel дастурини ишга тушириш учун буйруқ беради. Агар аксинча бўлса, у ҳолда ҳатолик ҳақида хабар беради.

8.4.1–расмдаги иккинчи тугма, яъни View Excel тугмасининг кодини кўриб чиқамиз.


```

1: void __fastcall TForm1::Button2Click(TObject *Sender)
2: {
3:     /*
4:     *     View Excel тугмаси босилганда
5:     */
6:     if (fStart) {
7:         vVarApp.OlePropertySet("Visible",true);
8:     }
9:     else
10:    {
11:        MessageBoxA(0,"Excel dasturi ishga tushirilmagan",
    "Natolik!", MB_OK);
12:    }
13: }

```

Бу дастурнинг олтинчи қаторида келтирилган шартни таҳлил қиладиган бўлсақ, fStart ўзгарувчисининг қиймати рост бўлса, яъни Excel дастури ишга туширилган бўлса, у ҳолда Excel дастурини кўрсатиш учун, Excel дастурини ифодаловчи объектнинг Visible хусусиятига true қийматини ўрнатиш еттинчи қаторда кўрсатилган. Агар Excel дастури ишга туширилмаган бўлса, у ҳолда дастур ишга туширилмаганлиги ҳақидаги ҳатолик кўрсатилади.

Юқорида келтирилган 8.4.1–расмдаги 2 та тугманинг кодини кўриб чиқдик. Охириги Quit Excel тугмасининг коди қуйида келтирилган.

```

1: void __fastcall TForm1::Button3Click(TObject *Sender)
2: {
3:     /*
4:     *     Quit Excel тугмаси босилганда
5:     */
6:     if (fStart) { vVarApp.OleProcedure("Quit");
7:         fStart = false;
8:     }
9:     else { MessageBoxA(0, "Excel dasturi ishga tushirilmagan",
    "Natolik!", MB_OK);
10:    }
11: }

```

Юқорида келтирилган дастурда Excel дастури ишга туширилган бўлса, Excel дастурини ёпиш учун объектнинг Quit методи ишга туширилмоқда.

Фойдаланувчи томонидан Excel дастури ишга туширилганида унинг ичида ҳеч қандай лист мавжуд бўлмайди. Фойдаланувчи листларни ҳам махсус буйруқлар ёрдамида яратади. Бу буйруқлар қуйидагилар ҳисобланади.

```
1: vVarBooks=vVarApp.OlePropertyGet("Workbooks");
2: vVarApp.OlePropertySet("SheetsInNewWorkbook",3);
3: vVarBooks.OleProcedure("Add");
```

Юқорида келтирилган буйруқлар тўплами ишчи китобга листлар қўшишни изоҳлайди. Ҳар бир қаторини кўриб чиқадиган бўлсак, биринчи қаторда жорий ишчи китобни белгилаш келтирилган. Бу махсус буйруқ ёрдамида жорий ишчи китобни vVarBooks номи билан белгилаб олинган. Иккинчи қаторни кўриб чиқадиган бўлсак, Excel дастурининг янги ишчи китоби учун лист қўшишлар сонини белгилаш келтирилган. Учинчи қаторда ишчи китобга лист қўшиш методини ишга тушириш буйруғи келтирилган.

Excel дастурининг ишчи китобидаги ҳар бир листини ҳам махсус номлар ёки ўзгарувчилар билан белгилаш мумкин. Китобнинг листларни алоҳида ўзгарувчи сифатида кўришдан олдин биз жорий ишчи китоби алоҳида ўзгарувчи сифатида қарашимизга тўғри келади.

```
1: vVarBook=vVarBooks.OlePropertyGet("Item",1);
```

Ишчи китобни ўзгарувчи сифатида белгилаб олганимиздан сўнг ишчи китобидаги бирча листларни алоҳида ўзгарувчи сифатида белгилаб оламиз.

```
1: vVarSheets=vVarBook.OlePropertyGet("Worksheets");
```

Листлар тўпламида ўзимизга керакли бўлган листни белгилаш учун қуйидаги дастур қисмини кўриб чиқамиз.

```
1: vVarSheet = vVarSheets.OlePropertyGet("Item",1);
2: vVarSheet.OlePropertySet("Name", "Hisobot");
```

Юқоридаги дастурнинг қисмида белгиланган листнинг номини ўзгартириш кўрсатилган. Яъни, ишчи китобининг 1 листининг номини Hisobot кўринишига келтирилган.

Листдаги катакчаларга маълумотларни киритиш учун буйруқлардан фойдаланилади. Бу буйруқлар қуйида берилган.

```
1: vVarSheet = vVarSheets.OlePropertyGet("Item",1);
2: vVarCell = vVarSheet.OlePropertyGet("Cells")
3:         .OlePropertyGet("Item",x,y);
```

Юқоридаги дастур кодида x ва y координатага эга бўлган катакча белгиланган. Катакчаларнинг гуруҳини белгилаш ҳам мумкин. Бу гуруҳни бир катак сифатида, яъни бирлаштириш ҳам мумкин. Бу буйруқлар тўплами қуйида кўрсатилган.

```
1: vVarCell=vVarSheet.OlePropertyGet("Range", "A1:C4");
```

2: vVarCell.OleProcedure("Merge");

Юқорида келтирилган буйруқлар тўпламида 12 та катакча бирлаштирилган. Катакларга ёзиладиган маълумотларни текислаш, шрифтларни ўзгартириш, маълумотларни киритиш қуйидагича амалга оширилади.

Матнни текислаш:

1: vVarCell.OlePropertySet("HorizontalAlignment", -4108);

2: vVarCell.OlePropertySet("VerticalAlignment", -4108);

Бу буйруқлар матнни горизонтал ва вертикал ҳолатда марказлаштиради. Текислашнинг ҳар бир буйруқлари ўзгармас сон ёки катталиклардир. Бу катталикларни қуйидаги жадвалда кўришимиз мумкин:

Катталикнинг номи	Қиймати
Горизонтал катталиклар	
xlHAlignCenter	-4108
xlHAlignJustify	-4130
xlHAlignLeft	-4131
xlHAlignRight	-4152
Вертикал катталиклар	
xlVAlignBottom	-4107
xlVAlignCenter	-4108
xlVAlignJustify	-4130
xlVAlignTop	-4160

Шрифт ўлчамини, рангини ва кўринишини ўрнатиш

1: vVarCell.OlePropertyGet("Font").OlePropertySet("Size",15);

2: vVarCell.OlePropertyGet("Font").OlePropertySet("Color",clBlue);

3: //Жирный

4: vVarCell.OlePropertyGet("Font").OlePropertySet("Bold",true);

5: //Курсив

6: vVarCell.OlePropertyGet("Font").OlePropertySet("Italic",true);

Шрифт номини ўрнатиш

1: vVarCell.OlePropertyGet("Font").OlePropertySet("Name","Arial");

Катакчага маълумотлар ва формулалар ёзиш

1: vVarCell.OlePropertySet("Value","Мой текст");

2: vVarCell.OlePropertySet("Value",25);

3: vVarCell.OlePropertySet("Value","=СУММ(A1:A10)");

Устун ёки сатрнинг ўлчамини ўзгартириш

```
1: vVarCell.OlePropertySet("RowHeight", 20);  
2: vVarCell.OlePropertySet("ColumnWidth", 10);
```

Юқорида келтирилган буйруқларга таянган ҳолда фойдаланувчи томонидан маълумотлар Excel дастурига юклаш мумкин.

Назорат саволлари:

1. Ҳисобот яратиш нима учун керак?
2. Ҳисобот яратишнинг қандай усулларни биласиз?
3. FastReport компоненталар палитраси компоненталарини санаб беринг?
4. RaveReport компоненталар палитраси компоненталарини санаб беринг?
5. MS Word дастурига маълумотлар базасидаги маълумотларни буйруқлар ёрдамида юклаш мумкинми?
6. MS Excel дастурига фойдаланувчи томонидан буйруқлар ёзиб диаграммалар ҳосил қилиш мумкинми?

Амалий иш №9: Ўрнатиловчи дастурлар яратиш.

Ишдан мақсад: Ушбу амалий ишида талабалар ўрнатиловчи дастур яратиш мақсади ва вазифасини ўрганадилар.

Керакли жиҳозлар ва ускуналар: Компютер, ведио проектер.

Назарий маълумот

Замонавий дастурлар дискетларда, CD ёки DVD–дискларда жойлашган бўлади. Дастурни ўрнатиш жараёнларини, яъни, керакли каталогларни яратиш ва унга мос ишчи файлларни ҳамда маълумотларни ўзида сақловчи файлларни жойлаштириш кўпгина фойдаланувчилар учун қийинчилик туғдиради. Шунинг учун, амалий дастурларни компютерга ўрнатишда махсус дастурлардан фойдаланилади. Бу дастурлар ўрнатиловчи дастурлар деб номланади.

Ўрнатиловчи дастурлар бошқа дастурлар сингари яратилиши мумкин. Лекин бу жуда ҳам мураккаб жараён ҳисобланади. Юқоридагилардан келиб чиққан ҳолда, ўрнатиловчи дастурларни яратувчи махсус дастурлар тузилган бўлиб, у ёрдамида дастурчи ҳеч қандай код ёзмасдан ўрнатиловчи дастур яратиши мумкин. Бундай дастурлар кўп дастурларни келтириш мумкин. Булардан энг содда ва қулайи бўлган InstallShield® Express™ дастури билан ишлашни ўрганамиз.

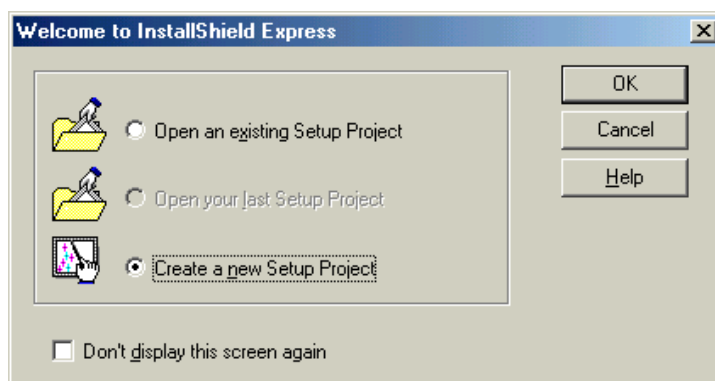
Амалий қисм.

InstallShield® Express™ дастури Borland фирмаси томонидан ишлаб чиқилган бўлиб, ўрнатиловчи дастурлар яратишда анчагина қулайликларни беради. Ўрнатиловчи дастурлар яратиш учун дастурнинг ўзи тайёр ҳолга келтирилган бўлиши керак бўлади. Дастур тайёр ҳолатга келтирилганидан сўнг унга керакли бўлган қўшимча файлларни кўздан кечирган ҳолда, ўрнатиловчи дастур яратиш бошланади.

Эслатма: InstallShield Express утилити Borland фирмаси пакетининг ичида жойлашган бўлади. Уни ўрнатиш учун Delphiни ўрнатувчи дастурни ишга туширилади ва ҳосил бўлган Delphi Setup Launcher мулоқот ойнасидан InstallShield Express Custom Edition for Delphi буйруғи танланади.

Ўрнатиловчи дастур яратиш учун InstallShield Express ни ишга туширилади. Ҳосил бўлган Welcome мулоқот ойнасидан Create a New

Setup Project (янги проект яратиш) ни танлаб, ОК тугмаси босилади (9.2.1–расм).

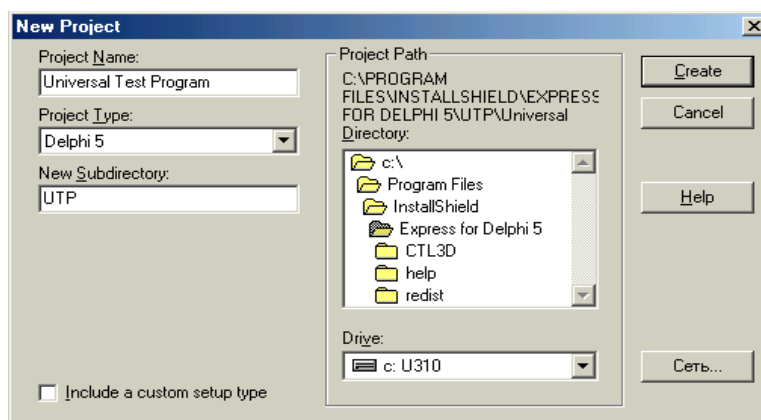


9.2.1–расм: Welcome мулоқот ойнаси

Ҳосил бўлган New Project мулоқот ойнасидан проект ҳақида маълумотлар киритилади (9.2.2–расм):

Project Name–проектнинг номи;

New Subdirectory–майдонига каталог номи, InstallShield Express дастури ушбу каталокка барча яратилувчи файлларни жойлаштиради. Directory ва Drive рўйхатдан фойдаланиб проект каталогини қаерга жойлашишини кўрсатиш мумкин.



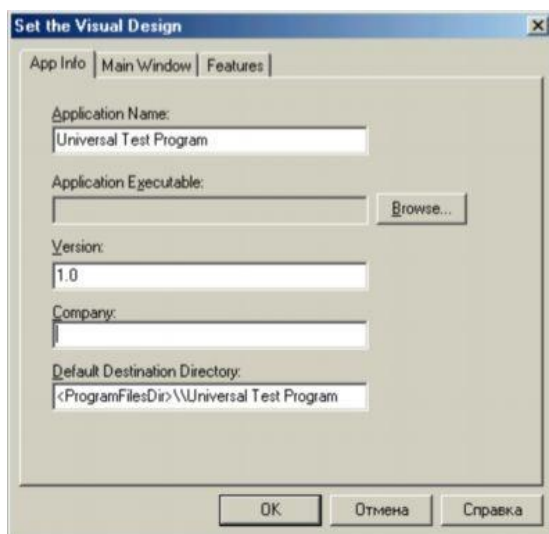
9.2.2–расм: New Project мулоқот ойнаси

Create тугмаси босилгандан сўнг ўрнатиловчи дастур яратувчи проект ойнаси очилади. Бу ерда яратилаётган ўрнатувчи дастур параметрларини аниқловчи буйруқлар кетма-кетлиги жойлашган.

Ўрнатиловчи дастурни сошлаш буйруқлари гуруҳларга ажритилган (гуруҳ сарлавҳаси қалин харфлар билан ажратилган). Буйруқни танлаш учун сичқончанинг чап тугмасини буйруқ сатрида чертиш етарли. Натижада буйруқнинг мулоқот ойнаси очилади.

Set the Visual Design гуруҳ буйруқлари ўрнатиловчи илова тўғрисидаги умумий маълумотларни бериш ва ўрнатувчи дастурнинг

бош ойна кўринишини белгилаш имконини беради. Application Information буйруғни танлаш билан App Info саҳифаси ойнаси очилади (9.2.3–расм). Бу ерда илова номи, бажарилувчи файл ва каталог номи кўрсатилади.



9.2.3–расм: App Info саҳифаси

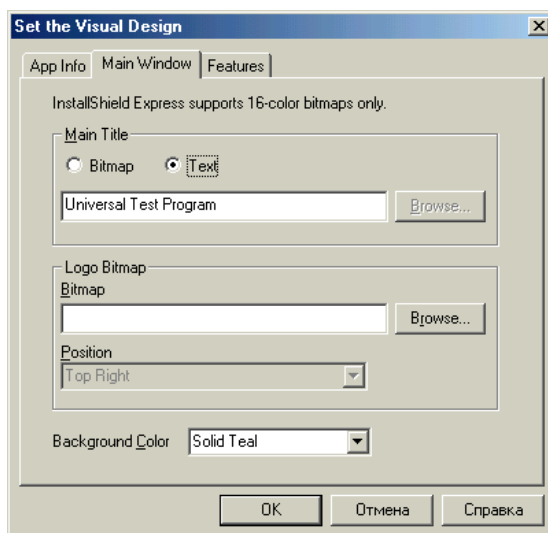
Саҳифанинг баъзи майдонлари автоматик тўлдирилганлигига эътибор қилиш лозим. InstallShield Express дастури илова номи (Application Name) ва илова каталоги номи (Default Destination Directory) сифатида проект номидан фойдаланишни таклиф қилади (дастурчи бошқа ном бериши ҳам мумкин).

Default Destination Directory майдонига эътибор берамиз. Бу майдонда ўрнатилувчи илова каталогининг номи ва унинг ҳолати кўрсатилган.

InstallShield Express дастурида фойдаланувчи Windows каталог кўрсаткичлари қуйидаги жадвалда келтирилган:

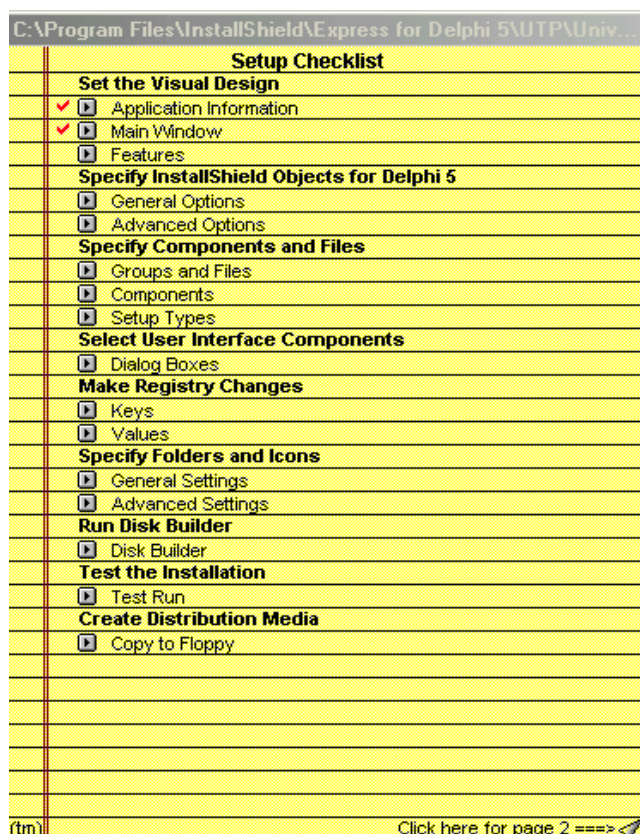
Кўрсаткичлар	Каталог
<WINDIR>	Windows каталоги, масалан, c:\Windows
<WINSYSDIR>	Windowsнинг системаси каталоги, масалан, c:\Windows\System
<ProgramFilesDir>	Дастур каталоги, масалан, c:\Program Files
<INSTALLDIR>	Ўрнатилувчи дастур иловани ўрнатувчи каталог

Main Window саҳифаси ўрнатувчи дастурнинг бош ойнаси характеристикасини бериш имконини беради (9.2.4–расм).



9.2.4–расм: Main Window саҳифаси

Main Title гуруҳи ўрнатилувчи дастур бош ойнасининг сарлавҳаси типини бериш имконини беради. Агар Text танланган бўлса, сарлавҳа ўзида матнни мужассамлаштиради. Агар Bitmap танланган бўлса, илова сифатида 16–рангли расм ишлатилади, файлни Browse тугмаси ёрдамида аниқланади.



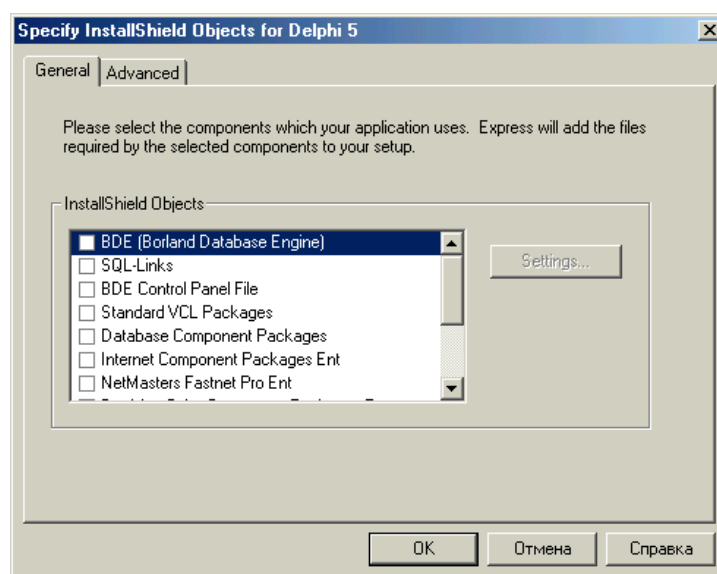
9.2.5–расм: InstallShield Express ойнаси

Logo Bitmap гуруҳида дастурчи ўзининг эмблемасини ўрнатиши мумкин. Бунинг учун Bitmap майдонига эмблема жойлашган файл номи киритилади. Position очилувчи рўйхат ёрдамида эмблема

жойлашувини белгилаш мумкин. Background Color очилувчи рўйхатдан бош ойнанинг рангини танлаш мумкин. ОК тугмаси босилгандан сўнг Set the Visual Design мулоқот ойнаси ёпилади.

InstallShield Express эса фойдаланилган гуруҳ буйруқлари олдига белги қўйиб қўяди (9.2.5–расм).

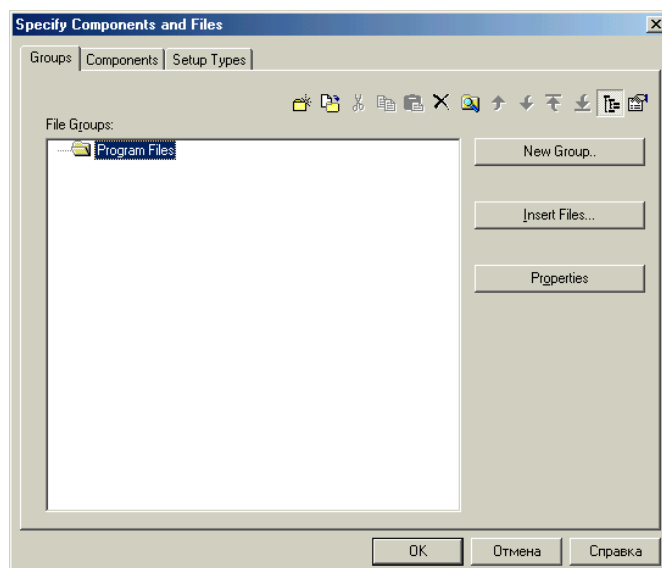
Specify InstallShield Objects for Delphi буйруқлар гуруҳи Delphiнинг қайси объектларини фойдаланувчи компьютерига кўчириш лозимлигини кўрсатиш имконини ҳосил қилади. Масалан, динамик кутубхоналар ёки компонентлар пакети бўлиши мумкин. Ўрнатиловчи дискларга жойлаштириш керак бўлган объектлар InstallShield Objects рўйхатдан танланади (9.2.6–расм).



9.2.6–расм: General саҳифаси

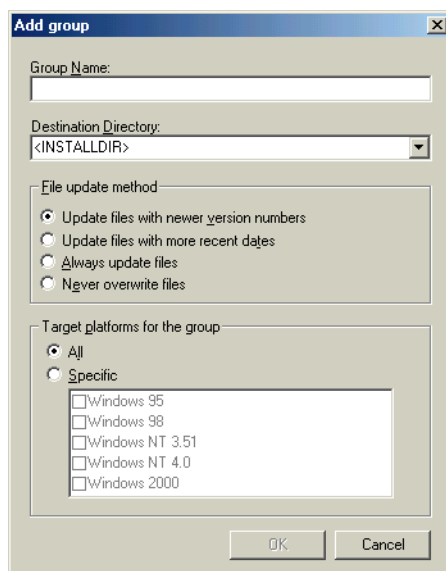
Specify Components and Files гуруҳ буйруқлари фойдаланувчи компютери ҳамда ўрнатувчи дискка кўчирилувчи файлларни юклаш имконини беради.

Эслатма: Катта дастурларни ўрнатишда ўрнатиловчи дастур фойдаланувчидан ўрнатиш типини танлашни сўрайди (тўла, энг кичик ёки танлаш). Танлаш жараёнида фойдаланувчи керакли компонентларни кўрсатиб ўтади. Graph and Files буйруғи бажарилишидан аввал қандай файлларни фойдаланувчи компьютерига кўчириб ўтиш кераклиги кўрсатилади. Дастур файли ва readme.doc файли илованинг бош каталогига, тест файли эса бош каталогнинг Sample ост каталогига жойлаштирилади. Groups and Files буйруғини танлаш натижасида Specify Components and Files мулоқот ойнаси очилади (9.2.7–расм).



9.2.7–расм: Specify Components and Files мулоқот ойнаси

Автоматик ҳосил қилинган Program Files гуруҳи ўрнатиловчи дастурнинг бош каталоги ҳисобланади. Одатда бу гуруҳга дастурнинг бажарилувчи файли, маълумотлар файли ва readme файли жойлаштирилади. Бу гуруҳга янги файлни қўшиш учун Insert Files тугмаси босилади ва ҳосил бўлган файлни очиш стандарт ойнадан керакли файл танланади. “Плюс” белгили квадратни босиш билан гуруҳни очиш ва у ерда қандай файллар мавжудлигини кўриш мумкин.



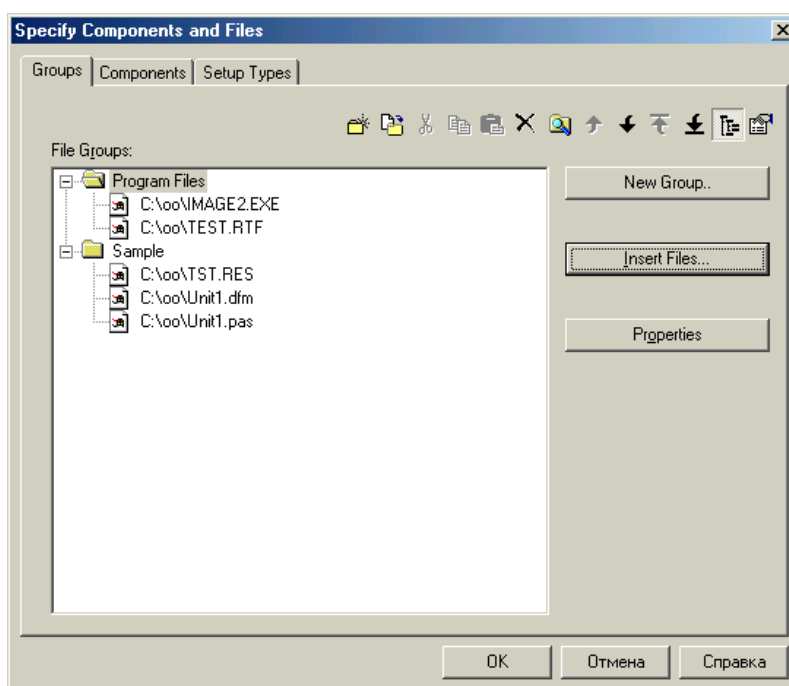
9.2.8–расм: Add Group мулоқот ойнаси

Дастурчи хоҳишига кўра баъзи файлларни алоҳида каталокка ажратиш мумкин. У ҳолда бу файллар учун янги гуруҳ ташкил қилишга тўғри келади. Янги гуруҳни ташкил қилиш учун New Group тугмасидан фойдаланилади. Натижада Add group (9.2.8–расм) мулоқот ойнаси ҳосил бўлади. Group Name майдонига гуруҳ номи,

Destination Directory майдонига эса гуруҳ файллари нусхаси жойлашиши лозим бўлган каталог номи киритилади. Стандарт ҳолатда янги гуруҳ <INSTALLDIR> каталогига эга, яъни илованинг каталоги.

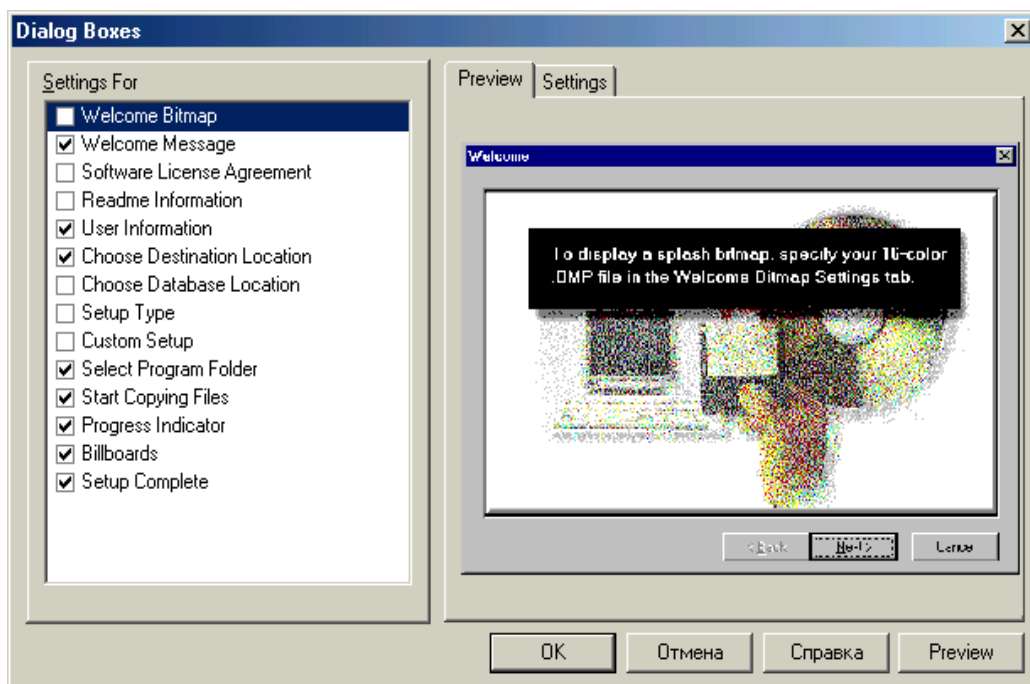
Агар гуруҳ файллари кўчирилувчи янги каталог илованинг бош каталоги ичида яратилса, у ҳолда янги каталогнинг номи <INSTALLDIR> дан сўнг “\” белгиси билан ажратиб ёзилади. Каталог номини очилувчи рўйхатдан танлаб кўрсатилиши ҳам мумкин.

9.2.9–расмда мисол сифатида Specify Components and Files мулоқот ойнасининг Sample гуруҳи ташкил этилган ва файллар гуруҳи танланган ҳолда кўрсатилган.



9.2.8–расм: Фойдаланувчи компьютерига кўчирилувчи файллар

Мулоқотни сошлаш: ўрнатилувчи дастур иши давомида кетма–кет ўзгариб борувчи стандарт мулоқот ойналарни кўришимиз мумкин. Ўрнатилувчи дастурини ташкил этиш билан дастурчи қандай ойналар ҳосил бўлиши ва уларнинг кўринишларини аниқлаши мумкин. Ўрнатилувчи дастур ишга туширилганда экранда ҳосил бўлувчи мулоқот ойнасини ташкил этиш учун Select User Interface Components гуруҳидан Dialog Boxes буйруғини танланади. Ҳосил бўлган мулоқот ойнадаги Settings For рўйхатда ўрнатилувчи дастур иши давомида ҳосил бўлувчи мулоқот ойна номлари берилган (9.2.9–расм).



9.2.9–расм: Dialog Boxes мулоқот ойнаси

9.2.9–расмда келтирилган ойнанинг Settings For келтирилган ойнанинг номлари қуйидаги келтирилган жадвалда берилган.

Muloqot oynasi	Aniqlanishi
Welcome Bitmap	Ўрнатиловчи дастур ҳақидаги маълумотлар чиқарувчи иллюстрация
Welcome Message	Ахборотли маълумотлар чиқариш
Software License Agreement	Файлда жойлашган лицензион маълумотни чиқариш
Readme Information	Ўрнатиловчи дастур ҳақидаги маълумотларни чиқариш
User Information	Фойдаланувчи тўғрисидаги маълумотларни киритиш (исми, фирма номи) ва ўрнатиловчи нусха рақами ҳам бўлиши мумкин
Choose Destination Location	Дастур ўрнатиш учун аниқланган каталогни фойдаланувчи учун ўзгартириш имконини бериш
Setup Type	Ўрнатилаётган дастур типини танлаш имконини бериш (Typical–оддий, Compact–минимал, Custom–танлаш билан)
Custom Setup	Танлаш (Custom) билан ўрнатилаётганда ўрнатиловчи компоненталарни белгилаш
Select Program Folder	Ўрнатиловчи дастурнинг ишга туширувчи буйруғини топириқлар панелининг қайси менюсига қўйишни танлаш имконини бериш

Muloqot oynasi	Aniqlanishi
Progress Indicator	Дастурни ўрнатиш вақтида бажарилган ишнинг фоизини кўрсатиш
Setup Complete	Ўрнатиш тугалланганлиги тўғрисида фойдаланувчини огохлантиради

Ушбу мулоқот ойналари ўрнатиловчи дастур иши давомида ҳосил бўлиши учун мулоқот ойна номининг чап тарафига байроқча ўрнатиш лозим. Мулоқот ойналарининг асосий қисмида параметр сифатида матн ёки *.bmp (16 рангли) файл номи сўралади.

Оддий ҳолларда ўрнатиловчи дастури қуйидаги мулоқот ойналари билан чегараланиши ҳам мумкин:

Readme Information;

Choose Destination Location;

Select Program Folder;

Progress Indicator

Setup Complete.

Назорат саволлари:

1. Ўрнатиловчи дастурлар нима учун керак бўлади?
2. Ўрнатиловчи дастурлар учун керак бўладиган мулоқот ойналарини санаб беринг?
3. Ўрнатиловчи дастурлар учун қандай кўринишдаги расмлардан фойдаланилади?
4. Windows каталог кўрсаткичларини санаб беринг?

Мундарижа

Амалий иш №1: С++ Builder интерфейси ва асосий менюлари.....	5
Амалий иш №2: Standard компоненталар палитраси. Хусусият, услуб ва ходиса.....	10
Амалий иш №3: Бир ва бир нечта форма билан ишлаш. Мулоқот ойналари ва уларнинг турлари.....	22
Амалий иш №4: Additional компоненталар палитраси компоненталари ёрдамида дастурларни лойиҳалаш.....	28
Амалий иш №5: С++ Builder нинг график имкониятлари.....	39
Амалий иш №6: DataAccess, DataControls, dbGo ёки ADO компоненталари палитраси.....	45
Амалий иш №7: Маълумотлар базаси билан ишлаш.....	51
Амалий иш №8: Fast Report ва Rave Report компоненталар палитраси.....	61
Амалий иш №9: Ўрнатиловчи дастурлар яратиш.....	77