

Мэдээллийн Системийн
Менежер-3



GaVaL.net

Delphi

PERSONAL

Разработка приложений для Windows и Linux

Borland Delphi 7.0
оюутны гарын авлага

DELPHI 8
PARA PLATAFORMA .NET
CURSO COMPLETO

2005

The Delphi.
development solution

38
38
38
38

Delphi
7.0

38
38
38
38

Монгол Улсын Их Сургууль

Математик Компьютерийн сургууль

Мэдээллийн системийн тэнхим

Delphi 7.0

Анхлан сурагчдад зориулсан гарын авлага

Улаанбаатар хот
2006 он

Өмнөх үг

Бид өнөөдөр програм бичнэ гэж бодвол их хүнд, болж бүтэхгүй, чадахгүй мэтээр төсөөлдөг. Мэдээж програм бичнэ, програм зохионо гэдэг бол хэн хүнссэн нь хийгээд байдаг хялбар ажил, хялбар үйлдэл биш бөгөөд таниас багагүй мэдлэг, нүр их хөдөлмөр, сэтгэлгээ шаарддаг нь нууц биш ээ. Компьютер технологи хөгжихийн хэрээр хүн бүхнээс компьютер, програмж хангамжийн мэдлэг шаардах болсон. Өнөөдөр бидний аав, ээж би гээд л хүн бүхэн л машин жолоожод чаддаг. Өөрөөр хэлбэл бид бүгд жолооч. Тэгвэл гаадаадын өндөр хөгжилтэй орнуудад програмист гэдэг бол энгийн хэрэглээ, хэрэгцээ, мэдлэг болсон нь бид бүхэн машин барьж чаддагтай ажилхан хүн бүхний зайлшгүй эзэмшсэн мэргэжил болсон байна. Гэхдээ бид өнөөдөр мэдээж програмист хүмүүс шиг өндөр түвшинд програм бичиж чадахгүй нь мэдээж. Тэгвэл визуал програмчлал нь бидэнд ийм боломжийг олгодог. Визуал програмчлал бэлэн компонент, хялбар хэрэглээний програм ашиглан өөрийн хэрэгцээг хангасан програм бичих боломжийг бидэнд олгодог.

Энэхүү ном нь “Визуал програмчлал-1” хичээлийн программын дагуу зохиогдсон бөгөөд семнарын ажиллуудаас үндэслэгдсэн болно. Анхлан сурагч, мэргэжлийн бус ангийн оюутнуудад мөн бидний аав, ээж зэрэг компьютер ашигладаг хүмүүст ашиглах материал, зөвлөмж ховор байдгыг харгалзан үзэж энэхүү гарын авлагыг бүтээлээ.

Би бол оюутан тиймээс үс үсгийн алдаанаас авахуулаад олон зүйл дээр эндэж алдсан байж болох тул эрхэм уншигч та гэгээн оюундаа тунгаан болгооно уу. Нуулгүй хэлэхэд би өөрийгөө зохиогч гэж хэл болох эсэхээч сайн мэдэхгүй л байна.

Зохиогч

Гарчиг

1-р бүлэг. Үндсэн ойлголт

- 1.1. RAD (Rapid Application Development)
- 1.2. Компонент (Visual Component Library)
- 1.3. Delphi IDE (Integrated Development Environment)
- 1.4. Delphi- д ашиглагддаг файлуудын төрлүүд
- 1.5. Help ашиглах

2-р бүлэг. Түгээмэл ашиглагддаг шинж, үзэгдэл, дүрмүүд

- 2.1. Түгээмэл ашиглагддаг шинжүүд
- 2.2. Түгээмэл ашиглагддаг үзэгдлүүд
- 2.3. Түгээмэл ашиглагддаг дүрмүүд

3-р бүлэг. Түгээмэл ашиглагддаг компонентууд

- 3.1. Standart tabs
- 3.2. VCL график компонентууд

Лабораторын ажил

- User хэрэглэгчийн admin эрхийн цонх
- Тооны машин
- ListBox, ComboBox ашиглах нь
- Notepad – г хийх

4-р бүлэг. График, мультимедиатай ажиллах

- 4.1. TCanvas класс
- 4.2. Canvas дээр дүрс зурах
- 4.3. Canvas дээр текст зурах
- 4.4. PlaySound API функц
- 4.5. TAnimation компонент
- 4.6. TMediaplayer компонент

Лабораторын ажил:

- $ax^2 + bx + c = 0$ функцийн график
- MediaPlayer
- Миний санасан тоо хэд вэ?

5-р бүлэг. Өгөгдлийн сантай ажиллах

- 5.1. Мэдээллийн сангийн тухай ойлголт
- 5.2. ODBC Data source үүсгэх
- 5.3. Мэдээллийн сангийн компонентууд
- 5.4. Тайлан Rave5.0

Лабораторын ажил:

- Dictionary хийх
- Банкны хадгаламжийн хүүгийн харицуулдаг програм

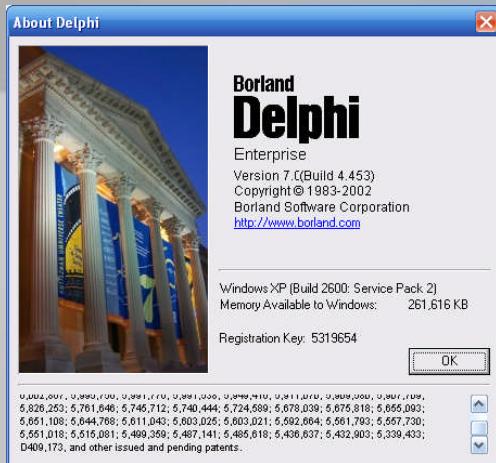


1-р бүлэг
Үйлдсэн ойлголт

RAD (Rapid Application Development)

Програмыг хялбар аргаар боловсруулах

Програмыг хялбар аргаар боловсруулах, олон цонхны горимд ажиллуулахыг тулд RAD технологи хөгжиж ирсэн. RAD – ыг хөгжүүлэгч хэл нь визуал хэл юм. Уламжлалт буюу процедур хандалт Borland Pascal дээр тулгуурласан ОХП буюу Object Pascal програм гарч ирсэн. Тэгвэл Borland фирм нь Object Pascal хэл дээр тулгуурласан визуал програмын Delphi хэл 1994 онд анх гарсан. 2004 оны сүүлчээр Delphi 9.0 буюу Delphi 2005 програм гарсан. Delphi 7.0 нь 2002 онд гарсан байна. Delphi-г ашиглан windows орчны бүрэн хэмжээний програм бичих боломжтой. Delphi дээр програм бичихийн тулд



шаардлагатай. (Runtime)

1. Бэлэн байгаа компонентуудыг угсрах (Design time)
2. Object Pascal хэл дээр кодыг бичих

VCL (Visual Component Library)

Компонент нь дизайн DesignTime-ийн үед удирдаж боловсруулах боломжтой визуал объект юм. Компонент бүр нь ямар нэгэн TComponent class-с удамшиж үүсдэг. Хэдийгээр компонент бүр нь класс боловч класс бүр нь компонент болж чадахгүй Компонентийг RunTime-ийн үед харагдах байдлаар нь

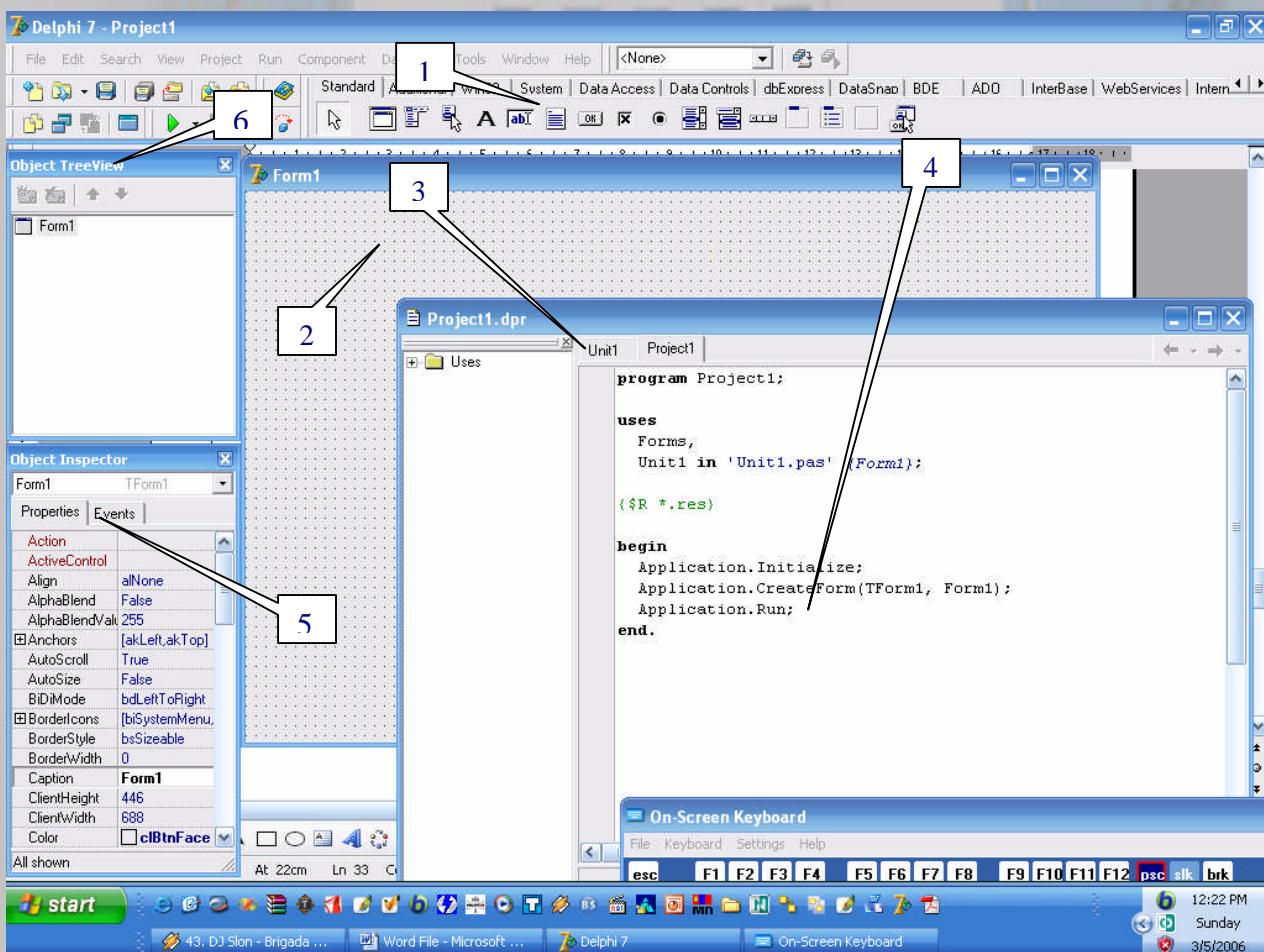
1. Визуал
2. Визуал биш гэж ангилдаг.

DesignTime, RunTime-ийн үед аль алинд нь харагдаж байдаг компонентууд нь визуал компонент болно. Delphi дээр програм бичихийн тулд VCL санд тодорхойлсон бэлэн компонентуудаас авч ашиглана. VCL – д орши тухайн компонентууд нь олон тооны үзэгдэл буюу Event, шинж буюу properties, хувьсагчийн өнгө солих дүрэм буюу method-с бүрдэнэ.

Delphi IDE (Integrated Development Environment)

Delphi 7.0-ын IDE (Integrated Development Environment – програмыг бүрдүүлэгч, үзэгдэх орчин) нь үдсэн 6 цонхиос бүрддэг.

1. Component Palette
2. Design Windows
3. Code Window
4. Project Window
5. Projects Object Inspector
6. Object TreeView



1. Component Palette: VCL-с ашиглаж болох бэлэн компонентууд. (товчлуур, цонх, меню... гх)
2. Design Window: Компонентуудыг угсарч програмын харагдах байдал, дизайныг зохиох талбар
3. Code Window: Угсарсан компонентуудын цаана ямар үйлдэл гүйцэтгэх вэ? Гэдгийг объект паскал хэлбээр бичиж өгөх талбар.
4. Project Window: Тухайн программын project-ын бүтцийг харуулна.

5. Properties Object Inspector: Компонентийн шинж чанар (Properties), үзэгдэлийг (Event) энд байрлуулсан байна.

Delphi-д ашиглагдах файлын төрлүүд

Delphi програм дээр програм зохиоход хадгалах, хуулах, зөөх хэрэг гардаг. Delphi файлуудын төрөл болон зориулалтыг өргөтгөлөөр нь таньж болно. Эдгээр файлын зориулалтыг дараах хүснэгтээр үзүүлэв.

Өргөтгөл	Тайлбар
.pas	Delphi, паскал програмын код бичсэн файл
.dfm, xfm	Форм файл. Форм нэмэхэд үүсдэг файл
.dot, .kof	Энэ файл нь Project Options харилцах цонхонд тодорхойлсон прожектийн харицах сонголтуудыг хадгалж байдаг
.exe	Програмыг complete хийсний үр дүнд үүсэх програмын шууд ажиллуулах файл
.cfg	Энэ файл нь проектод хийсэн тохиргоог хадгалж үлддэг.
.dcu	Complete хийсний дараа үүсэх binary объект файл
.dpr	Delphi програмыг бүрдүүлэгч проект файл
.bpg	Энэ файл нь хэд хэдэн пројек файлаар бүлэг үүсгэдэг
.res, .dcr	Энэ файл нь програмд ашиглаж байгаа віртар, icon зэрэг өгөгдлийг хадгалж байдаг файл
.dpk	Програмын болон компонентын код бичсэн модуль файлуудыг нэгсэн файл
.bpl	Энэ файл нь Delphi-ийн онцлог шинж чанарыг агуулсан DLL файл юм.
.dcp	Пакетийн толгой хэсэг болон dcs файлуудын нийлүүлж багцалсан нөөц файл
.ddp	Компонентуудын харьцааг үзүүлсэн диаграм файл
.dsk	Энэ файл нь Delphi-ийн IDE дээр үзэгдэж байсан орчин хадгалж үлддэг файл

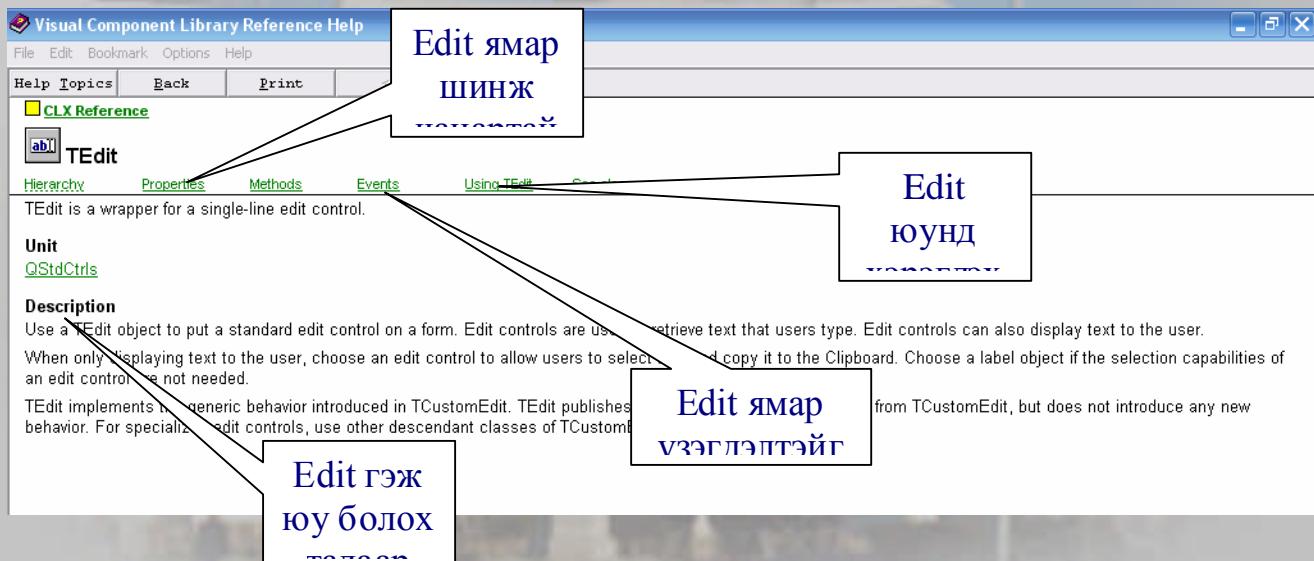
Хагийн жижижхэн гар

АНУ-ын иргэн Дэвид Леви гэгч дэлхийн хамгийн
жижижхэн гарыг бүтээж, 1997 оны гуравдугаар сарын 18-
нд Гиннесийн номонд оржээ. Энэхүү гар нь 7,62x3 см
хэмжээтэй бөгөөд цагаан толгойн үсгүүд цифр болон бусад

Help ашиглах

Ихэнх програмд F1 эсвэл Help зэсний командыг дуудаж тухайн программын тусламжийн файлыг ажиллуудаг. Энэхүү файл нь тухайн програмыг бүтэц, хэрхэн ажилдаг болон паскал кодийн жишээг ч бас агуулсан байх тул түүнийг тогтмол ашиглаж сурх хэрэгтэй. Жишээ нь Edit компонентийн талаар тусламж хүсье.

1. Форм дээр байрлах Edit1 компонентийг идэвхижүүлээд F1 товчлуур дарна.
2. Үүсэх Topics Found цонхноос Display товчлуурыг дарна.



3. Үүсэх Visual Component Library Reference Help харилцах цонхоонс өөрийн хүссэн мэдээллээ авч болно.
4. Ногоон бичгэнүүд дээр үргэлжлүүлэн дарж цааш нь илүү дэлгэрэнгүй мэдээлэл, жишээ кодыг авах боломжтой.

Хамгийн жижижхэн хулгана

Дэлхийн хамгийн жижижхэн хулганыг 2000 онд Тайванд бүтээжээ. Ердийн хулганатай харицуулахад жижиг бөмбөг мэт санагдах CatEyeFinRing гэгч энэхүү бяцхан хулгана ердөө л 4,7 см хэмжээтэй. Даширамд



2-р бүлэг.

**Түгээмэл ашиглагддаг
шинж, үзэгдэл, дүрмүүд**

Delphi 7.0 програмын VCL сан нь 420 гаруй компонентуудтай байдаг ба компонент бүр нь 20-25 шинж чанар, үзэгдэл, функцуудтай. Тийм болохоор эдгээр компонентуудыг нэг бүрчлэн үзэх боломжгүй. Харин үүний оронд VCL компонентуудад нийтлэг байдаг шинж чанар, үзэгдэл функцуудыг авч үздэг. Borland фирм Delphi програмыг жил бүр сайжруулан шинэ хувилбаруудыг гаргахдаа компонентуудыг нэмсэн байдаг байна.



Түгээмэл ашиглагддаг шинжүүд

Сомпонентийн шинжүүд нь тухайн компонентийн байрлал хэмжээ, өнгө, текст, фонд гэх мэт мэдээллүүдийг хадгалж байдаг. Шинжүүдийг дотор нь

1. DesignTime-ийн үед Object Inspector цонхонд өөрчилж болдог
 2. Зөвхөн програм дотроос өөрчилж болдог
- хэмээн хоёр ангилна.

Name – тухайн компонентийн програмд харагдах нэр. Онцгой шаардлагагүй бол өөрчлөх хэрэггүй.

Anchors - Тухайн компонент формтой цуг дагаж томорч, жижигрэх эсэх тохируулга хийнэ.

Caption, Text – Тухайн компонент дээр бичигдэх бичгийг энд бичиж өгнө.

Color - Тухайн компонентийн өнгийн тодорхойлно.

BorderStyle – Тухайн компонентийн хүрээний хэлбэрийг тодорхойлно.

Fond – тухайн компонент дээр бичигдэх үсгийн фондыг тодорхойлно.

Height - компонентийн өндөр

Width – компонентийн өргөн

TabOrder – tab дарах товчлууруудын дарааллийг зааж өгнө.

Cursor – тухайн компонент дээр хулганы заагч ямар хэлбэртэй болохыг зааж өгнө.

DragCursor – компонентийг чирэх үед курсор ямар хэлбэртэй болохыг зааж өгнө.

PopUpMenu – тухайн компонент дээр хулганы баруун товчлуур дээр дарах үед үүсэх зэстэй холбоно.

Visible – компонентийг үзэгдэх эсэхийг тодорхойлно.

Hint – компонент дээр хулганы заагчийг байрлуулахад үзэгдэх мэдээллийг тодорхойлно.

Түгээмэл ашиглагддаг үзэгдлүүд

- ❖ Удирдлага хүлээн авах, алдах үеийн үзэгдлүүд
 - *OnEnter, OnActivate:* Уг компонентэд удирдлага шилжиж ирэх үеийн үзэгдэл
 - *OnExit, OnDeActivate:* Уг компонентоос удирдлага өөр компонентэд шилжүүлэх үеийн үзэгдэл юм.
- ❖ Гарны товчоор гүйцэтгэх үзэгдлүүд
 - *OnKeyDown* – гар дээрээс товч дарах үеийн үзэгдэл
 - *OnKeyPress* – Товч дарсан үеийн үзэгдэл
 - *OnKeyUp* – Товч дараад авах үеийн үзэгдэл
 - *OnChange* – компонент дээр бичиглэл хийх үеийн үзэгдэл
- ❖ Хулганы товч дарах үеийн үзэгдлүүд
 - *OnClick* – Компонент дээр хулганаар дарах үеийн үзэгдэл (зүүн)
 - *OnDoubleClick* – компонент дээр хоёр дарах үеийн үзэгдэл
 - *OnMouseDown* – компонент дээр хулганы дарах үеийн үзэгдэл (баруун/зүүн)
 - *OnMouseUp* – хулганы товч дараад авах үеийн үзэгдэл
- ❖ Системийн үйл ажиллагаанд үүсдэг үзэгдлүүд
 - *OnCreate* – компонентийн санах ойд үүсэх үеийн үзэгдэл
 - *OnPaint* – компонент зурах үеийн үзэгдэл
 - *OnTimer* – нэгж хугацаа өнгөрөхөд хийх үйлдлийг тодорхойлно

Түгээмэл ашиглагддаг дүрмүүд

- *Hide* – тухайн компонентийг харагдахгүй болгох дүрэм
- *Show* – тухай компонентийг харагдуулах дүрэм
- *Destroy* – тухайн компонентийг санах ойгоос устгагч функц
- *ClientToScreen* – client талбайгаас өгөгдсөн цэгийг дэлгэцний кординатад хөрвүүлнэ.
- *Free* – компонентийн ашигласан санах ойг чөлөөлнө
- *Perform* – компонентруу мэдээ дамжуулдаг
- *Repaint* – уг компонентийг дахин дуудна

Хамгийн өндөр харайсан Робот

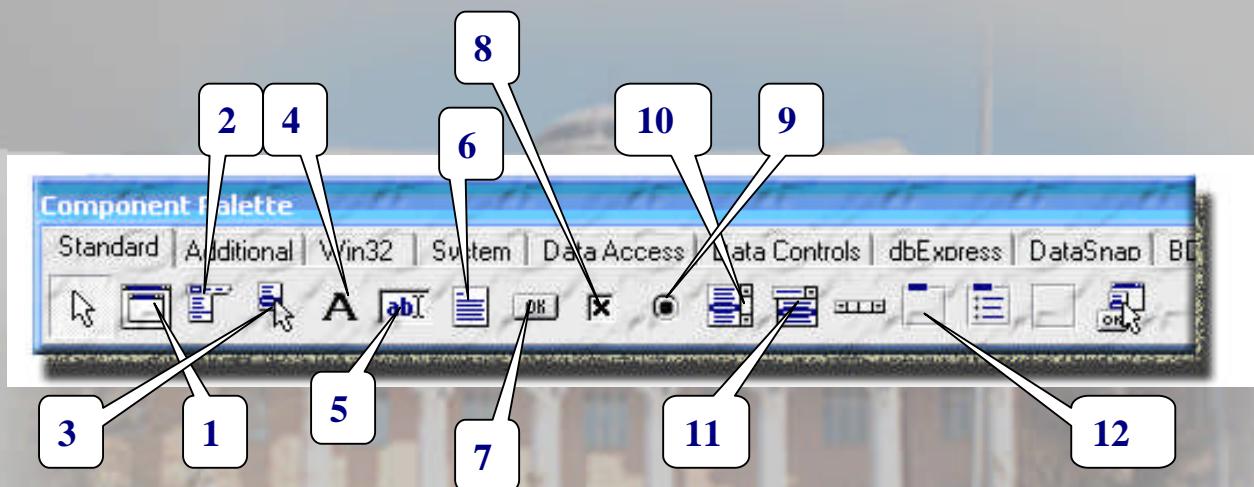
АНУ-ын Sandia лабораторийн бүтээсэн Hopper робот нь 2001 оны гуравдугаар сард 9метр харайсан амжилттай Гиннесийн номонд оржээ



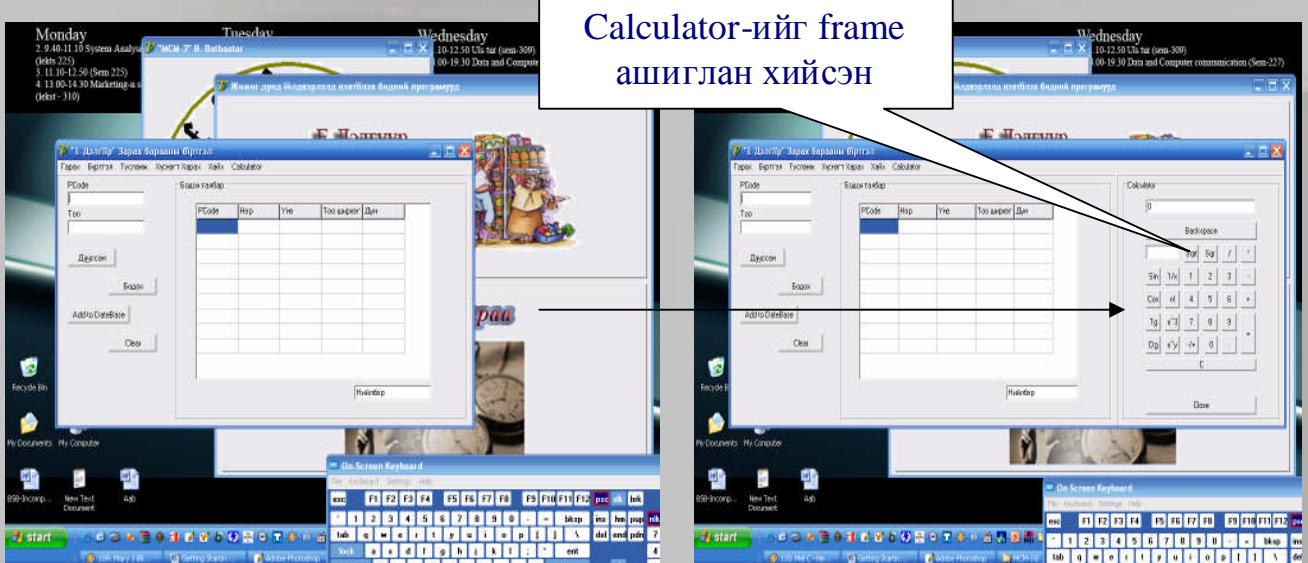
З-р бүлэг.
Түгээмэл ашиглагддаг
компонентууд

Standart tabs

Standard Component Palette-д бидэн хэрэглэгдэх стандарт товчуур, хэрэглэлүүд байдаг байна.

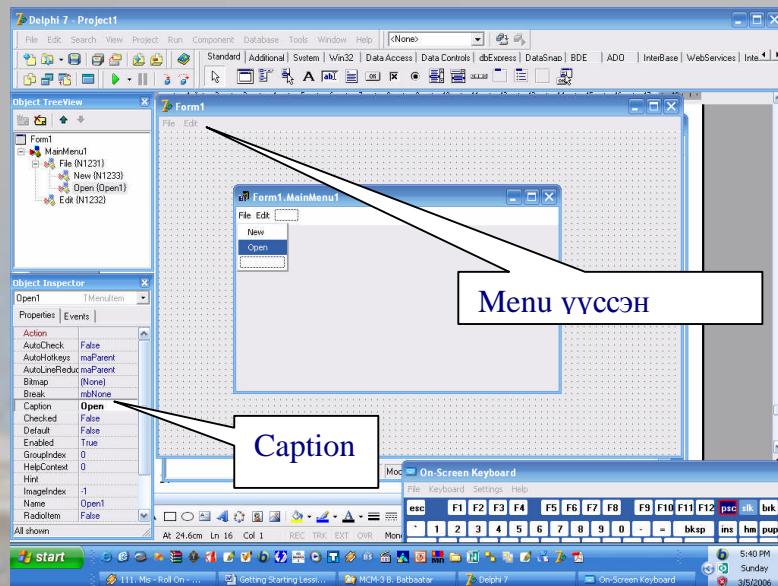


1. TFrame: Бүлэг товлууруудыг үүсгэхэд зориулагдсан компонент юм. Түүнтэй ажиллах нь формтай ажиллахтай ерөнхийдөө төстэй. Гэхдээ форм шиг show үйлдлээр дуудах боломжгүй бөгөөд форм дээр Frame компонентийг байрлуулан холболт хийн дуудна.



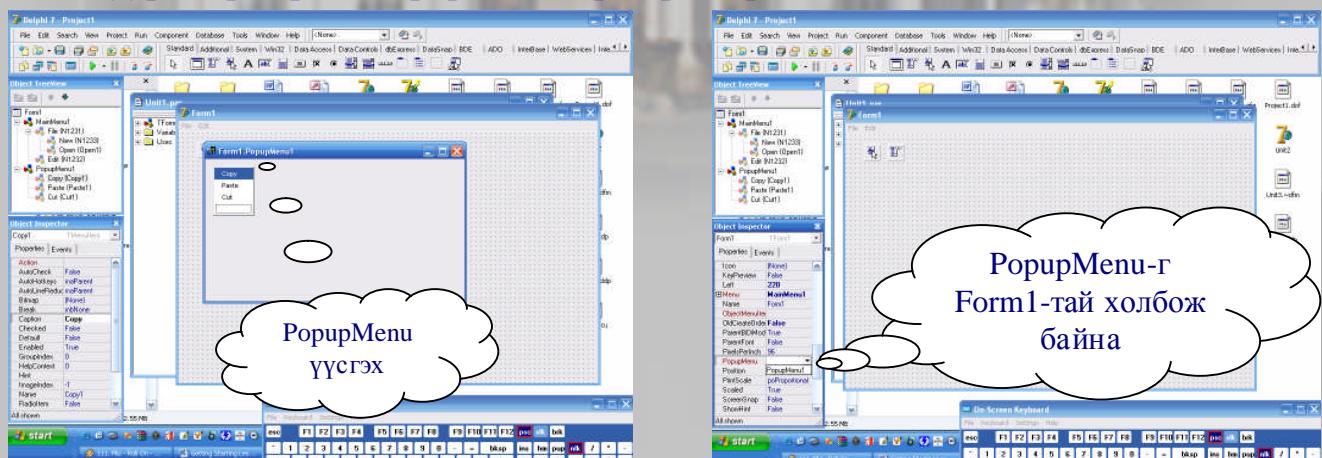
2. MainMenu: Програмын меню-ыг хийдэг компононт. Меню-г хийх

- a. MainMenu компонентийг форм дээр байрлуулна
- b. Форм дээр байрлах MainMenu компонент дээр 2 дарна

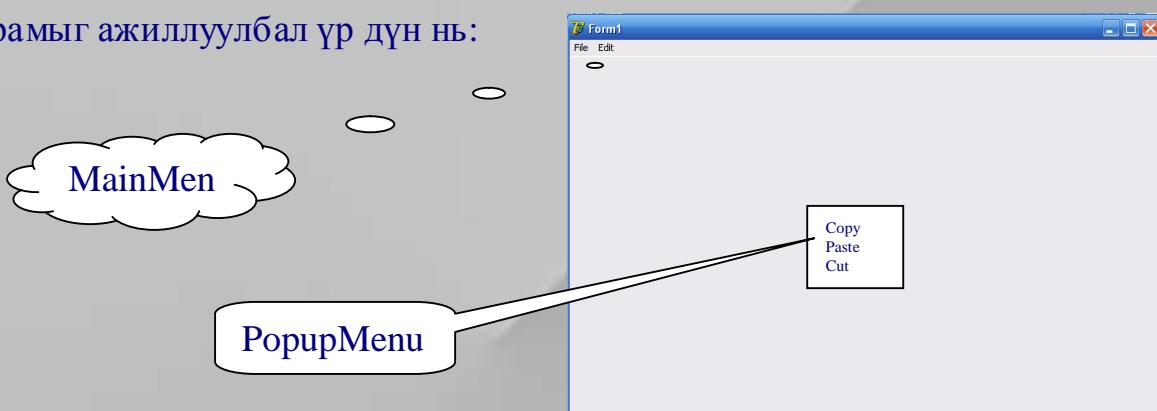


- c. Form1.MainMenu цонхонд өөрийн меню-г хийх ба caption шинжид menu-ний нэрийг зааж өгнө.

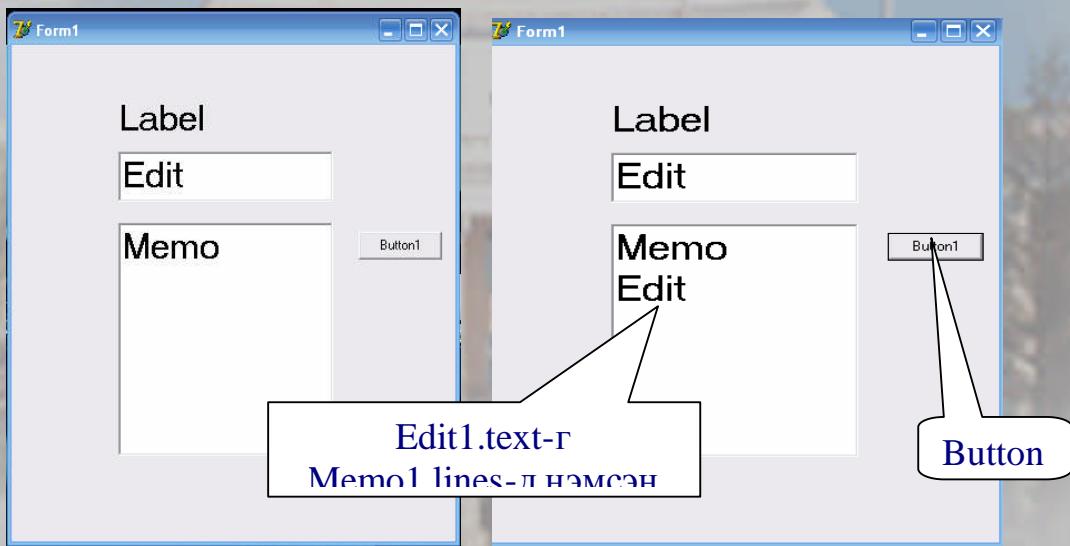
3. PopupMenu: Үүнийг үүсгэх нь MainMenu-тай адилхан. Бид хулганы баруун товчуур дээр дарвал PopupMenu гарч ирэхээр холбож өгдөг.



Програмыг ажиллуулбал үр дүн нь:



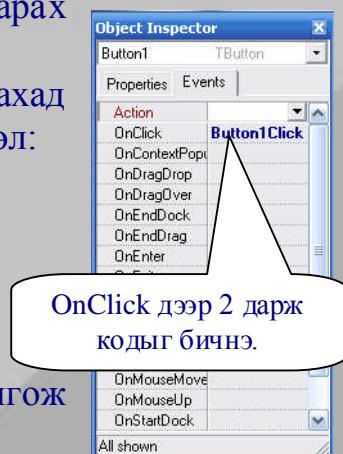
4. Label: Форм дээр тогтмол текст мэдээллийг үзүүлэхэд ашигладаг визуал компонент юм. Label дээр гарах текстийг Properties-ийн caption-д бичиж өгнө.
5. Edit: Програм ажиллаж байх үед хэрэглэгч мэдээллийг бичиж оруулахад зориулагдсан компонент. Хэрэглэгчийн оруулсан бичлэгийг Properties-ийн text-д хадгалсан байдаг.
6. Memo: Edit-тай ижилхэн үүрэгтэй бөгөөд гэхдээ илүү урт, мөртэйгээр бичлэгийг авдаг байна. Ашиглагдах гол шинжүүд нь:
 - a. Lines: String төрөлтөй хувьсагч бөгөөд Memo-d бичсэн текст-тэй ажилладаг. Жишээлбэл: Memo1.Lines.Add(Edit1.text);



7. Button: Энэ нь Delphi програмын товчлуур юм. Гол шинжүүд ба үзэгдлүүд:

 - a. Caption-д тухайн товчлуур дээр гарах бичиглэлийг хийдэг
 - b. OnClick үзэгдэл тухайн товчлуур дээр дараахад ямар үйлдэл хийхийг бичиж өгнө. Жишээлбэл:

```
Procedure TForm1.Button1Click(Sender: TObject);
begin
  Memo1.Lines.Add(Edit1.Text);
end;
```



8. CheckBox, 9. RadioBox нь гол үүрэг нь сонголтыг олгож байдаг. Гол үзэгдэл нь:
 - a. Checked: дүрэм юм. Энэ дүрэм нь false эсвэл true гэсэн Boolean төрлийн утгыг өгдөг. Өөрөөр хэлбэл Сонгосон (Checked=true), сонгоогүй (Checked=false) гэсэн утгыг өгдөг
10. Combobox, 11.ListBox компонентууд нь цэсэнд өгсөн утгуудаас сонголт хийх зориулалттай байдаг. Ашиглагддаг гол шинж нь
 - a. Items: Энэ нь нэг талаараа Memo1.Lines-тай адил үүрэгтэй буюу Combobox, ListBox-ын утгатай ажиллах үүргийг гүйцэтгэдэг.
12. GroupBox: Энэ компонент нь форм дээр компонентуудын бүлгийг үүсгэдэг.

VCL график компонентууд



TShape компонент

TShape компонентийг форм дээр дүрс буюу дугуй, эллипс, тэгш өнцөгт, квадрат гэх мэт олон дүрсийг зурахад ашигладаг.

Эдээр дүрсийг зурах болон TShape компонентийн зурахад түгээмэл хэрэглэгддэг дүрм нь:

Brush: Тухайн дүрсийн background-ын өнгийг тодорхойлж өгнө.

Pen: Дүрсийг зурж буй үзэгний шинж чанарыг (өнгө, дүрс... гм) тодорхойлно.

Shape: Ямар дүрс зурахыг сонгоно.

stCircle – дугуй

stRoundRect – үзүүр нь дугараг тэг өнцөгт

stEllipse – эллипс

stRectangle – тэгш өнцөгт гэх мэт олон

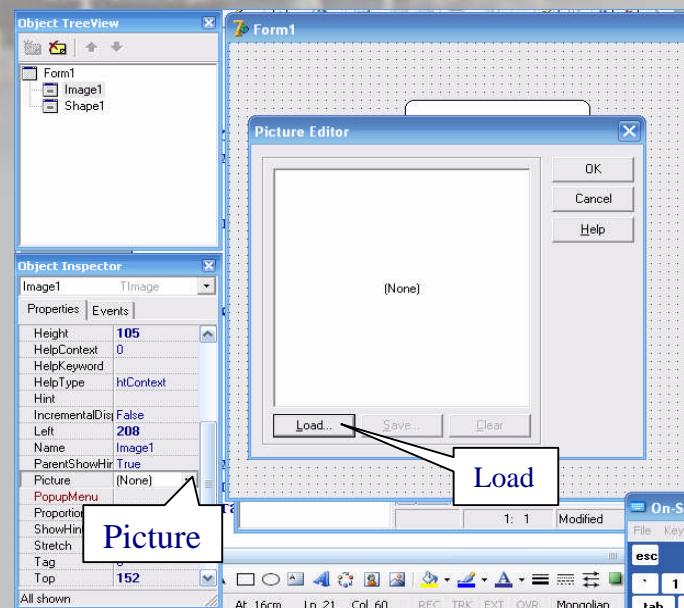
сонголттой.



TImage компонент

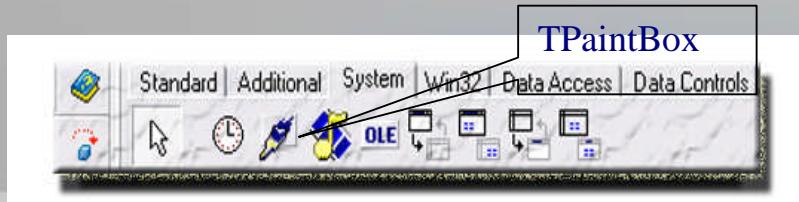
TImage компонент нь форм дээр зураг оруулах зориулалттай компонент юм. Бидний програмд зургийн

файлуудыг оруулж өгвөл програмын дизайн маань илүү гоё болох болно. Энэ компонентод зургийг оруулж өгөх гол шинж нь object inspector цонхны Picture шинжийг сонгох дараад Picture Editor харилцах цонх үүсэх ба load... талбарт зургийн компьютерт байгаа газрыг зааж өгөхөд таны зураг холбогдон таны формд гарч ирнэ.



TPaintBox компонент

TPaintBox компонент нь формын тодорхой нэг хэсэг, талбайд зураг зурах үед ашиглагдана. PaintBox компонентийн хамгийн чухал функц нь бол Canvas функц юм. PaintBox дээр зураг зурахдаа PaintBox-ын object Inspector цонхны event буюу зүгдлүүдээс onpaint үзэгдэлд Canvas функцыг ашиглан зурна.



Жишээлбэл: PaintBox дээр тэгш өнцөгт зуран, түүн дээр 'Rectangle' гэж бичье.

```
procedure TForm1.PaintBox1Paint(Sender: TObject);
begin
Paintbox1.Canvas.Rectangle(10,10,80,80);
Paintbox1.Canvas.TextOut(15,15,'Rectangle');
end;
```

TTimer

TTimer компонент нь програмд хугацааны тоолуур болгон визуал бус компонент юм. Timer нь дараах онцлог шинж чанартай.

Enabled: хугацааны тоолуур идэвхитэй байх эсэхийг тодорхойлно.

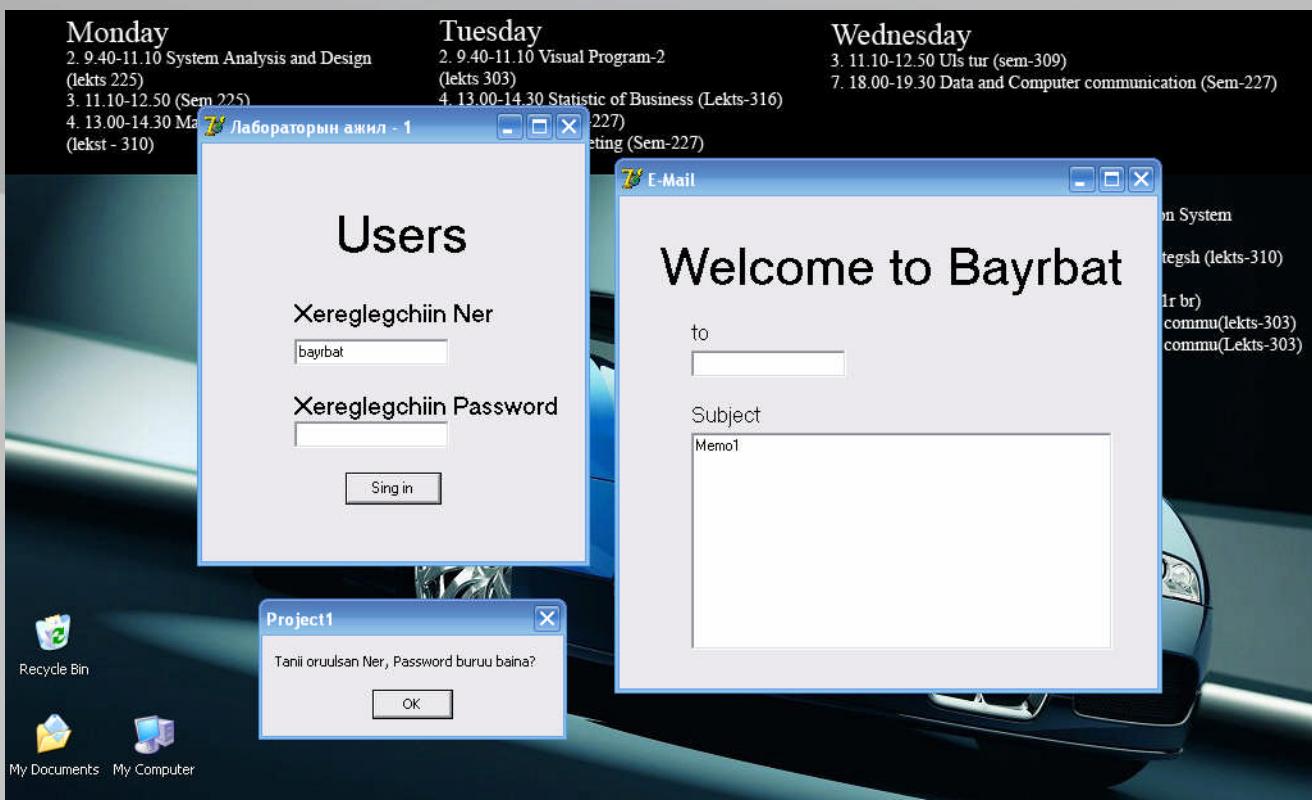
Interval: Timer компонентийн нэгж хугацааг тодорхойлно. Default утга нь 1 секунд байдаг.

OnTimer: Энэ үзэгдлээр timer компонент нэгж хугацаанд юу гүйцэтгэхийг тодорхойлно.

Лабораторын ажил

Сэдэв: User хэрэглэгчийн admin
эрхийн цонх

Users цонхны ерөнхий бүтэц, дүр зураг



Цонхны ажиллах горим:

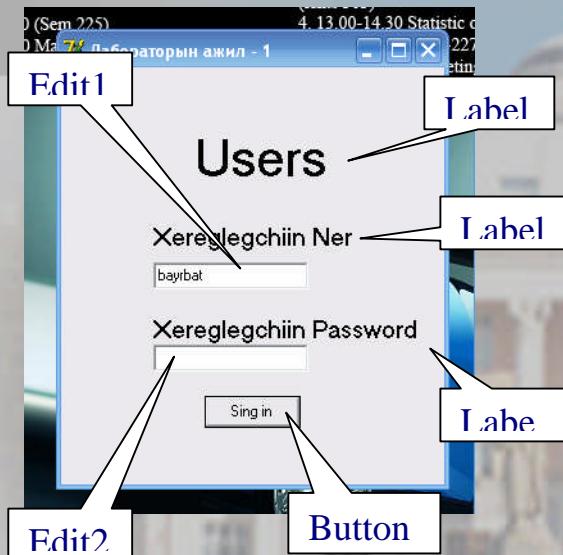
Бид өдөр болгон майлаа шалгадаг. Энэ цонх нь тэгвэл яаг майл шалгахтай адилхан. Чи хэрэглэгчийн нэр талбарт өөрийн програмд оруулж өгсөн нэрийг, паспорт талбарт нь програмд оруулж өгсөн паспортаа оруулаад Sing in товчлуурыг дарна. Хэрэв чи зөв оруулсан бол Е-Майл цонх нээгдэнэ. Хэрэв буруу оруулбал ‘Таны оруулсан Нэр, паспорт буруу байна’ гэсэн бичлэг гарч ирнэ.

Цонхыг хийх

1. Form1 дээр Standart tabs-с нэг button, хоёр Edit, турван Label авч тавин байршуулна.
2. Label1-ийг сонгон Object Inspector цонхны Properties талбарыг сонгоно. Дараа нь уг талбарын Caption шинжид “Users” бичиж өгөх ба Font->Size-д 30 хэмээн зааж өгнө.
3. Тэгвэл label2-ийн caption-нд ‘Xereglegchiin Ner’, label3-ийн caption-нд ‘Password’, button1-ийн caption-нд ‘Sing in’ гэж тус тус бичиж өгнө.
4. Edit1, Edit2-ийн object Inspector цонхны properties->text-с бичгийг арилгах ба Edit2 – ийн properties->PasswordChar шинжид *-ыг бичиж өгнө. Энэ чи

паспортыг оруулж байхад чамайг ямар паспорт оруулж байгаас үл хамааран чиний паспортыг нууцалж *-ын цуваагаар нүдэнд харуулдаг.

5. File/New/From хэмээн сонгож Form2- ийг үүсгэнэ.
6. Form2-ийн Caption-нд ‘E-Mail’ гэж бичиж өгөх ба Form2 дээр label, edit, memo1-ийг тавьж харгалзах нэрийг нь солих бичгийг нь арилгах үйлдлүүдийг хийнэ.
7. Button1 товчлуур дээр хоёр дараход TForm1.Button1Click(Sender: TObject); процедур үүсэх ба дараах томъёог бичиж өгнө.



```
procedure TForm1.Button1Click(Sender: TObject);
begin
if (edit1.Text='bayrbat') and
(edit2.Text='123456')
then
form2.Show
else
showmessage('Tanii oruulsan Ner,
Password buruu baina?');

end;
```

тайлбар: Бид програмыг ажиллуулахад харгалзах өгөгдлийг edit1, edit2-т бичиж өгөх ба edit1- нь Баярбат гэж бичсэн байна уу, Edit2 нь 123456 гэж паспортыг зөв оруулсан байна уу гэж шалгана. Хэрэв Баярбат, 123456 хэмээн өгөгдлийг зөв оруулбал form2 нь show буюу харагдана(Нээгдэнэ) гэсэн үг. Харин өгөгдлийг ямар нэгэн байдлаар буруу оруулбал ‘Tanii oruulsan Ner, Password Buruu Bain?’ гэсэн мсж-ийг бидэнд харуулна. Showmessage – нь дэлгэцэнд мэдээлэл харуулдаг процедур юм.

8. F9 товчлуурыг дарна
9. Үүсэх ‘Yes, No, Cancel’ сонголттой цонх нь Form1-ийг Form2-той холбох уу гэж асууж байгаа цонх тул ‘Yes’ сонголтыг даран дахин F9 товчлуурыг даран програмыг ажиллуулна.
10. Нэмж хэлэхэд 7-р үйлдэл дээр Баярбат гэсэн нэрийн оронд өөрийн нэрийг, 123456 тооны оронд өөрийн паспортыг оруулж өгч дахин програмыг ажиллуулж үзээрэй болно.

Хамгийн их дархсан компьютерийн вирус

Гиннесийн номонд дэлхий даяар тархаж, нийт 3,1 сая компьютерт халдварласан амжилтаар 2000 оны 5-р сарын 1-нд I Love You вирус опсон байна

Лабораторын ажил

Сэдэв: Calculator

Тооны машины бүтэц



Calculator хийхэд ашиглагдах хувьсагчуудаг зарлах.

```
a, b, c: real;  
m: real;  
i: byte;  
str: string;
```

Тайлбар: Програм ажиллах үед бид Edit1-д тоог оруулж өгдөг. Тэгвэл Edit1-д бичсэн утга нь String төрөлтэй байдаг учир бодолтыг хялбар болгох

```
Unit1 Unit2  
  
procedure Button27Click(Sender: TObject);  
procedure Black1Click(Sender: TObject);  
procedure Exit2Click(Sender: TObject);  
procedure Copy2Click(Sender: TObject);  
procedure Paste2Click(Sender: TObject);  
procedure AboutCalculator1Click(Sender: TObject);  
procedure Button29Click(Sender: TObject);  
  
private  
  { Private declarations }  
public  
  { Public declarations }  
end;  
  
var  
  Form1: TForm1;  
  a,b,c:real;  
  m:real;  
  i:byte;  
  str:string;  
  
implementation  
  
uses Unit2;  
  
{$R *.dfm}  
procedure TForm1.About1Click(Sender: TObject);  
begin  
  Form2.show;  
  form2.label1.caption:='MCM-3 Batbaa';  
end.  
  
On-Screen Keyboard
```

Тайлбар: Бид нэмэх болон хасах үйлдлэл хийхийн тулд нэмэх товчуур дарахаас өмнөх бичсэн утгыг, нэмэх тэмдэг дарснаас хойш бичигдсэн утгуудыг нэмж нэмэх үйлдлийг гүйцэтгэнэ. Тэгвэл нэмэх тэмдэг дарахаас өмгөх утгыг а хувьсагч, нэхэмэх тэмдэг дарсны дараах утгыг б гэсэн хувьсагчид хадгалах үүднээс а ба б хувьсагчийг орууж өгнө. Бидний бас нэгэн гол хувьсагч бол str хувьсагч юм. Яагаад тоо бодох ёстой байдал яагаа мөр төрлийн хувьсагч авах болов гэж надаас асууж болох юм. Бид програм ажиллаж байх үед бидний бичсэн тоо маань Edit1.text-д string төрлийн хувьсагч болон хагалагдаж байдаг. Тиймээс бид str хувьсагчийн авсан гэж хэлж болно. Харин бид бодолт хийхдээ мөр төрлийг бодит тоо төрөл рүү, бодит тоог мөр төрөл рүү шилжүүлэн дэлгэцэнд edit1- ийн тусламжтайгаар харна. Тэгэхлээр бидэнд мөн мөр төрлийг бодит тоон төрөл рүү StrToFloat(); болон бодит тоог мөр төрөл рүү FloatToStr(); хөрвүүлэх функциуд хэрэг болно.

Calculator-г хийх

1. Form1-ийн Properties->Caption-нд ‘Calculator’ хэмээн бичиж өгнө.
2. Standard Tab-с MainMenu1 компонентийг авч тавьна.
3. MainMenu1 дээр хоёр дарж Form1. MainMenu цонхнонд Edit (Edit дотроо Copy, Paste, -, Exit), Color (Black, Blue, Yellow..), Help (About Calculator) гэсэн зэсүүдийг үүсгэнэ.
4. Form1.Main цонхыг хаана. Form1 цонх дээр Edit, Button-нгуудыг байрлуулан харгалзах нэрүүдийг өгнө. Edit1-н Text-д 0-ийг, Edit2-ийн

Properties->ReadOnly-ийг true болгоно. Энэ нь програм ажиллах үед Edit2 дээр бичилт хийж болохгүй гэсэн үг.

Тайлбар: Edit2 нь тухайл бал хасах үйлдэл хийгдэхэд edit2 дээр ‘–’ тэмдэг бичигдэнэ, Tg функцийг ажиллуулахад ‘Tg’ гэсэн бичилт Edit2 дээр гарна.

5. Code-ууд:

- Form1 дээр байрлах Edit->Copy товчлуурыг дарна. Тэгээд дараах кодыг бичнэ.

```
edit1.SelText;
```

```
edit1.CopyToClipboard;
```

Тайлбар: Edit1 дээр бичигдсэн бичлэгийн идэвхижуулсэн хэсгийг санах ойд буюу Clipboardад нөөцөлнө.

- Form1 дээр байрлах Edit->Copy товчлуурыг дарна. Тэгээд дараах кодыг бичнэ.

```
edit1.PasteFromClipboard;
```

Тайлбар: Edit1-ийн бичлэгийг залгуулан ClipBoardад хадгалсан бичлэгийг залгуулан бичнэ.

- Form1 дээр байрлах Color->Blue товчлуурыг дарна. Тэгээд дараах кодыг бичнэ.

```
form1.Color:=clBlack;
```

Тайлбар: Энэ үйлдэл нь Form1-ийн background-ийн өнгийг хар өнгөтэй болгоно гэсэн код.

- Бусад өнгийг дээр С үйлдлийн адил оруулж өгнө.

- Form1 дээр байрлах Help->About Calculator товчлуурыг дарна. Тэгээд дараах кодыг бичнэ.

```
ShowMessage('Ene calculator-g Delphi 7.0 program deer bvteew');
```

- Одоо 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 гэсэн товчлууруудыг бид 4-р үйлдэл дээр үүсгэсэн байгаа. Эдгээр Button-нгууд дээр дараад бичих код нь ерөнхийдөө ижилхэн. Жишээ болгож 1 товчлуур дээр дараад

```
str:=str+floattostr(1);
```

```
edit1.Text:=str;
```

Тэгвэл 2 гэсэн товчлууд дээр дараад

```
str:=str+floattostr(2);
```

```
edit1.Text:=str; гэх мэт цааш нь бусад товчлууруудыг бичиж болно.
```

- ‘.’-ыг мөн дээрхийн адил бичиж өгч болно.

```
str:=str+'.';
```

```
form1.Edit1.Text:=str;
```

- Тэмдэг солих буюу +/- товчлуурын код нь:

```
str:=floattostr(-1*strtofloat(form1.Edit1.Text));
```

```
edit1.Text:=str;
```

Тайлбар тухайн тоог (-1)-р үржүүлбэл тухайн тооны тэмдэг нь хялбар солигдоно.

- i. +, /, *, -, үйлдлүүдийг яаж хийх вэ? Гэсэн асуулт бидний дараагийн ажил болно. Код бичилт нь:

```
edit2.Text:='+'; { Edit2 дээр + тэмдэг гарч + үйлдэл хийгдлээ гэж заана }
a:=strtofloat(Edit1.Text); { Үлдэл хийгдэхээс өмнөх утгыг а хувьсагчид олгоно. }
str:=""; { Үйлдэл хийгдэсний дараа хувьсагчийн утгыг чөлөөлж дараагийн тоог хадгалахад бэлэн болно }
```

Тайлбар: Бусад -, /, * үйлдэл нь мөн дээрх зарчмаар явагдана.

- j. Одоо тэгвэл '=' буюу бодох үйлдэл.

```
var
i:integer;
j:real;
r:real;
begin j:=1;
b:=strtofloat(str); { +, -, *, / үйлдэл хийгдэсний дараах утгыг б хувьсагчдад хадгалж авах }
if edit2.Text='+' then begin c:=a+b; a:=c; end; { + үйлдэл хийгсэн бол хоёр тоог нийлбэрийг }
if edit2.Text='-' then begin c:=a-b; a:=c; end; { - үйлдэл хийгсэн бол хоёр тооны ялгаврыг }
if edit2.Text='*' then begin c:=a*b; a:=c; end; { * үйлдэл }
if edit2.Text='/' then begin c:=a/b; a:=c; end; { / үйлдэл }
if edit2.Text='x^y' then begin
    for i:=1 to round(b) do
        j:=j*a;
        c:=j;
        a:=c;
    end;
```

form1.Edit1.Text:=floattostr(c); бодогдосны дараах утгыг Edit1-д хэвлэн гаргана.

- k. Sin, Cos, Tg, Ctg функцууд. Код бичилт нь

```
edit2.Text:='Sin(x)=';
a:=strtofloat(form1.Edit1.Text); { edit1-ийн утгыг бүхэл тооруу шилжүүлэн хадгална }
c:=sin(a); { sin(x) функцыг бодож утгыг нь с-д хадгална }
edit1.Text:=floattostr(c); { edit1-д с-ийн утгыг string төрөлд хөрвүүлэн хэвлэнэ }
str:="";
```

Талбар: Cos, Tg, Ctg функцууд нь дээрхтэй адилхан бөгөөд мэдээж c:=tg(a) эсвэл c:=cos(a) гэх мэтээр жаахан л өөрчлөгднө.

- l. BackSpace буюу бичсэн зүйлийг нэг нэг орноор нь арилгах үйлдэл.

```
var
i:integer;
begin str:="";
begin for i:=1 to length(edit1.Text)-1 do
    str:=str+edit1.Text[i];
    if str<>" then edit1.Text:=str
    else edit1.Text:='0';
end;
```

Тайлбар: length-нь мөр төрөлтэй хувьсагчийн уртыг олдог функ тиймээс length-ээс 1-р хасч дэлгэцэнд хэвлэж байна. Мөн хэрэв бид тоог нэг нэг орноор нь хассаар байгаад тоон нь дуусчихвал автоматаар дэлгэцэнд 0 гэж гарч ирдэг. Тиймээс str<>’ нөхцөл биелэхгүй буюу хоосон байвал Edit1.Text:=0 гэж өгнө.

м. Sqrt, Sqr функцууд. Sqrt - өгсөн тооноос язгуур авах үйлдэл. Мэдээж сөрөр тооноос язгуур авч болохгүй учир тоог сөрөг тоо эсэхийг шалгах хэрэгтэй.

```
a:=strtofloat(str);
if a<0 then begin label1.Caption:='Tanii oruulsan ugugdul buruu bn';
           edit1.Clear;
           str:="";
       end
else begin
           edit1.Text:=floattostr(sqrt(strtofloat(edit1.Text)));
       end;
```

Харин Sqr бол өгсөн тоог квадрат зэрэгт дэвшүүлдэг учир ямар нэгэн нөхцөл байхгүй. Өгсөн edit1.text-ийн мөр төрлийн хувьсагчийг бодит тоолуу хүрвүүлээд, хүрвүүлж гарсан бодит тоогоо квадрат зэрэг дэвшүүлээд иргээгээд бодит тоогоо мөр төрөл рүү шилжүүлэн edit1.text-д буцааж оноож өгнө.

```
edit1.Text:=floattostr(sqr(strtofloat(edit1.Text)));
```

н. Бүх зүйлийг устгах С товчлуур

Мэдээж бүх зүйлийг устгах тул санасан тоо, хийгдсэн үйлдлүүд цөм цуцлагдана. Тиймээс дараах бичлэгт биелэгдэнэ.

```
form1.Edit1.Text:='0'; { Дэлгэцэнд 0 гэж хэвлэгдэн гарна }
```

```
str:="; { санасан тоог хоосон болгох буюу устгана }
```

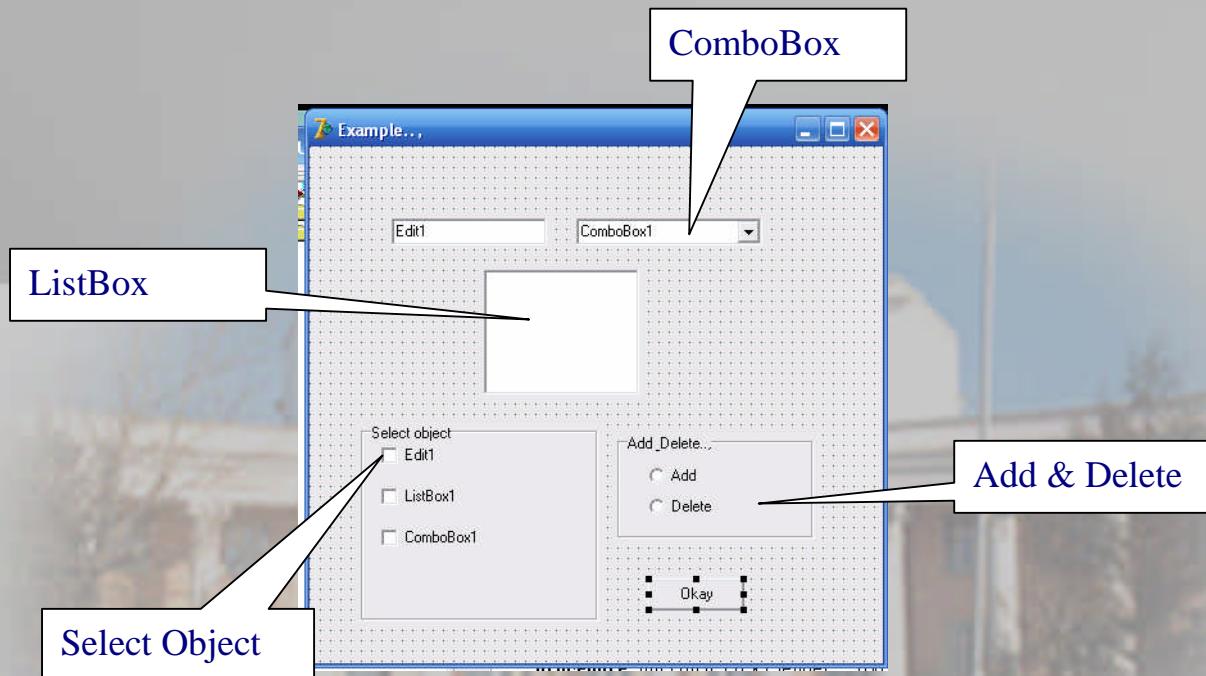
```
edit2.Text:=" { Хийгдсэн үйлдлүүд цуцлагдана }
```

о. F9 товчлуурыг дарж програмыг ажиллуулна.

Лабораторын ажил

ListBox, ComboBox ашиглах нь

Програмын бүтэц:



Програмын ажиллах горим:

- Edit1 талбарт Банкны нэрийг оруулаад Select Object талбараас Edit, ListBox, ComboBox-ийг сонгоод Add командыг идэвхижүүлж океу товчлуурыг сонгоно. Энэ үйлдлийн үр дүнд сонгосон объект-д банк нэр нэмэгдэн орно.
- Select Object талбараас объект-ийг сонгож идэвхижүүлээд Delete командыг сонгоход сонгосон объект-ээс утгууд нь нэг нэгээр хасагдах болно.

Програмыг хийх горим:

1. Форм дээр Edit1, ComboBox1, Listbox болон GroupBox-с хоёрыг авч тавин байрлуулна.
2. GroupBox1-ийг идэвхижүүлэн GroupBox-д 3-н CheckBox-ийг авч тавин холбогдох нэрүүдийг нь олгоно.
3. GroupBox2-ийг идэвхижүүлэн GroupBox-д 2 RadioButton-ийг байрлуулан холбогдох нэрүүдийг өгнө.
4. Button1 дээр хоёр дараах кодыг бичнэ.

```
if CheckBox2.Checked and RadioButton1.Checked then Listbox1.Items.Add(edit1.Text);
if CheckBox3.Checked and RadioButton1.Checked then ComboBox1.Items.Add(Edit1.Text);
if CheckBox1.Checked and RadioButton2.Checked then Edit1.Clear;
if CheckBox2.Checked and RadioButton2.Checked then Listbox1.Items.Delete(strtoint(edit1.Text));
if CheckBox3.Checked and RadioButton2.Checked then ComboBox1.Items.Delete(Strtoint(Edit1.Text));
```

Тайлбар: Checked гэдэг нь тухайн CheckBox, RadioButton нь сонгогдсан(чагтлагдсан) байна уу гэдгийг шалгана.

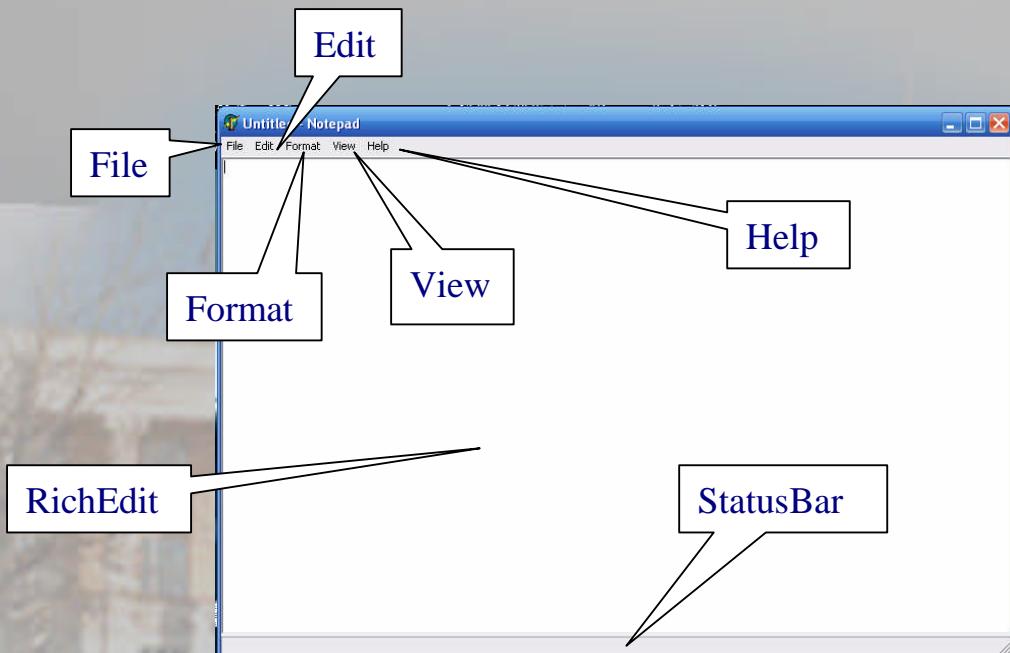
5. F9 товчлуурыг даран програмыг ажиллуулна.

Лабораторын ажил

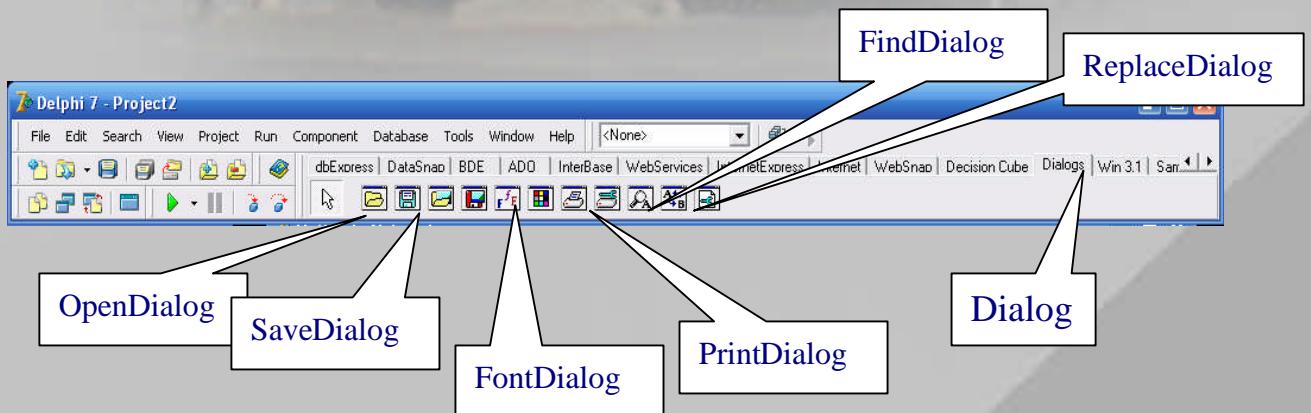
Сэдэв: Notepad

NotePad програм нь .txt өргөтгөлтэй файлтай ажилладаг бөгөөд text файлыг үүсгэх, засах үйлдлийг гүйцэтгэж байдаг text-ийн editor юм.

Notepad програмыг бүтэц



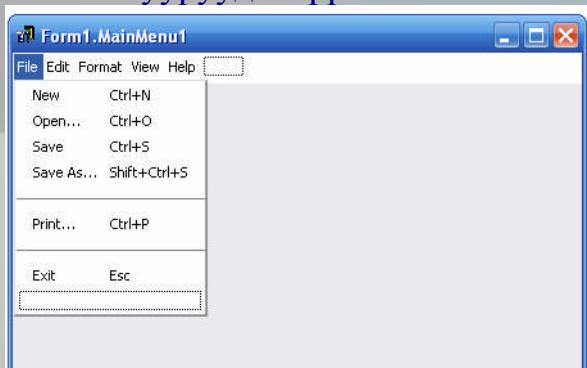
Notepad хийгдэхэд ашиглагдах нэмэлт компонентууд



Dialog Tabs: Бид windows үйлдлийн системийн зарим харилцах цонх буюу Dialog-ийн ашиглаж болно. Жишээлбэл бид хуудсыг принтерлэхийн тулд Print Dialog-той ажилладаг.

NotePad програм хийх:

1. Шинээр формыг үүсгэнэ.
2. Standard tabs-с MainMenu1, Win32 tabs-с RichEdit1 компонентийг форм дээр байрлуулна.
3. MainMenu1 компонент дээр хоёр дараан Form1.MainMenu1 харилцах цонхыг нээж File, Edit, Format, View, Help зэснүүдийг болон тэдгээрт харгалзах товчлууруудыг үүсгэж өгнө.

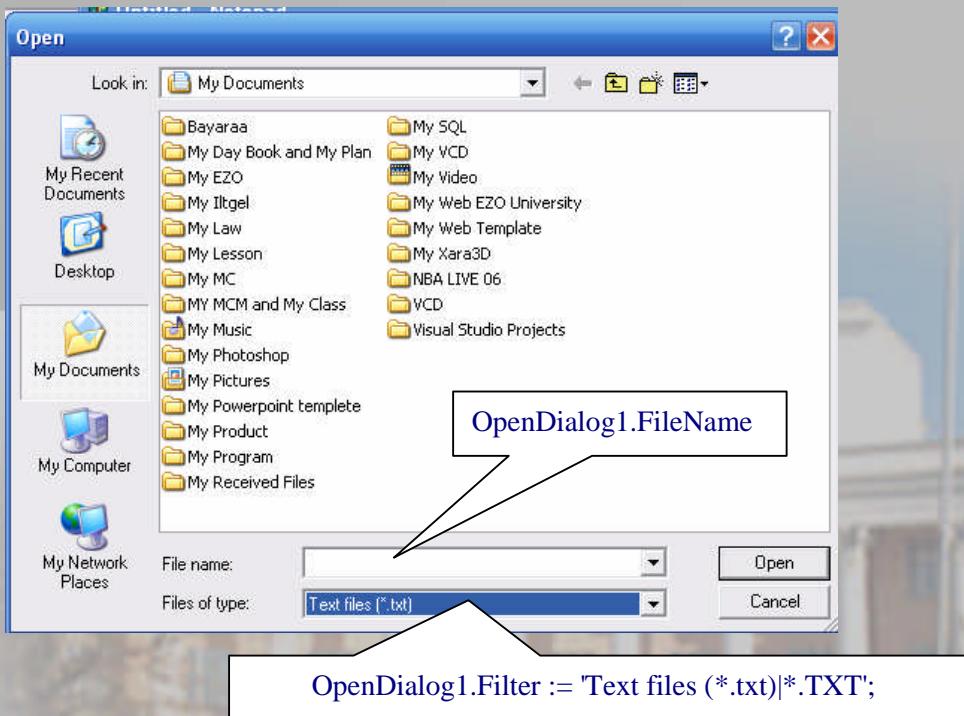


File зэсэнд New, Open, Save, Save As, Print, Exit
Edit зэсэнд Undo, Copy, Cut, Paste, Del, Find, Select All, Time/Data
Format зэсэнд Font
View зэсэнд StatusBar
Help зэсэнд About Notepad зэрэг товчлууруудыг үүсгэж өгнө.

4. RichEdit1-ийг сонгон идэвхижүүлж түүний properties->Align шинжийг alClient болгоно.
5. Одоо Dialog tabs-с Opendialog, SaveDialog, Fontdialog, Printdialog, FindDialog компонентуудыг форм дээр байрлуулна. Эдгээр компонентийн талаар дэлгэрэнгүй мэдээллийг авахыг хүсвэл тухайн компонент дээр дараад F1 товчлуурыг дараан help файлаас харж болох юм.
6. File зэсний үйлдлүүдийг хийх:
 - a. File->New-ийг сонгоно үүсэх талбарт RichEdit1.Clear;
 - b. File->Open-ийг сонгоно үүсэх талбарт OpenDialog1.Filter := 'Text files (*.txt)|*.TXT';
if OpenDialog1.Execute then
RichEdit1.Lines.LoadFromFile(OpenDialog1.FileName)
else
RichEdit1.Lines.Clear;

Тайлбар: 'Text files (*.txt)|*.TXT'; - нь бид зөвхөн компьютерээсээ .txt өргөтгөлтэй файлуудыг ялган харуулдаг. Учир нь notepad нь зөвхөн .txt файлтай ажилладаг. OpenDialog1.Execute- гэдэг нь OpenDialog цонх нээгдэн дэлгэцэнд харагдаж буйг шалгадаг функц хэрвээ OpenDialog цонх нээгдсэн бол Execute-нь true утгыг авдаг болно.

RichEdit1.Lines.LoadFromFile(OpenDialog1.FileName) – нь RichEdit1-д таны openDialog-р сонгосон файлыг нээнэ.



c. File->Save As

```
var filename:string;
```

```
begin
```

```
SaveDialog1.Filter := 'Text files (*.txt)|*.TXT';
```

```
if SaveDialog1.Execute then
```

```
begin
```

```
filename:=SaveDialog1.FileName+'.txt';
```

```
RichEdit1.Lines.SaveToFile(Filename);
```

```
Form1.Caption:=FileName;
```

```
end
```

```
else
```

```
RichEdit1.Lines.Clear;
```

Тайлбар: RichEdit1.Lines.SaveToFile(Filename); - нь SaveDialog-д таны өгсөн нэрээр файлыг хадгална.

d. File -> Save

```
if filename = " then
```

```
begin
```

```
SaveDialog1.Filter := 'Text files (*.txt)|*.TXT';
```

```
if SaveDialog1.Execute then
```

```
begin
```

```
filename:=SaveDialog1.FileName+'.txt';
```

```
RichEdit1.Lines.SaveToFile(Filename);
```

```
Form1.Caption:=FileName;
```

```
end
```

```
else
  RichEdit1.Lines.Clear;
end
else RichEdit1.Lines.SaveToFile(Filename);
```

Тайлбар: Save команд нь Save As командаас ялгаатай тал нь тухайн файл нь өмнө нь хадгалагдсан байна уу гэдгийг шалгаад хэрэв өмнө нь хадгалцан байвал түүнд дээрээс нэмж хадгална. Харин өмнө нь хадгалагдаагүй байвал Save As команыг гүйцэтгэнэ. Тиймээс filename=’’ хэмээн нөхцөл тавсан байна.

e. File->Print

```
form1.PrintDialog1.Execute;
```

Тайлбар: энэ нь зөвхөн PrintDialog-ийг нээх үйлдэл бөгөөд бусад зүйлсийг нь та help-с харна уу.

f. File->Exit

```
form1.Close;
```

7. Edit зэсний үйлдлүүдийг хийх:

a. Edit->Undo

```
RichEdit1.Undo;
```

b. Edit->Copy

```
RichEdit1.SelText;{Copy хийх хэсгийг идэвхижүүлнэ}
```

```
RichEdit1.CopyToClipboard; {идэвхижүүлсэн хэсгийг хуулна}
```

c. Edit->Cut

```
RichEdit1.SelText;
```

```
RichEdit1.CutToClipboard;
```

d. Edit->Paste

```
RichEdit1.PasteFromClipboard;
```

e. Edit->Delete

```
RichEdit1.SelText;
```

```
RichEdit1.ClearSelection;
```

f. Edit->Find {бичлэгээг хайлт хийх}

```
FindDialog1.Position := Point(RichEdit1.Left + RichEdit1.Width, RichEdit1.Top);
```

```
FindDialog1.Execute;
```

Тайлбар: Энэ үйлдэл нь зөвхөн Find харилцах цонхыг нээх ба хайлт хийхийн тулд FindDialog –ийн Object Inspector цонх OnFind үзэгдэл (Event)-д
var

```
  FoundAt: LongInt;
```

```
  StartPos, ToEnd: Integer;
```

```
begin
```

```
  with RichEdit1 do
```

```
begin
```

```
  { begin the search after the current selection if there is one }
```

```
  { otherwise, begin at the start of the text }
```

```
  if SelLength <> 0 then
```

```
StartPos := SelStart + SelLength
else

  StartPos := 0;

{ ToEnd is the length from StartPos to the end of the text in the rich edit control }

ToEnd := Length(Text) - StartPos;

FoundAt := FindText(FindDialog1.FindText, StartPos, ToEnd, [stMatchCase]);
if FoundAt <> -1 then
begin
  SetFocus;
  SelStart := FoundAt;
  SelLength := Length(FindDialog1.FindText);
end;
end;

кодыг бичиж өгнө. Нэмэлт тайлбарыг help-с авна уу.
g. Edit->Select All
RichEdit1.SelectAll;
h. Edit->Time/Date {бичлэгт тухай он, сар, өдөр, цаг, минутыг нэмэх}
var
  SystemTime: TSystemTime;
begin GetLocalTime(SystemTime);
  RichEdit1.Lines.Add(Datetimetostr(SystemTimeToDateTime(SystemTime)));
end;
```

Тайлбар: Тухайн мөчийн цаг хугацаа он сар өдрийг системийн цагаас авдаг.
тиймээс SystemTime функцыг хэрэглэсэн.

8. Format зэс

a. Format->Font

```
if form1.FontDialog1.Execute then
```

```
  RichEdit1.Font:=form1.FontDialog1.Font;
```

Тайлбар: Фонд диалогоос бидний сонгосон фонд, үсгийн хэлбэр өнгийг формийн фондод өгнө.

9. Help зэс

a. Help -> About NotePad

```
showmessage('Лабораторын ажил Notepad');
```

10. F9 товчлуурыг дарж программыг ажиллуулна.



4-р бүлэг.
График, мультимедиатай
ажиллах

TCanvas класс

Delphi програмд порм дээр объектийг зурахдаа TCanvas классаас удамшсан Canvas функцийг хэрэглэж байдаг. Canvas шинж чанарыг run-time үед л зөвхөн ашиглах боломжтой object Inspector цонхонд үзэгддэгүй байна. Гол шинж чанар нь:

Brush: Дүрсийг дүүргэж будах өнгө, хэв загварыг тодорхойлно.

Font: Canvas дээр бичигдэх текстийн фондыг олгоно.

Pen: Canvas дээр дүрс зураг үзэгний өнгө болон өнгийг тодорхойлно.

Canvas-ийн дүрс болон текстийг зурах функциуд

Canvas.Arc() – Canvas дээр нум зурна

Canvas.Ellipse() – Canvas дээр эллипсийг зурна

Canvas.Polygon() – оор олон өнцөгтийг

Canvas.RoundRec() – дугуй булантай тэгш өнцөгтийг

Canvas.PolyLine() – олон шулууныг зурна

Canvas.TextOut(X,Y,'Your Text'); - өгсөн x,y цэг дээр текстийг бичнэ.

Canvas.TextRect(Rect,X,Y,'Your Text'); - өгсөн цэг дээр тэгш өнцөгт зуран түүн дотроо текстийг бичнэ.

Мультимедиа

Delphi-ын VCL-д мультимедиа (Sound, video) файлыг тоглуулах TAnimation, TMediaPlayer компонент болон API-ийн PlaySound функцыг өргөн хэрэглэдэг.

PlaySound API-ын функц

PlaySound API функцийг wave медиа файлуудыг тоглуулахад хэрэглэдэг. Энэ функцыг ашиглахыг тулд формын interface хэсгийн uses модулийн жагсаалтанд mmSystem модулыг бичиж өгч тодорхойлно. Энэ функц нь дараах бүтэцтэй

PlaySound('C:\WinNt\Media\Chord.wav', 0, SND_FILENAME);

'C:\WinNt\Media\Chordwav' – тухайн тоглуулах гэж байгаа файлын нэр байршил SND_FILENAME – дискнээс аудио файлыг тоглуулахад ашигладаг флаг. Мөн зориулалтаасаа хамаарч энэ параметр нь SND_ASYNC, SND_SYNC, SND_ALIAS, SND_NODEFAULT гэсэн утгуудыг авдаг.

TAnimate компонент



TAnimate компонентийг windows үйлдлийн системд ашиглагддаг жижиг хэмжээний avi видеог тоглуулахад хэрэглэдэг. Онцлог шинжууд нь гэвэл:

Active: true үед видео файлыг тоглуулж эхэлнэ.

commonAVI: энэ шинж нь стандарт aviFindFolder, aviFindFile, aviEmptyRecycle, aviFindComputer, aviCopyFiles, aviRecycleFile, aviDeleteFile утгуутдай бөгөөд нэрнээс нь бараг ямар анимашин хөдөлгөөн болох нь тодорхой байхаа.

FileName: avi видео файлыг өөрөө гаднаас оруулж тоглуулах үед нэрийг нь тодорхойлж өгнө.

Repetition: Видео хэдэн удаа давтагдахыг тодорхойлно.

startFrame, *stopFrame*: эдгээр шинж чанаруудад тоглуулах видео файлын эхлэл, төгсгөлийн фрэймийн дугаарыг олгоно.

Timers: frame-г цаг ашиглан үзүүлэх эсэхийг тодорхойлно.

TMediaPlayer компонент

Энэ компонент нь mp3, cdi, wav, mpg, dat.. форматтай файлуудыг тоглуулах чадалтай. Програмд Sound, Video файлыг тоглуулахдаа panel авч тавин TMediaPlayer-ийн ControlBar-тай холбож өгдөг. Онцлог шинжууд:

AutoEnable: TMediaPlayer-ийн open функцийг дуудах үед автоматаар идэвхитэй болох эсэхийг тодорхойл.

AutoOpen: FileName шинж чанарт медиа файлыг олгосон үед MediaPlayer-ын open функцийг автоматаар дуудах эсэхийг тодорхойлно.

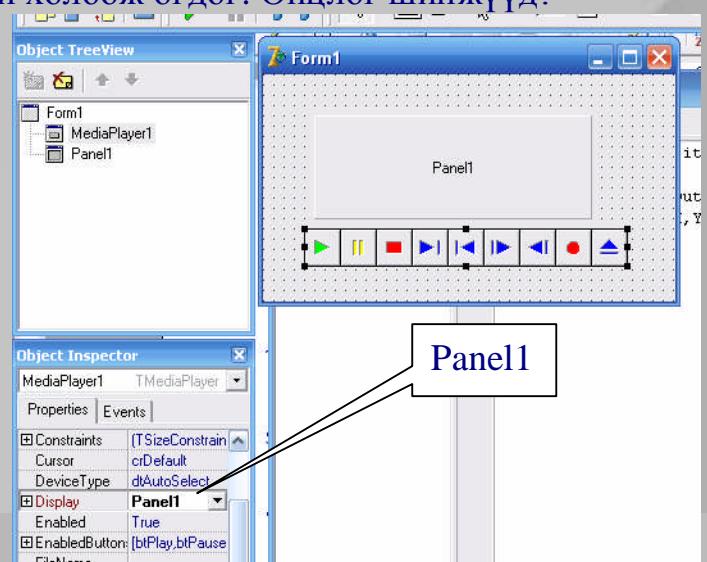
AutoRewind: true үед файлын төгсгөлд ирхэд дахин эхнээс тоглоно.

DeviceType: Мультимедиа төхөөрөмжийн төрлийг тодорхойлно.

Display: Видео файлыг гаргах window-г зааж өгнө.

DisplayRec: Видео файлыг үзүүлэх талбайг Rect төрлөөр тодорхойлж өгнө.

EnabledButtons: MediaPlayer дээр үзэгдэх товчнуудыг тодорхойлно.



Endpos: Медиа файлын төгсгөлийн байрлал.

FileName: тоглуулах гэж буй файлаан нэрийг зааж өгнө.

Mode: Төхөөрөмжийн төлөв байдлыг тодорхойлно. *mpPlaying*, *mpRecording*, *mpSeeking*, *mpPaused*, *mpOpen*, *mpNotready*, *mpStopped* гэсэн утгуудыг авдаг.

Notify: гүйцэтгэсэн үйлдлийг мэдээлэх эсэхийг тодорхойлно.

NotifyValues: Хамгийн сүүлд мэдээлсэн үйлдлийн үр дүн

Position: Медиа файлаан тухайн байрлалыг тодорхойлдог.

TimeFormat: Тухайн төрөөмжинд зориулж ашигладаг цагийн форматыг тодорхойлж өгнө.

Tracks: Медиа файлаан агуулах track-ийн тоог тодорхойлно.

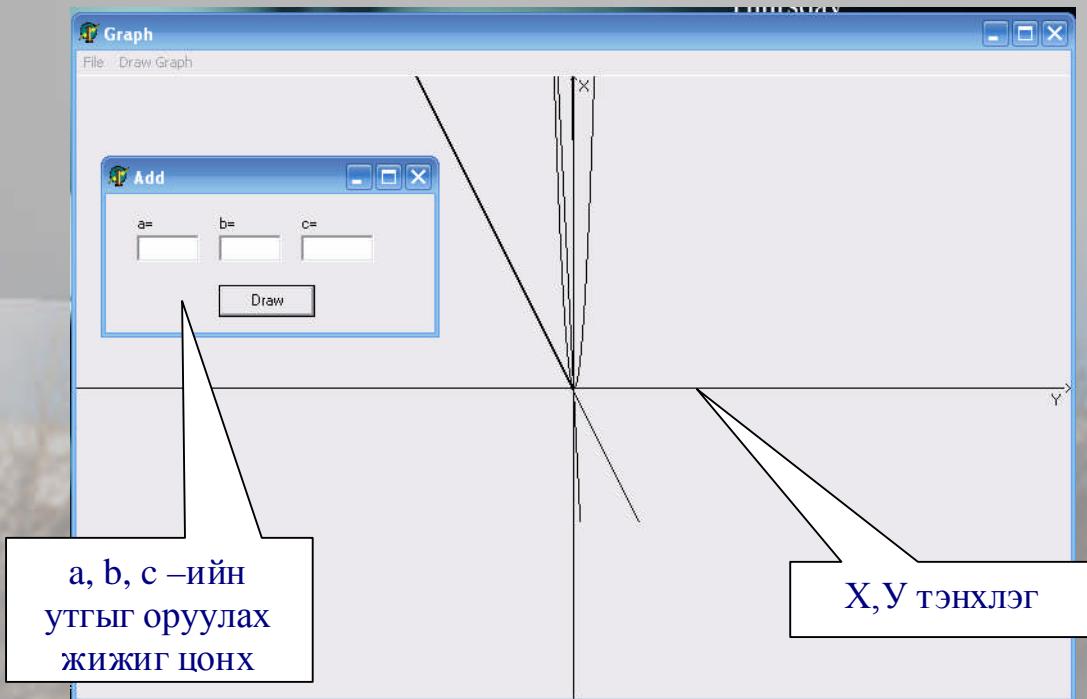
VisibleButtons: ControlBar дээр үзэгдэх MediaPlayer-н товчийг тодорхойлж өгнө.

Wait – Програмыг дуудах үед удирдлагыг буцаах эсэхийг тодорхойлно.

Лабораторын ажил

$ax^2 + bx + c = 0$ функциянын график

Програмыг бүтэц



Графикийг зурах:

1. Form1-ийн object inspector цонх (event) ўзэгдлүүдээс OnPaint үзэгдэл дээр хоёр дараах кодыг бичнэ.

```
form1.Canvas.Pen.Color:=clblack;  
form1.Canvas.MoveTo(0,form1.ClientHeight div 2);  
form1.Canvas.LineTo(form1.ClientWidth, ClientHeight div 2);  
form1.Canvas.TextOut((Form1.ClientWidth div 2) + 5, 1, 'X');  
form1.Canvas.TextOut(Form1.ClientWidth div 2 -3, -1 , '^');  
form1.Canvas.MoveTo(form1.ClientWidth div 2 ,0);  
form1.Canvas.LineTo(form1.ClientWidth div 2 ,ClientHeight);  
form1.Canvas.TextOut(Form1.ClientWidth-15, Clientheight div 2+3 , 'Y');  
form1.Canvas.TextOut(Form1.ClientWidth-5, Clientheight div 2-7 , '>');  
end;
```

Тайлбар: Энэ үйлдэл нь форм дээр Х, У тэнлэгийг зурна. Canvas.MoveTo() функц нь зурах эхлэлийн цэгийг зааж өгнө. Canvas.LineTo() функц нь эхлэлийн цэгээг уг цэг хүртлэх шулууныг зурна.

2. Form дээр MainMenu компонентийг тавин Exit, Draw Graphi зэснүүдийг үүсгэнэ.

3. Exit зэсэнд дээр дарахад

```
Form1.close;
```

4. File зэснээс New->Form командыг өгөх буюу шинэ формыг үүсгэнэ.

5. Draw Graphi зэсийг сонгоход

Form2.show;

6. Form2 дээр 3-н edit, 1 button тавьна.

Тайлбар: 3-н Edit-д бид а, b, c-ийн утгуудыг бичиж өгнө.

7. Form2 дээр байрлах button дээр хоёр даран

var

a,b,c:integer;

i:integer;

begin

a:=strtoint(form2.Edit1.Text);

b:=strtoint(form2.Edit2.Text);

c:=strtoint(form2.Edit3.Text);

form1.Canvas.MoveTo(form1.ClientWidth div 2, form1.ClientHeight div 2);

for i:=(form1.ClientWidth div 2) to form1.ClientWidth div 2 do

form1.Canvas.LineTo(i+form1.ClientWidth div 2, -a*i*i+b*i+c+form1.ClientHeight div 2);

form2.Edit1.Clear;

form2.Edit2.Clear;

form2.Edit3.Clear;

form2.Close;

end;

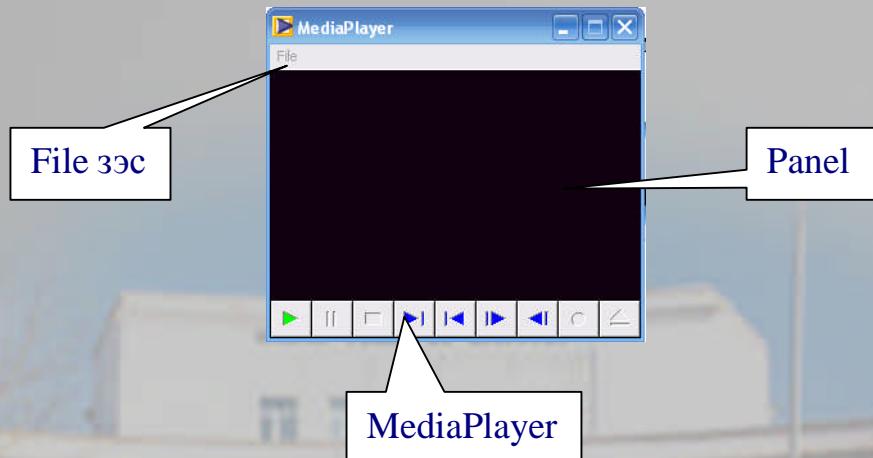
Тайлбар: form1.Canvas.LineTo(i+form1.ClientWidth div 2, -a*i*i+b*i+c+form1.ClientHeight div 2); - нь $y=ax^2+bx+c$ функц бичсэн болно.

8. F9 товчлуурыг дарж програмыг ажиллуулна.

Лабораторын ажил

Мультимедиа

Програмыг бүтэц



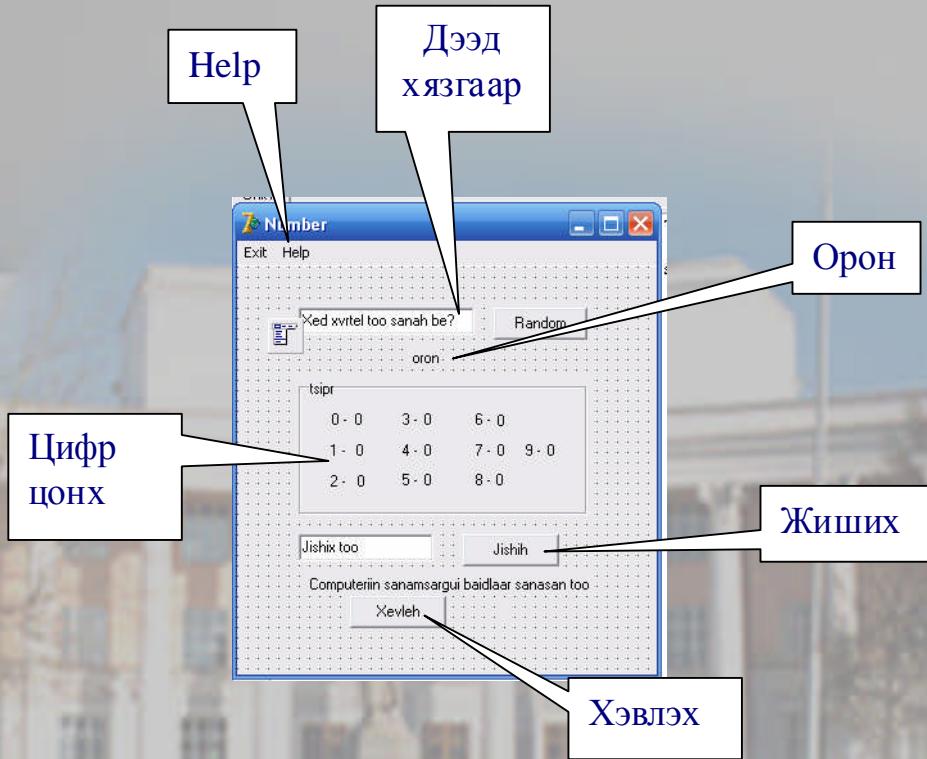
Програмыг хийх:

1. Form дээр Standard tabs-с Panel, MainMenu, Dialog tabs-с OpenDialog-ийг, System tabs-с MediaPlayer компонентыг тус тус байрлуулна.
2. MainMenu компонент дээр хоёр даран File зэсийг үүсгэх ба файл зэс нь дотроо Open, Exit гэсэн талбартай байна.
3. MediaPlayer компонентийн properties(шинж)-ийн display талбарт Panel1-ийг зааж өгнө.
4. File зэсний командууд
 - a. File->Exit
Form1.Close;
 - b. File->Open
if opendialog1.Execute then
begin
mediaplayer1.FileName:=opendialog1.FileName;
mediaplayer1.open;
mediaplayer1.DisplayRect:=Panel1.ClientRect;
end;
5. Таньд баяр хүргье та MediaPlayer програмыг зохиочихлоо. Та одоо F9 товчлуур дээр даран програмыг ажиллуулж үзнэ үү.

Лабораторын ажил

Миний санасан тоо хэд вэ?

Програмын бүтэц



Программыны ажиллах горим:

‘Xed xvrtel too sanah be?’ талбарт та өөрийн дээд хязгаарыг бичиж өгнө. Жишээлбэл: 2000 дараа нь Random товчлуурыг дараход компьютер санамсаргүй байдлаар 0-2000 хүртлэх тооноос нэг тоо санах ба тэр тоо нь хэдэн оронтой болохыг ‘орон’ талбараас харж болно. Хэрэв таахад хэцүү байвал Help зэсийг сонгох хэрэгтэй. Тэгвэл ‘tsipr’ талбарт 0-9 хүртлэх цикрууд хэдэн ширхэг орсон нь тоологдоод гарыад ирнэ. ‘Jishih too’ та энэ талбарт таамаглаж буй тоогоо хийгээд ‘Jishih’ товчлуур дээр даран шалгаж болно. Эцэст олохгүй бол ‘Xevleh’ товчлуур дээр дарж компьютер ямар тоо санасныг үзэж болно.

Ашиглагдах хувьсагчид

Number: real;

teg,neg,hoer,gur,dvr,tav,zur,dol,naim,es:integer;
str:string;

Тайлбар: Нэг гэсэн цифр хэдэн ширхэг орсон байгааг тоог хадгалдаг хувьсагч бол ‘neg’ гэх мэтээр хувьсагчийг сонгон авсан.

Хийх дараалал:

1. Form дээр MainMenu-1, Edit-2, Groupbox-1, Button-3, Label-3 ширхэгийг тавьна.
2. GroupBox-ийн идэвхижүүлээд дээр нь 20 ширхэг label ахин нэмж тавих ба тохигох нэрийг өгнө.

3. **MainMenu** компонент дээр хоёр даран From1.MainMenu1 харилцах цонхонд Exit, Help зэсийг үүсгээд цонхыг хаана.
4. **Random** товчлуур дээр дараад хийгдэх үйлдэл:

```
number:=Random(StrToInt(Edit1.Text));
str:=FloatToStr(number);
label13.Caption:=IntToStr(length(str));
```

Тайлбар: Компьютер Random функцын тусlamжтайгаар санамсаргүй байдлаар тоог санах болно. Тэр тоогоо Number хувьсагчид өгч хадгална.

5. **Help** зэсийг сонгоход хийгдэх үйлдэл.

```
var
i:integer;
begin teg:=0;neg:=0;hoer:=0;gur:=0;dvr:=0;tav:=0;zur:=0;dol:=0;naim:=0;es:=0;
For i:=1 to length(str) do
begin
if str[i]='0' then teg:=teg+1; {0 цифр хэд орж байгааг тоолж байна.}
if str[i]='1' then neg:=neg+1;
if str[i]='2' then hoer:=hoer+1;
if str[i]='3' then gur:=gur+1;
if str[i]='4' then dvr:=dvr+1;
if str[i]='5' then tav:=tav+1;
if str[i]='6' then zur:=zur+1;
if str[i]='7' then dol:=dol+1;
if str[i]='8' then naim:=naim+1;
if str[i]='9' then es:=es+1;
end;
label1.Caption:=inttostr(teg); { 0 цифр хэдэн ширхэг байгааг label-ээр дамжуулан дэлгэцэнд хэвлэж байна. }
label2.Caption:=inttostr(neg);label3.Caption:=inttostr(hoer);
label4.Caption:=inttostr(gur);label5.Caption:=inttostr(dvr);
label6.Caption:=inttostr(tav);label7.Caption:=inttostr(zur);
label8.Caption:=inttostr(dol);label9.Caption:=inttostr(naim);
label10.Caption:=inttostr(es);
end;
```

6. **Jishih** товчлуурыг дараад хийгдэх үйлдэл:

```
if number=StrToFloat(Edit2.Text) then showmessage('you are win')
else showmessage('Tentsehgui baina');
```

Тайлбар: хэрвээ таны оруулсан тоо зөв байвал ‘you are win’ бичиг гарах ба харин таны оруулсан тоо тэнцэхгүй байвал “Tentsehgui baina” гэж гарч ирнэ.

7. ‘Xevleh’ товчлуурыг дарвал:

```
label12.Caption:=FloatToStr(number);
```

8. F9 товчлуурыг дарж програмыг ажиллуулна уу?



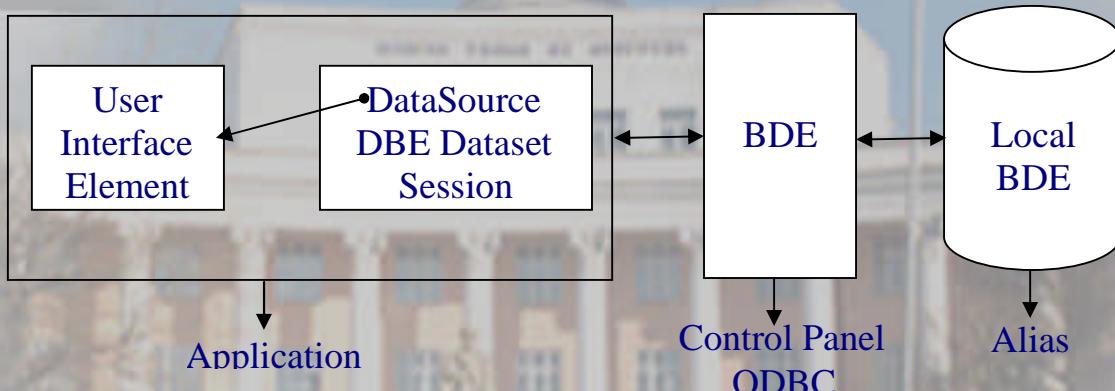
5-р бүлэг.
Өгөгдлийн сантай
ажиллах

Мэдээллийн сангийн тухай ойлголт

Хоорондоо холбоо бүхий хүснэгтуүдийн нэгдлийг өгөгдлийн бааз гэнэ. Хүснэгтийн баганыг талбар(Field), мөрийг бичлэг(record) гэж нэрлэдэг. Мэдээллийн сан нь түүнийг удирдах систем дээр үүсгэж боловсруулдаг.

Delphi програм нь өгөгдлийн баазын гурван өөр механизм дээр тулгуурласан application боловсруулж чадна. Үүнээс бид зөвхөн нэгийг нь дэлгэрүүлж үзэх болно.

LocalDataBase – файл суурьтай



1. User Interface Element:



Delphi програм дээр өгөгдлийн баазыг хэрэглэлчдэд харуулах боломжтой

компонентууд байрлах бөгөөд эдгээрийг Data aware компонент гэдэг. Эдгээр компонентууд нь Data Control tab-д байрладаг. BDGrid, DBEdit, DBLocal, DBComboBox.. гэх мэт. Эдгээр компонентууд дээр Datasource гэсэн шинж properties-д байдаг. Энэ шинжид тухайн форм дээр тавигдсан Datasource-д холбож өгнө.

2. DataSource: энэ нь Хэрэглэгчийн User Interface Element-ийг BDE Data Set-д холбож өгдөг.
3. BDE Data Set энэ нь Session-р дамжин өгөглийн баазтай холбогддог. Үүнд Table, Query, StoredProc орно.
4. Session: BDE alias-г бүртгэх удирдах устгах үйлдлүүдийг гүйцэтгэдэг.

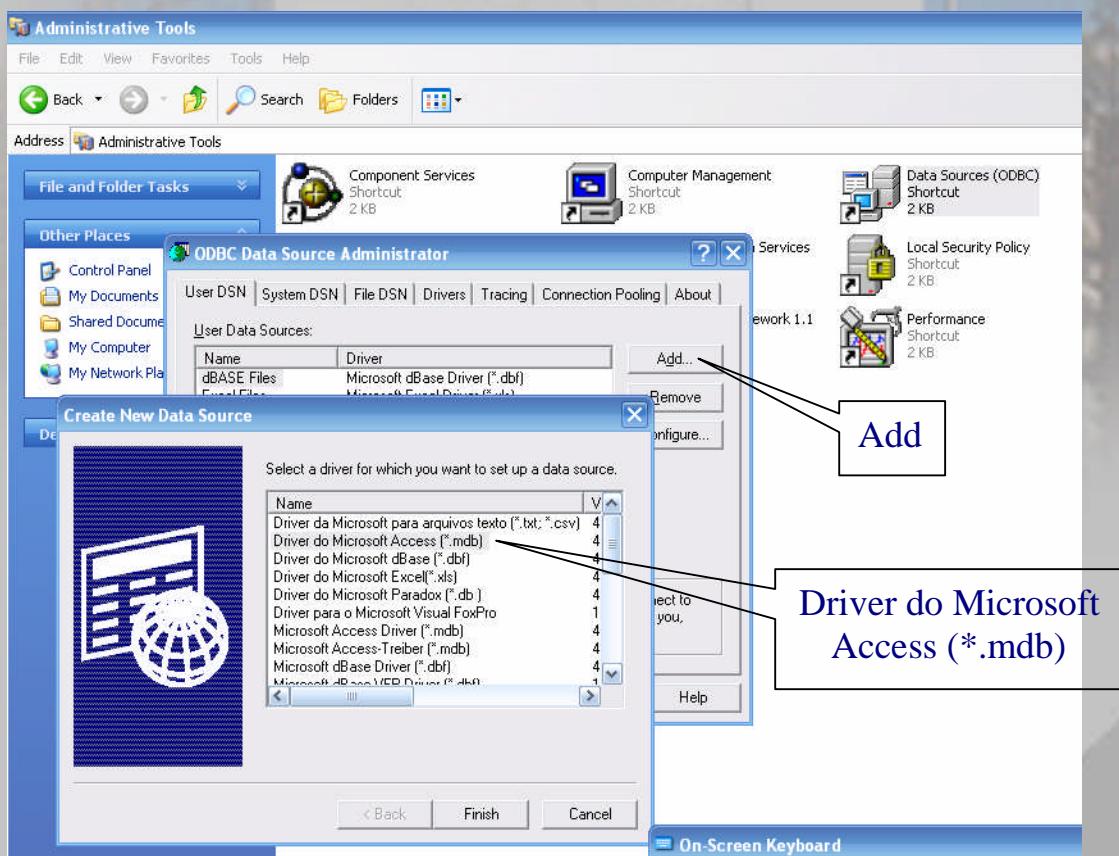
Энэ нь хэрэглэж буй технологиос хамаарч өөр өөрийн гэсэн онцлог шинж бүхий компонентуудыг ашигладаг. Үүнд:

BDE	InterBase	ADO	dbExpress
TDataBase	TIBDatabase	TADOConnection	TSQLConnection
TTable	TIBTable	TADOTable	TSQLTable
TQuery	TIBQuery	TADOQuery	TSQLQuery
TStoredProc	TIBStoredProc	TADOSStoredProc	TSQLStoredProc
	TIBDataset	TADODataset	TSQLDataset

ODBC Alias үүсгэх

Alias гэдэг нь таны өгөгдлийн баазыг компьютерт хаана байрлаж байгааг програмд зааж өгөх бөгөөд програм тус замын дагуу хандаж өгөгдлийн баазтай харилцана. Тиймээс үнэн зөв алдаагүй байх хэрэгтэй

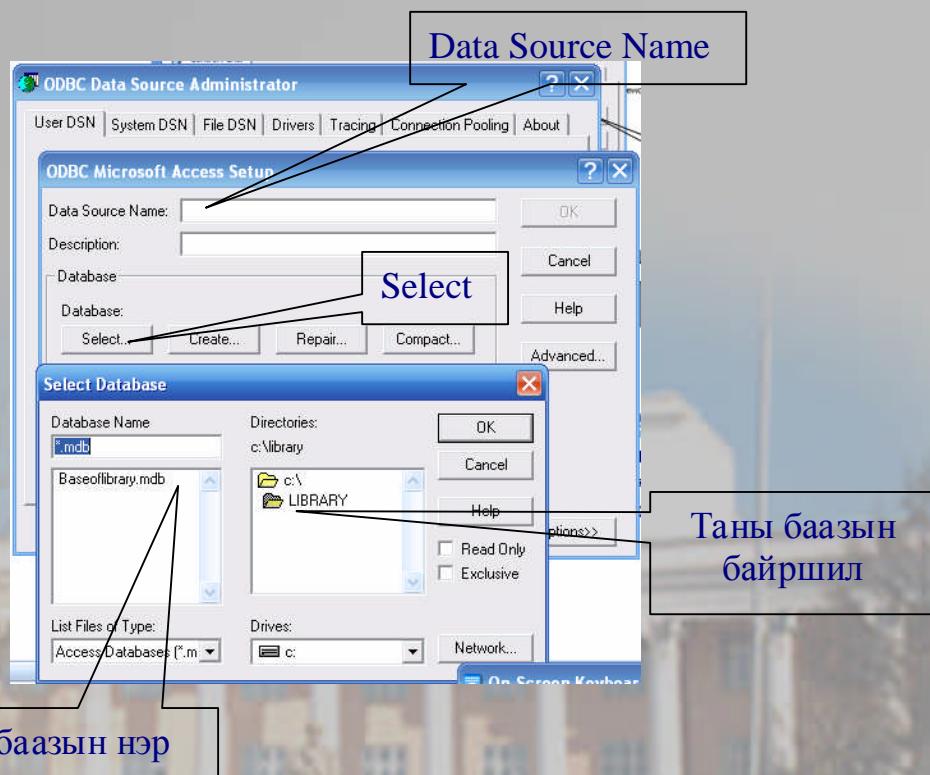
1. Access програм дээр өөрийн програмд ашиглагдах өгөгдлийн баазын үүсгэнэ.
2. Start меню -> Control Panel -> Administrative Tools -> Data Sources (ODBC)-ыг сонгож орно.
3. Үүсэх ODBC Data Sources Administrator цонхноос Add командыг сонгоно.
4. Create New Data Source харилцах цонх нээгдэх ба Driver do Microsoft Access (*.mdb) – сонгон идэвхижүүлж Finish товчлуурыг дарна.



5.

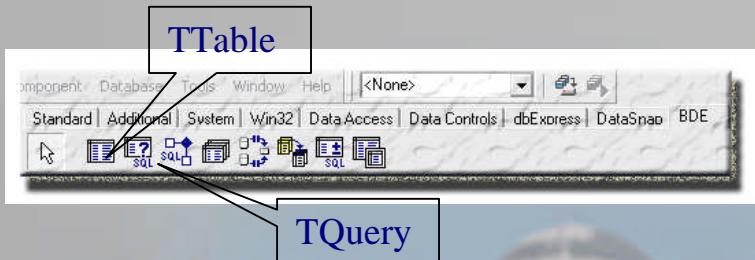
Та хэрвээ үйлдлийн зөв гүйцэтгэсэн бол ODBC Microsoft Access Setup цонх нээгдэнэ. Энэ цонхны Data Source Name талбарт та өөрийн дуртай нэрийг өгч болох бөгөөд энэ таны alias-ийн нэр болно.

6. Description талбар та хэрвээ шаардлагатай гэж үзэвэл товч тайлбарыг бичиж болно. Бичихгүй байсан ч болно.
7. Database талбарын Select товчлуурыг дарж үүсэх Select Database цонхонд та өөрийн 1-р үйлдэл дээр үүсгэсэн Access-ийн баазыг сонгоод ok товчлуурыг дарна.



8. Та дээрх үйлдлийг амжилттай гүйцэтгэсэн бол иргээд ODBC Microsoft Access Setup цонх гарч ирэх ба дахин ok товчлуурыг сонгоно.
9. Та үйлдлүүдийг амжилттай гүйцэтгэлээ. Та дээрх үйлдлүүдийг зөв гүйцэтгэсэн эсэхийг ODBC Data Source Administrator цонхноос харж болно. Хэрэв таны үйлдэл зөв алдаагүй хийгсэн бол энэ цонхонд Data Source Name талбарт өгсөн нэр буюу alias-ийн нэр тус цонхонд нэмэгсэн байгааг харж болно.
10. ok товчлуурыг дарж үйлдлийг дуусгана.

Мэдээллийн сангийн компонентууд



TTable компонент

TTable компонент нь мэдээллийн сангийн хүснэг буюу table-тай богино хугацаанд, хялбар холбогдож ажилладаг ба энэ хүснэгтээр өгөгдлийн баазын хүснэгтийг програмаас удирдаж болно. Та TTable компонентийг ашиглахын тулд *DatabaseName*, *TableName*, *Active* шинж чанаруудад тохирох утгуудыг олгох хэрэгтэй.

DatabaseName: Дээр үүсгэсэн Alias-ийн нэрийг зааж өгнө.

TableName: Өгөгдлийн баазаас холбогдох хүснэгтийн нэрийн зааж өгнө.

Active: Table-ийг идэвхижүүлэх үүргийг гүйцэтгэнэ. *DatabaseName*, *TableName* талбарууд заавал утгаа авсан байх ёстой бөгөөд сонголтыг true болно.

Table – шинж буюу properties-үүд

IndexDef: хүснэгтийн индек буюу түлхүүр талбариудын тухай мэдээллийг агуулна.

IndexFieldCount – тухайн хүснэгтэнд хэдэн түлхүүр талбар байгааг тоолно

IndexFieldNames – түлхүүр талбариудын нэр

Indexname – Хүснэгт дээрх хоёрдогч түлхүүр талбарыг тодорхойлно

MasterFields – Холбоо хийнэ

ReadOnly – Хүснэгтийг зөвхөн уншихаар болгоно.

Индексийн тухай

Хүснэгтээс мэдээлэл хайх, хүснэгтийн мэдээллүүдийг эрэмбэлэх үүднээс түлхүүл буюу индекс талбарыг үүсгэдэг. Индекс талбарыг үүсгэснээс хүснэгтээс мэдээлэл хайх, өгөгдлийн эрэмбэлэх, эмх цэгтэй хайх зэрэг үйлдлүүдийг хялбаршуулж өгдөг.

MasterSource, MasterFiled шинжүүд

Бидний ашилаж байгаа өгөгдөл буюу хүснэгтүүд нь хоорондоо холбоотой байдаг. Жишээлбэл Бат, Дорж, Дондог гэдэг 3 оюутантай ангид Ариунаа багш хичээл ордог гэж бодьё. Гэхдээ Ариунаа багш зөвхөн Дорж болон Дондогт л хичээл заадаг тул хичээлийн эрхлэгч асуувал Ариунаа багш би зөвхөн Дондог, Дорж хоёрт л хичээл заадаг хэмээн энэ ангиас буюу энэ хүснэгтээс Дорж, Дондог хоёрыг ялгаж авах хэрэгтэй болдог. Өөрөөр хэлбэл шүүлт хийдэг хэмээн хэлж болно. Энэ дохиолд Ариунаа багшнийг агуулсан хүснэгт нь MasterTable болох ба Бат, Дорж, Дондогийг агуулсан хүснэг нь Detail table буюу MasterTable-р шүүгддэг.

Table – компонентийн үндсэн функцууд

AddIndex – Хүснэгтэнд шинээр индексийг үүсгэдэг

Create table – шинээр хүснэгтийг үүсгэдэг

DeleteIndex – хоёрдогч индексийг устгана.

DeleteTable – Хүснэгтийг устгана.

EditKey – хүснэгтийг dsSetkey төлөвт оруулж түүнээс хайсан бичлэгүүдийг засварлах боломж олгодог.

FindKey – Индекстэй талбараас бичлэгийг хайж олоод курсорыг байрлуулна.

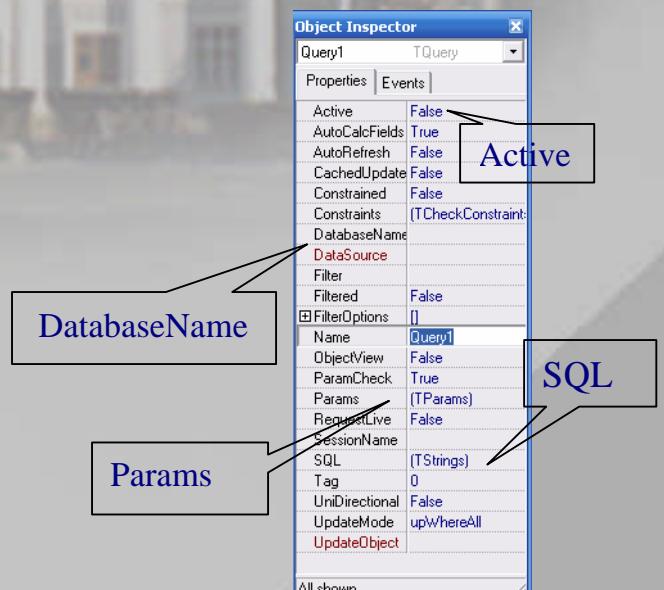
GotoKey – Findkey-тэй ижил үүргийг гүйцэтгэнэ.

GotoNearestKey – Индекстэй ижил буюу ойролцоо утгатай бичлэгийг хайж олоод курсорыг байршуулна.

Setkey – Хүснэгтийг dsSetKey төлөвт оруулна. GotoKey8 GotoNearest функцуудтэй цуг хэрэглэгдэнэ.

TQuery компонент

Delphi-ийн өгөгдлийн сангийн програмчлалд бид TQuery компонентийг ашиглаж баазаас буюу хүснэтээс мэдээллийг шүүж, ялгаж авч болдог. TQuery-ийг баазтай холбож өгдөг онцлог шинжүүд нь бол DatabaseName, SQL, Active; шинж чанарууд юм. DatabaseName болон Active шинжүүд нь TTable-ийн DatabaseName болон Active-тай адилхан үүргийг гүйцэтгэдэг бөгөөд SQL шинж нь онцгой. Учир нь SQL шинжид бид SQL кодыг бичилтийг хийж query-ийн хийх үйлдлийг тодорхойл өгдөг. Жишээлбэл: бидэнд дараах хүснэгт өгөгджээ.



Сонирхол

Сурагчийн нэр	Хүйс	Сонирходог хичээл
Бат	Эр	Маркетинг
Гэрлээ	Эм	Менежмент
Дорж	Эр	Монголын түүх

Тэгвэл бид дээрх хүснэгтээс зөвхөн эрэгтэй хүүхдүүд ямар хичээл сонирхдогийг шүүж ялган бичвэл

Select сурагчийн нэр, сонирходог хичээл
Form сонирхол
Where хүйс=p1

Тайлбар:

Select – Гэдэг нь сонгосон талбараудыг дэлгэцэнд хэвлэдэг функц

Form – Аль хүснэгтээс дуудахыг заана

Where – тухайн нөхцлийг бичнэ

P1 – query – ийн параметр буюу хувьсагч энэ жишээн дээр энэ хувьсагч нь эр болон эм гэсэн хоёр л утгыг авах зориулалттай.

Хэрэв бид p1=эр гэж утга өгвөл хүснэгт маань

Сурагчийн нэр	Сонирходог хичээл
Бат	Маркетинг
Дорж	Монголын түүх

Харин бид p1=эм гэж утга өгвөл

Сурагчийн нэр	Сонирходог хичээл
Гэрлээ	Менежмент

Болно. Бид Select талбарт зөвхөн Сурагчийн нэр, Сонирходог хичээл талбaryг л бичиж өгсөн учир зөвхөн энэ хоёр талбaryг хэвлэж Хүйс талбaryн хэвлэхгүй. Магадгүй бидэнд олон талбартай хүснэгтээс бүх талбaryг нь дэлгэцэнд хэвлэж гаргахыг хүсвэл Select * буюу '*' тэмдгийг тавьхад л хангалттай. Тэгвэл Run-Time-ийн үед query-ийн параметрт дараах байдлаар утга онооно.

```
Query1.Close;  
Query1.ParamByName('p1').Value:=edit1.text;  
Query1.Open;
```

(жич: та Query ашиглахыг тулд SQL билтийн талаар ойлготтой байх хэрэгтэй)

TField компонент

TField компонент нь өгөгдлийн баазын хүснэгтийн нэг талбaryг төлөөлж байдаг. TField ашиглаг талбaryн шинж чанарыг олгож болно. Design Time-ийн үед TTable, TQuery хулганаар хоёрт тогшиход Fields Editor цонх үүсдэг. Энэ цонхон дээр хулганы баруун товчыг дараход үүсэх Add Field командаар TField-ийг үүсгэдэг. Fields Editor цонх дээр нэмсэн TField талбараудаас нэгийг сонгоход түүний шинж чанар Object inspector цонх дээр үзэгдэнэ. Тухайлбал: FieldName, Fieldking, Index, Size... гэх мэт.

Тайлан Rave5.0

Rave тайлан бэлтгэх програм нь Delphi 7-с өмнөх хувилбаруудад байдаггүй. Rave програм нь Delphi 7.0 програмыг суулгахад дагаад суудаг болно. Rave программыг Tools зэсний Rave Designer командаар ажиллуулна. Rave програм дээр бэлтгэгдсэн тайланг Rave tabs-ийн компонентуудаар програмтай холбогдож ашигладаг. Жишээ болгож Rave-р тайланг үүсгэж үзье.

- 
1. Шинээр project үүсгээд Table компонентийг форм дээр байрлуулна. Table компонентийг өөрийн баазтайгаа холбон Active шинж чанарыг true болгоно.
 2. Rave tabs-с TRvDataSetConnection компонентийн форм дээр авч тавиад dataset шинж чанарт нь table1-ийг олгоно.
 3. Rave Visual Designer-ыг ашиглан Rave project файлыг дараах байдлаар үүсгэнэ.
 - a. Delphi-ийн Tools зэсний Rave Designer-ийг сонгож rave программыг ажиллуулна.
 - b. File зэсний New data Object-ыг сонгоход Data Connection харилцах цонх үүснэ.
 - c. Data Object Type жагсаалтаас Direct Data View –г сонгоод next товчийг дарна.
 - d. Active Data Connection жагсаалтаас RVDataSetConnect1-ыг сонгоод finish товч дарна.
 - e. Tools зэсний ReportWizard\Sample Table-ыг дараахад Simple Table wizard цонх үүснэ.
 - f. DataView1-ыг сонгоод Next товчийг дарна.
 - g. Тайланд үзүүлэх талбариудыг сонгоод Next товчийг дарна.
 - h. Талбариудыг үзүүлэх дараалал, Report Title болон Report margin, фонтыг сонгох цонхнуудаас өөрийн сонголтоо хийнэ.
 - i. Wizard хуудасны төгсгөлд Generator товчийг дараахад Page Designer цонх дээр тайлангийн интерфэйс үзэгдэнэ.
 - j. F9 дарж шалгаад, шаардлагатай гэвэл тайлангийн интерфейсийг зас.
 - k. File зэсний Save As командаар *.rav проект файлыг Delphi дээр үүсгэсэн проектын директорд хадгална.
 - l. Delphi програмд буцаж орно.
 4. Rave tabs-с TRvProject компонентийг форм дээр байрлуулна.
 5. TRvProject-ын object inspector цонх дээр үзэгдэх ProjectFile шинж чанарт 3-р алхмаар үүсгэсэн .rav файлыг зааж өгнө.
 6. Standars tabs-с Button авч тавиад onClick үзэгдлийг үүсгэ.
 7. OnClick үзэгдэлд RvProject компонентийн Execute функцыг дуудаж байгаа үйлдлийг тодорхойлно.
RvProject1.Execute;
 8. F9 дарж программыг ажиллуулаад товч дээр дараахад таны хүснэгтээр үүссэн тайланг үзэх болно.

Лабораторын ажил

Dictionary хийх

Толь бичиг хийх дараалал:

Толь бичгээс хайлт хийх

1. Access програм дээр English, Mongol гэсэн хоёр талбартай хүснэгтийг үүгэнэ.
2. ODBC alias-ийг үүсгэнэ.
3. Delphi 7.0 программыг ажиллуулж шинэ project файл үүсгэнэ.
4. Form дээрээ BDE tabs-c Table компонентийг авч танин Databasename, TableName-ийг зааж өгөөд active-ийг true болгоно.
5. DataControls tabs-c DBGrid, Data Access tabs-c DataSource компонентийг авч тавина.
6. DataSource1 компонентийн DataSet шинж чанарт Table1-ийг зааж өгнө.
7. DBGrid1 компонентийг DataSource шинжид DataSource1 холбож өгнө.
8. Standard tabs-c Edit- компонентийг форм дээр авч байруулна.
9. Edit1 компонентийн OnChange үзэгдэлд орон дараах кодыг бичнэ.
Table1.Locate('English',Edit1.Text,[LoPartialKey]);
Тайлбар: хүснэгийн English талбараас edit1.text дээр бичсэн бичлэгийг хайна.

Толч бичигт засвар оруулах

10. Form дээр 2 Edit, 1 Button нэмж байрлуулна. Button-ний caption-д ‘edit’ бичиж оруулна.
11. Edit2 компонентийн caption шинжид ‘English word’, Edit3 компонентийн Caption шинжид ‘Mongol word’ хэмээн бичиж өгнө.
12. DBGrid компонент дээр дарж идэвхижүүлээд object inspector цонхноос OnDblClick Үзэгдэлд орон дараах кодыг бичнэ.

```
Edit2.text:=DBGrid1.Fields[0].AsString;  
Edit3.text:=DBGrid1.Fields[1].AsString;
```

Тайлбар: програм ажиллаж байх үед засах гэж буй мөрөн дээр хоёр дараад тухайн хүснэгтэнд харагдаж байсан утгууд Edit3 болон Edit4 дээр гарна. Тэгээд та өөрийн засах гэсэн зүйлээ засна.

13. Button1-ийн OnClick үзэгдэлд орон дараах бичилтийг хийнэ.

```
table1.Edit;  
Table1.FieldByName('English').AsString:=Edit2.Text;  
Table1.FieldByName('Mongol').AsString:=Edit3.Text;  
Table1.Post;
```

Тайлбар: хүснэгтэнд өөрчлөлт оруулахын тулд заавал хүснэгтийг Edit горимд оруулах ба хүснэгттэй ажиллаж дууссан бол ба заавал post хийж Edit горимоос шилждэг. Энэ товчлуур дээр дараад таны 12-р үйлдэлд дээр хийж заасан өгөгдөл чинь баазад, хүснэгтэнд хадгалагдана.

Толь бичгэнд үг нэмэх

14. Standard tabs-c Button-г авч форм дээр байрлуулна.
15. Button-ний caption шинжид ‘insert’ бичиж оруулна.
16. Button-ний OnClick үзэгдэлд нэвтрэн орон дараах кодыг бичнэ.

```
table1.Edit;  
Table1.Insert;  
Table1.FieldByName('English').AsString:=Edit2.Text;  
Table1.FieldByName('Mongol').AsString:=Edit3.Text;  
Table1.Post;
```

Тайлбар: Table1.insert функ нь хүснэгтэнд шинээр бичлэг нэмэх ба Edit2, Edit3 – р бичигдсэн өгөгдлүүд баазад нэмэгдэнэ. Магадгүй та хүсвэл Edit2, Edit3 өөр компонент ашигласан ч болно.

17.F9 товчлуурыг даран програмыг ажиллууна.



Бидний хаяг

ЭЗО - СМЗ - 3

Д. Баярбат

Brbt_ezo@yahoo.com

Утас: 99678933

МКС - ПХ - 3

Э. Цог - Эрдэнэ

Tsogoo_enkh@yahoo.com

Утас: 99666566

МКС - МСМ - 3

Б. Батбаатар

Btta_1@yahoo.com

Утас: 99668315