ЛЕКЦИЯ № 2

Тақырыбы: Объектіге бағытталған тілдердің негізгі концепциясы. Объектіге бағытталған тілдердің (Object Pascal, C++, Java, VBasic, SmallTalk және т.б.) негізгі конструкциялары.

Лекция жоспары:

1. Объекті-бағытталған программалау тілдерінің негізгі концепциясы
2. Программа құрылымы
3. Тәсіл. Модуль құрылымы

Лекция мазмұны

1. Объекті-бағытталған программалау тілдерінің негізгі концепциясы

Объекті-бағытталған программалау тілдерінің негізгі концепциясы– құрылатын қосымша өзара байланысқан негізгі объектілерден тұрады. Объекті-бағытталған технологияда қолданушы үш базалық элементпен: объектілер, хабар және класстармен жұмыс істейді. Объектілер дегеніміз бірнеше рет қолданылатын программалық модулдерден, яғни байланысқан мәліметтер мен процедуралардан тұрады. Объект құрылымы екі бөліктен тұрады: айнымалылар және әдістер. Әдістер объект функциясының алгоритмін анықтайтын процедуралар мен функциялар жиынынан тұрады. Объектілі айнымалылар жәй мәліметтерден (сан, массив, текст) және күрделі құрылымды информациялардан (график, дыбыс т.б.) тұрады. Объектілердің өзара байланысуына хабарлар қолданылады және үш бөлімнен тұрады: объект идентификаторы, ағымдағы объектіде қолданылатын әдіс аттары және таңдалған әдіс режимін қалпына келтіретін қосымша информациялар. Күрделі программалар бірнеше біртипті объектілерді қолдануы мүмкін. Бұл жағдайда әр объект үшін әдістер мен айнымалылар туралы информацияны жазу тиімсіз. Бұл мақсатқа объектілер класы деген түсінік енгізілген. Класс дегеніміз біртипті объектілерге арналған шаблон және объектілі айнымалылар типтері мен әдістерін анықтайтын информациялардан тұрады. Объекті-бағытталған технологияға негізделген программалау тілдері: SmallTalk/v, Object Pascal, ACT++, C++, Simula, Actor, Classic–Aga және т.б. **Объекті-бағытталған программалаудың негізгі үш принципі бар**: *инкапсуляция, тұқымқуалау, полиморфизм.*

2. Программа құрылымы

Delphi-де **программа** (**проект**) екі бμлімнен тұрады. Алғашқы автоматты түрде **project1** атауы берілетін *проект файлы (негізгі модуль)* және**unit1. pas** атауы берілетін *модуль.* Олар жеке терезелерде орналастырылған. *Модульге* оқиғаларға сәйкес іс – әрекеттерді орындайтын программа мәтіні (*процедуралар*) енгізіледі (олар 1.4 –

тақырыпта кең түрде қарастырылған). Программа мәтінін **программалық код** деп, терезені **программалық код терезесі** не қысқаша **редактор терезесі** деп те атайды. 1.1-тақырыпта ескертілгеніндей, Delphi іске қосылған кезде ол форма терезесінің астында кәрінбей тұрады. Он экранға шығару тәсілдері:

- форманы жабу (жабу түймесін шерту);
- код терезесінің бір шеті форма астында кәрініп тұрса, оны шерту.

Терезе белсендірулі түрде ашылады да, онда *процедура дайындамасы* (үлгісі) кәрінеді. Оның тақырыбы нүкте арқылы бәлінген класс және процедура атауларынан тұрады (1.5-сурет). т.б.

Жалпы формадан код терезесіне әту және код терезесінен формаға әту үшін **F12** клавишін басу желкілікті. 1.5 – суретте кәрініп тұрғаны – код терезесіне енгізілген *процедура дайындамасы.* Оқиғаға байланысты құрылған процедура **оқиғаны әңдеуіш** не оқиғаны әңдеу процедурасы делінеді. *Процедура* дайындамасының жазылу түрі:

Procedure <атау> **(Sender: TObjiect);**

Сипаттау бәлімі

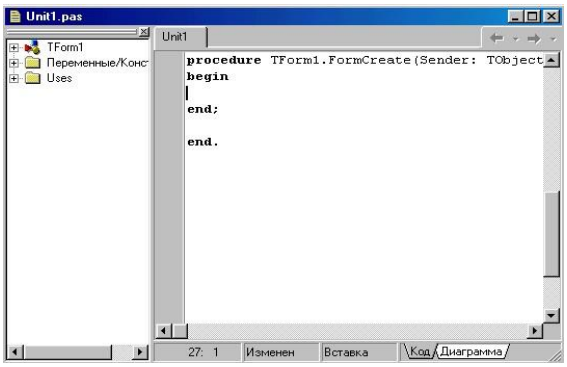
Begin

Процедура денесі

End;

Мұндағы **Sender** параметрі құрылатын процедураның қай класна тиістілігін анықтайды.

Терезенің сол бәлігіндегі – *браузер* терезесі. Онда код терезесінде барлық жарияланулардың құрылымын кәріп шығуға болады.



Сурет 2 - Редактор терезесі. Онда көрінген процедура дайындамасы

Кез келген программа **program** сәзінен басталатын проект файлы мен бір не бірнеше модульдерден тұрады да, қолдан құрылатын программа модуль ішіне енгізіледі. Проект файлы **dpr**, модуль **pas** кеңейтілуі бойынша сақталады. Проект файлын **негізгі модуль** деп атайды. *Негізгі модульдің мазмұны* проектінің жалпы

сипаттамасынан тұрады. Delphi іске қосылған кезде он ол *автоматты түрде дайындап шығады.* Негізгі модульге ерекше жағдайлардан басқа кездері қосымша нұқсқаулар енгізулің қажеті жоқ. **Негізгі модуль (проект):**

```
Program Project1;
Uses
  Forms
   Unit in —Unit1. pas‘ {Form1};
  {$R*.RES}

begin
  Application. Initialise;
   Application. CreateForm(Tform1, Form1);
  Application. Run;
End;
```

Мұндағы,

Progect1 – негізгі модуль (проект) атауы. Проектіні дайындап, жаңа атау бойынша сақтаған кезде ол соңғы атауға алмастырылып қойылады;

Uses (қолдану) – Турбо Паскальдағы сияқты, қызметші сμз. Оның соңына программада пайдаланылатын стандартты (кітапханалық) *Forms* модулі мен Delphi-дің модульге алғашқы рет меншіктеген атауы. (*Unit1*) жазылған Unit1-ден соң оның қайсы модульдік файлда құралатыны (*In ‘Unit1. pas’*) жєне онымен байланысты форма атауы кμрсетіледі. ({*Form*}).

{**$R*.RES**}-нұсқау. Ол компиляторға *қосымшаның* (Windows кμмегімен дайындалған қолданбалы программаның) қор сипаттамаларын, мысалы, шартбелгілер (пиктограммалар) сақталған файлды жєне т.с.с. пайдалану керектігін кμрсетеді (файлдың кеңейтілуі -.**res**);

Begin – end операторларының аралығына енгізілген. Соңғы бμлім- қосымшаның алғашқы жүктелуін қамтамасыз ететін тәсілдер (Delphi-де арнайы іс-єрекетті орындайтын *процедура, функция жєне командалар* **тәсілдер** делінеді):

Application.Initialize – қосымша обьектісін инициалдау (программаны алғашқы рет дайындау) *тәсілі;*

Application.CreateForm – проект құрамына енетін форманы дайындап, экранда кμрсету *тәсілі* (create-құру);

Application.Run – программаны іске қосуды қамтамасыз ету *тәсілі.*

3. Тәсіл. Модуль құрылымы

Delphi-де тәсілдің командалық түрде жазылуы:
<Обьект>.<Тәсіл>

Мысалы, **Application.Initialize –** Application обьектісінің *Initialize* тәсілін орындау.

Кейбір жағдайда бµлімге проект сақталатын бума атын меншіктеу командасын қосып қою да мүмкін, т.б.

Жалпы, Delphi-де программаның орындалуы автоматты түрде негізгі модульді орындаудан басталады.

Модуль-түрлі іс-єрекеттерді орындауға арналған программа бµлімі. Модуль тақырыбы **Unit** (модуль) қызметші сµзінен басталып, соңына єдеттегідей нүктелі үтір (;) таңбасы мен аяқталатын модуль атауы жазылады. Delphi-дің модульге алғашқы рет меншіктейтін атауы- **Unit1.** Жаңа проект ашылған кезде модуль дайындамасы да автоматы түрде құрылады:

```
unit Unit1;
Interface
Uses
    Windows, Messages, SysUtils, Clfsses,
    Graphics, Controls, Forms, Dialogs;
Type
  Tform1=class(TForm)
 Private
   {Privare declarations}
 public
   {Public declarations}
 end;
var
   Form1: Tform1;l
Implementation
{$R*.DFM}
end.
```

Інтерфейс (interface) бμлімі **interface** кілттік сμзімен басталады да оған бμлімдер енгізіледі: **uses -** Турбо Паскальда пайдаланатын бμлім сияқты, оған стандартты модуль атаулары жазылады, бμлімге пайдаланушы пайдаланған модуль атауын кірістіріп қоюы да мүмкін. Одан єрі, Delphi дайындалған форма *типі* сипатталады (онда *μрістер, қасиеттер, компоненттер* сипатталып, олардан соң модульде жазылатын *процедуралар* мен *функциялар (программа элементтері*) жарияланады, т.б.).

Private (жеке, дербес**)** бμліміне тек ағымдық модульге тиісті элементтер енгізілуі мүмкін (*элемент* - μрістер, тәсілдер, қасиеттер мен оқиғалар); **Public** (кμпшілік) бμлімінің ішінде ағымдық модульге қол жеткізуге болатын кез келген программа не модульдің кμрінетін элементтері, облыстары енгізіледі. Олар класқа енетін элементтердің пайдалану облыстарын ғана анықтайтын болғандықтан, єдетте (кμп жағдайда) олар бос кμрінеді.

Implementation (іске асыру, орындау) бμліміндегі **{$R*.DFM}-.dfm** кеңейтілуі бойынша жазылған файлды пайдалану нұсқаукы. Ол модульді сєйкес форманың сипаттамасымен байланыстырады (файла форма *қасиеттерінің мєндері* жазылып қойылған. Ол формаға қойылған компоненттер қасиеттерінің де сипаттамаларын бойында сақтайды. Қасиеттер сєйкес **Objiect Inspector** *терезесінде кμрінеді).* Одан соңғы қатарларға программалаушы Delphi тілінде қажетті *процедураларды* қолдан кірістіру керек. Олардың ішіндегі оқиғаны μңдеуіш процедуралардың тақырыптары модульдің **интерфейс** бμлімінде автоматты түрде жазылып қойылады, мысалы, 3.7.4, 3.11-тақырыптары μңделетін арнайы информация үшін толық модульдер құрылған.

Кейде модульдің соңына *инициалдау (initialization) бμлімі* енгізіледі. Бμлім модуль айнымалыларын инициалдау (бастапқы мєндер беріп), программаны дайындау үшін қажет. Егер де толтырылса, бұл бμлім басқаруды программа денесіне беру ден бұрын орындалады. Бμлім нұсқауларын **begin** жєне **end** кілттік сμздерінің арасына енгізу керек. Жоғарыда кμрсетілген сиқты, бμлім толтырылмаса, **begin** сμзі жазылмай, оған тек **end** μзі енгізіледі. Ол – модульдің соңын білдіретін кілттік сөз.

Бақылау сұрақтары

1. Объектіге бағытталған тілдердің негізгі концепциясы?
2. Объектіге бағытталған тілдердің негізгі конструкциялары?

Ұсынылатын әдебиеттер

1. Бадд Т. Объектно-ориентированное программирование в действии. Питер. 1997.
Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++, 2-е изд./Пер. сс англ. –М.: «Издательство Бином», Спб.: «Невский диалект», 2001. 3. Бьярн Страуструп. Язык программирование С++. Киев: Диасофт, 1993. 1,2 часть.