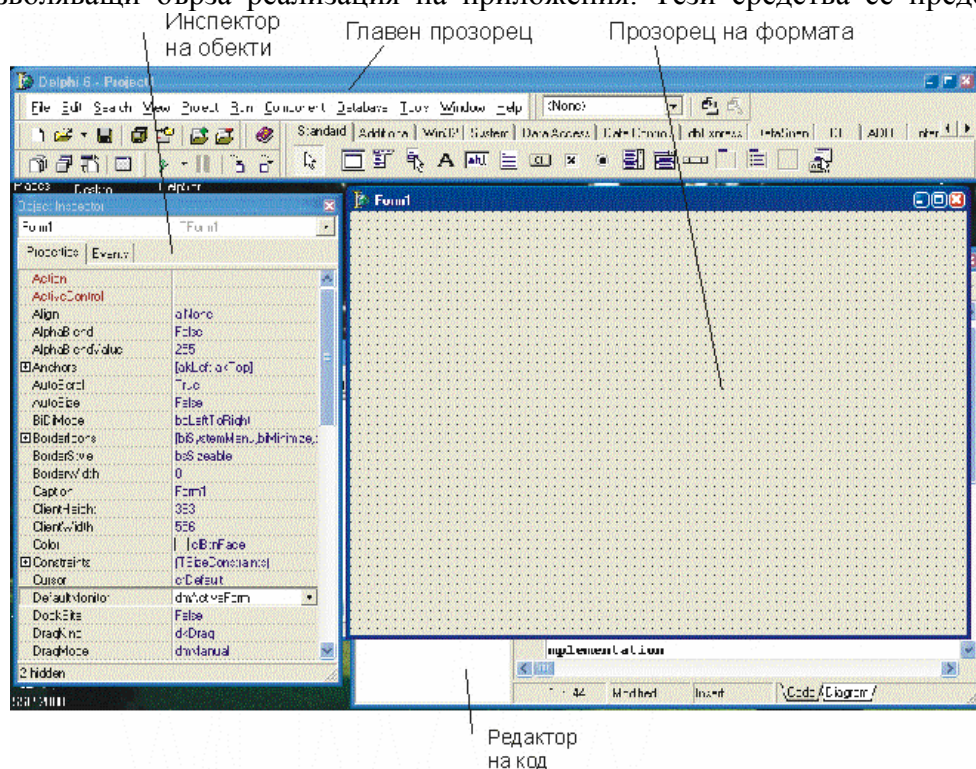


Лекция 2

1 Интегрирана програмна среда Delphi

Програмирането в графични среди (Windows) изисква доста опит и познания. Интегрираната програмна среда Delphi подпомага програмирането в графичната среда, като скрива голяма част от подробностите извършвани при програмирането на ниско ниво. Въпреки че извършва голяма част от програмирането, тази среда все пак не може да програмира сама програмистите. Това което прави много привлекателна тази програмна среда е, че улеснява усилията на разработчиците на програмни продукти, като ги освобождава от писането на досадни еднообразни програмни елементи свързани с оформление на интерфейса на програмите. За програмистите остава да напишат само специфичната за конкретното приложение част от софтуера.

Средата за програмиране Delphi предлага един гъвкав и съвременен набор от средства, позволяващи бърза реализация на приложения. Тези средства се представят посредством



Фиг.6.1 Програмна среда Delphi

четири прозореца, управлявани като многопрозоречно приложение: главен прозорец, инспектор на обекти, прозорец на формата и редактор на програмния код (Фиг.6.1).

Главният прозорец управлява останалите елементи от програмната среда. Когато той е минимизиран или затворен с него се минимизират или затварят и останалите прозорци. Той се разполага в горният край на работното пространство върху екрана и съдържа главно меню, ленти с инструменти и палитра (палет) с компоненти. Посредством тези елементи на основният прозорец, се предоставят богати функционални възможности за разработване на сложни приложения.

Инспекторът на обекти (**Object inspector**) по подразбиране е разположен в долният ляв ъгъл на работната област на Delphi. Инспекторът на обекти съдържа две страници, отбелязани с етикети: страница със свойствата (properties) и страница със събитията (events) за отделните компоненти (обекти). Страницата със свойствата показва достъпните свойства на обекта, който е маркиран (избран) в прозореца на дадена форма. Когато не е избран компонент, в

прозореца на Object inspector, се показват свойствата на активната форма. Страницата на събитията, показва тези събития на които избраният обект може да отговаря (реагира).

Прозорецът на формата (**Form Designer**) по подразбиране е разположен в дясната част на работното пространство. В него се съдържа информацията, която е необходима на потребителя за да взаимодейства с приложението. По време на разработване на програмно приложение, тази част от работното пространство на Delphi е основната работна област. Тук се извършва проектирането на интерфейса и се оформят основните структурни и функционални взаимодействия между отделните модули на приложението. То може да се състои от няколко форми, които проектантът може да разработва последователно или едновременно, като организира взаимодействието между тях съгласно алгоритъма и логиката на задачата. Всяка програма трябва да има минимум една форма.

Четвъртият елемент от програмната среда е редакторът на програмен код (**Code Editor**). Този прозорец позволява на програмиста да пише необходимият програмен код (процедурите за обработка на събитията). Програмният език подържан от Delphi е Object Pascal. Прозорецът на редактора на програмен код е многостраничен. Чрез съответен етикет с името на програмния файл, може да се избира съответна страница с програмен модул.

В Delphi могат да се разработват еднократно приложения (Single document interface – SDI) и многократно приложения (multiple document interface – MDI). Програмите, които имат многократно интерфейс (например Microsoft Word) се състоят от “родителски” прозорец, включващ един или няколко дъщерни прозореца. В MDI, дъщерните прозорци са ограничени от областта на родителския прозорец и не могат да се разполагат извън границите му.

При SDI работата с прозорците протича по друг начин. За разлика от програмите с MDI, SDI-приложенията нямат дъщерни прозорци. Те се състоят от множество “родствени” прозорци. Тук има един главен прозорец, за който не е задължително да съдържа други. Обикновено основният прозорец съдържа лентата с основното меню и лента с бутони, както и някои основни графични и текстови елементи описващи основното предназначение на програмата.

А. Главен прозорец на интегрираната програмна среда Delphi

Посредством главният прозорец на Delphi се осъществяват основните функции на управление на процеса на проектиране и разработване на програмни приложения. Тази дейност се извършва с помощта на главно меню, лента с инструменти и палитра с компоненти, намиращи се в главният прозорец.

- **Главно меню.** Менюто в Delphi е статично, т.е. не се изменя при преход от един прозорец към друг. То позволява да се организира работата от едно място. Посредством главното меню, програмната среда може да се настрои според желанията на потребителите. Освен статичното главно меню, програмната среда Delphi използва и така нареченото контекстно меню. То се появява при натискане на десният бутон на мишката. Нарича се контекстно, защото съдържанието на елементите в него се определя от местоположението на показалеца на мишката. Когато показалецът се намира в главният прозорец, контекстното меню съдържа елементи за конфигуриране на главния прозорец, когато се намира в прозореца на формата то съдържа елементи за редакция на обектите във формата, когато е в прозореца за програмния код – елементи за редакция на кода и т.н. Общо казано, съдържанието на контекстното меню се определя от местоположението на показалеца на мишката (контекста).

- **Панел с бутони за бърз достъп.** Панелът за бърз достъп (Speed bar) представлява редица бутони, които осигуряват бърз избор на елементи от менюто с помощта на мишката. Този панел може да се настрои по вкуса на всеки потребител, посредством добавяне или премахване на бутони от панела. Конфигурацията на Speed bar панела се извършва лесно с помощта на команда Configure достъпна от контекстното меню на Delphi, когато показалеца на мишката се намира върху панела Speed bar.

- **Палитра с компоненти.** Палитрата с компоненти представлява каталог на визуалните и приложни обекти, които могат да се включат в потребителската форма на приложенията. Delphi разполага с голямо количество компоненти, които за улеснение на работата с тях са разделени в групи. Всяка група компоненти е представена в отделна страница от палитрата на компоненти. Същността на обектно-ориентирания подход на Delphi се състои в наличието на множество компоненти, които се явяват средствата, за осигуряване на достъп до стандартните възможности на Windows. В Delphi тези възможности се реализират чрез компоненти, “опаковани” в потребителските форми. За да се поставят необходимите компоненти в потребителската форма, те трябва просто да се пренесат от палитрата с компонентите във формата. Поместеният във формата компонент става истински обект, чийто свойства могат да се редактират и който е готов за изпълнение на команди и инструкции.

Палитрата с компоненти е инструмент с широки възможности, който се използва за поставяне на обекти в потребителската форма. За удобство на потребителите, Delphi предоставя възможност за адаптация на палитрата – промяна на разположението на компонентите, добавяне на нови компоненти и отстраняване на съществуващи.

Б. Инспектора на обекти (Object Inspector)

Инспекторът на обекти е в основата на всички действия и манипулации с обектите в Delphi. Прозорецът на Object Inspector съдържа следните елементи: комбиниран панел (Combo box) за избор на обект; страница със свойства (Properties Page) и страница на събитията (Events Page). Страницата със свойствата съдържа полета със стойности на свойствата на текущо избрания обект, а страницата на събитията съдържа полета с връзките между програмният код и вградените събития за текущо избрания обект.

От комбинираният панел (Combo box) може да се избере обект от текущата форма за който да се разглеждат свойствата или събитията. Обект може да се избере и посредством непосредственото му маркиране чрез показалеца на мишката във формата. Всеки път, когато се избира нов обект, съдържанието на свойствата и събитията в Object Inspector се променят и приемат стойностите на избрания обект.

- **Действия със свойствата.** Всички достъпни в Delphi компоненти имат точно определени свойства. Самата форма също има набор от свойства, които могат да се променят при проектирането или по време на изпълнение на програмата. С помощта на страницата със свойствата в инспектора на обектите може да се разглеждат свойствата на обектите зададени по подразбиране, включително и свойствата на формата. В лявата колона на страницата са разположени наименованията на свойствата (Caption – заглавие, Font - шрифт, Width – ширина и т.н.), а в дясната – стойностите им. Свойствата по подразбиране съществуват заедно с компонента, докато той се намира в палитрата с компоненти. Когато даден компонент се разположи във формата, неговите свойства могат да се променят в зависимост от изискванията на задачата. Промяната на свойствата се извършва много лесно – просто се променя стойността в инспектора на обектите.

В Delphi всеки обект притежава свойството Name, което определя наименованието на обекта. Когато се поставя компонентът във формата, по подразбиране на него се присвоява уникално име, съответстващо на поредността на поставянето му във формата. Например, на елемент от типа поле за текст се присвояват наименования *Edit1*, *Edit2* и т.н. по реда на тяхното поставяне във формата. За да се получи добро съответствие между обектите и тяхното предназначение е удобно да се зададат подходящи наименования на обектите. Например, добре е бутонът който предизвиква отпечатване на принтера да се преименува в *btnPrint* (или само *Print*), вместо подразбиращото се наименование *button1*. По време на разработване на програмите, често се налага обръщания към различните обекти (компоненти) с техните имена, затова доброто обмисляне на имената на обектите е от съществено значение (както обмисляне наименованията на променливите при стандартното процедурно програмиране).

Изменението на свойствата в процеса на проектиране и изпълнение на програмата се осигурява от операторите на Object Pascal, които присвояват нови стойности на свойствата при описанието на обектите.

- **Действия със събитията.** Множеството от събития по подразбиране е определено за всеки компонент, както и за самата форма. Реакцията на обекта при определено събитие се задава от специална процедура. Например при натискане на клавиш от клавиатурата “А” в момент, когато е позициониран (фокусиран) обект за поле за текст (*Edit*), този обект ще получи индикация за три събития: *OnKeyDown*, *OnKeyPress* и *OnKeyUp*. Поведението на тези събития по подразбиране е в това, че в съответствие със стандарта Windows ще проследи натискането на клавиша. Когато се натисне например клавиш “А”, обектът *Edit* по подразбиране прибавя символът “А” към съдържанието на областта в текстовото поле (или замества избрания текст него). Да предположим, че трябва да се реагира при натискане на клавиш “А” в обекта *Edit* за да се осъществи някакво действие – например да се подаде

```
procedure TNi_Form.Edit1KeyPress(Sender: TObject; var Key: Char);  
begin  
    { Your code is placed here }  
end;
```

звук сигнал. За да се обработи събитието, най-напред трябва да се избере (маркира) обекта *Edit*, който реагира на събитието. След това се отворя страницата Events в инспектора на обектите (Object Inspector). Двойното натискане на бутона на мишката върху събитието *OnKeyPress* предизвиква появата на специален блок с кода на процедурата за обработка на събитията. Delphi създава името на процедурата за обработка на събитието, комбинирайки името на обекта с името на обработваното събитие (без представката On). При задаване на името на обработчика на събития Delphi използва някои общи съглашения, които могат да се видят в примера.

В примера е показана процедура за обработка на събитие *OnKeyPress* за обект *Edit1* разположен във формата (прозорец) *Ni_Form*. Наименованието на процедурата се образува от наименованието на формата “*Ni_Form*” с добавена буква T и отделен от другата част с “.” и името на обекта (компонент) “*Edit1*” и наименованието на събитието – *OnKeyPress* (без частицата On). Необходимият код, който трябва да добави програмиста се разполага между логическите скоби **begin** – **end**. Представеният в примера програмен фрагмент е подготвен от Delphi и за програмиста остава да напише само специфичния код, като използва синтаксиса на програмния език Object Pascal.

В. Прозорци на формата и редактор на кода

Прозорците на формата и на редактора на кода са тясно свързани помежду си. Както е известно, терминът модул (Unit) в езика Pascal се използва за файловете с изходен (source) код. За всяка форма Delphi създава отделен програмен модул (Unit), в който се записва информацията за всички обекти (компоненти) във формата и процедурите за обработка на събитията на които реагират компонентите.

Прозорецът на формата и редакторът на кода могат едновременно да са разположени в работното пространство или да се припокриват. Когато се натисне двукратно с показалеца на мишката върху някакъв обект, автоматично се активира редакторът на код в който директно може да се пише необходимият програмен код.

Приложенията могат да съдържат една или няколко форми. За всяка форма, програмната среда създава модул с изходен програмен код. Добавянето на нови форми за приложението се извършва много лесно – от главното меню File се избира опцията New Form и Delphi добавя нова форма с подразбиращи се свойства, които програмистът може да променя така, че да отговори на изискванията на разработваното приложение.

За разлика от формите, модулите с изходен код могат да нямат съответен файл за форма. Това се налага, поради необходимостта от създаване на програмен код, който да може да се използва в няколко приложения, за разлика от кода за обработка на събития, който е специфичен за всяка форма. Добавянето на нов програмен модул се извършва също толкова лесно, както и формите – посредством главното меню File опцията New Unit. В този случай Delphi създава нов файл с разширение “.pas”, в който може да се пише програмен код посредством редактора на код.

Г. Създаване на приложение с Delphi

Разработването на програмно приложение с Delphi се извършва посредством проект на приложението. За целта се създава специален проектен файл на ресурсите на проекта (project file – с разширение “.dpr”), в който се води инвентаризация на всички форми и програмни модули, от които се състои приложението. Първоначално този файл е празен, но с добавянето на всяка нова форма или програмен модул той се запълва. Delphi автоматично създава този файл, при започване на нов проект – от главното меню File и опцията New Project. Основни компоненти на приложението са формите – мястото, където се изгражда интерфейса на програмата. Създаването на приложенията се извършва в определена последователност. Най-напред се построява визуалният интерфейс. Това се постига чрез поставянето във формите на съответните компоненти – бутони, текстови полета, списъци, менюта и др. Посредством инспектора на обектите се установяват необходимите свойства за всеки компонент – текстове, цвят, размери, шрифтове и др. Към дадената форма се добавя необходимият програмен код, чрез създаване на процедурите за обработка на събитията.

Разполагането и манипулирането на компонентите се извършва по установения от графичната среда Windows начин. Разполагането на компонентите става, като съответният компонент се избере от палитрата с компоненти и след това се кликне с показалеца на мишката на определено място във формата. Манипулирането на компонента може да стане директно – маркиране и разтягане, свиване или влачене до определена позиция или индиректно – чрез инспектора на обекти – задаване на размери, местоположение, цвят и други.

Проектът на приложението е съвкупност от файлове, които работят заедно за създаване на самостоятелен изпълним файл (“.exe” файл) или библиотечен файл за динамично свързване (“.DLL” файл). Delphi съдържа специален мениджър на проекти, който позволява да се управляват изходните файлове за даден проект. Основните типове файлове които съдържа проект на приложение в Delphi са:

- Проектен файл (разширение “.dpr”). Този файл е с общо предназначение и съдържа обща информация за проекта. Той се създава и обслужва от програмната среда. Потребителят (програмист), не работи директно с този файл.

- Файлове за форми (разширение “.dfm”). За всяка форма използвана в приложението съществува по един файл за форма, който съдържа информация за свойствата и управлението на формата и обектите в нея.

- Файлове с програмни модули (разширение “.pas”). Това са файлове съдържащи програмния код на приложението. За всяка форма съществува по един файл с програмен код, но може да има и файлове за които не съществува форма. Основният модул на формата съдържа декларациите на класа и дефиницията на основния за формата клас. В един проект на приложение съществуват и някои други типове файлове, в които се записват конфигурационни параметри на работното пространство при разработката на приложението, както и компилираните модули на програмния файл и на ресурсите на приложението.

При компилиране на проекта, се компилират всички програмни модули, и ресурсните файлове на формите и иконите в програмата. След това свързващият редактор обработва всички двоични файлове, създадени по време на компилацията и добавя необходимите

библиотечни файлове. Като свърже всички тези ресурси в една обща програма, свързващият редактор създава изпълним файл или модул за динамичната библиотека.

2. База данни в Delphi

Програмирането на бази данни е свързано с използването на някои специфични инструменти и подходи. Най-общо, базите данни представляват съвкупност от един или няколко файла, които съдържат записно-ориентирани съвкупности от данни и които най-често имат някакви логически връзки. Под логически връзки трябва да се разбира възможност за свързване на различни физически таблици според обща типова информация. Много често в тези случаи се използва термина релационни бази от данни. В термините на Delphi, под база от данни се разбира мястото, където се съхраняват данните във формата на физически файлове. Като име на базата от данни може да се използва както името на директорията в която се намират файловете, така и псевдоним (alias). Псевдонимът е кратко наименование на базата от данни, който съдържа и пътят до физическото разположение на базата данни (директория и поддиректории). По този начин, приложенията стават независими от конкретното разположение на файловете от базата данни.

Delphi предоставя много богати възможности за работа с бази от данни. Тук може да се разработи както много проста, така и доста сложна система за управление на бази от данни без да се напише и един ред програмен код. При създаването на приложения за работа с бази от данни много важно е създаването на таблиците които съдържат данните. Това може да стане предварително (извън програмната среда на Delphi) или по време на изпълнение на приложението (програмно създаване на базата данни). Правилното проектиране на таблиците с отчитане на връзките между отделните елементи на данните е въпрос който изисква дълбоки познания в областта на релационните бази от данни. Колкото до обработката и управлението на базите от данни, Delphi е много добра система, при която необходимостта от написване на програмен код е сведен до възможният минимум.

За създаването и работата с бази от данни в Delphi се използват обектно-ориентирани компоненти. За да се създаде приложение, работещо с бази от данни, трябва във формата да се поместят необходимите компоненти и да се установят техните свойства. Разработено в Delphi приложение може да създава и получава достъп до тях самостоятелно или с помощта на други допълнителни програми за работа с тях.

Приложенията, използващи бази от данни, работят подобно на всички други приложения, създадени с Delphi. Разликата е, че те използват някои допълнителни управляващи елементи, компоненти и класове. В Delphi има специална категория компоненти за работа с бази от данни, наричани се компоненти за управление на данни (data-control components). Те са достъпни от страниците Data controls и Data Access в палитрата с компоненти. Наричат се така, защото създават управляващи елементи, имащи специални възможности за работа с данни.

Към тази група компоненти трябва да се отнесат компонентите за достъп до данните – TTable, TQuery; достъп до информацията в таблиците – TDataSource; компоненти за управление на данни – TDBEdit, TDBGrid, TDBNavigator и други. Част от тези компоненти са визуални - TDBEdit, TDBGrid, TDBNavigator, а друга част невизуални - TTable, TQuery, TDataSource.

За създаване, поддръжка и управление на бази от данни, Borland разполага с няколко полезни инструменти:

- **DataBase Desktop (DBD)** – това е програма, явяваща се съкратена версия на Paradox for Windows – една от най-използваните системи за работа с бази от данни. Тя не може да отпечата или да генерира справки, но може да създава, чете, модифицира и реструктурира таблици с бази от данни. Тя може да се разглежда, като независим редактор на данните. DBD се стартира самостоятелно и работи с базата данни независимо от главното приложение на Delphi, като с него може да се задават и псевдоними на базите от данни.

- Машина за бази от данни на Delphi (**Borland Database Engine – BDE**) – това е 32-битов Windows програмен продукт и е част от обкръжението на програмната среда на Borland за управление на бази от данни. За създаването на приложения, работещи с бази от данни не е необходимо да се познава BDE, нито пък това е нужно на потребителите на приложенията. BDE се инсталира автоматично заедно с пакета на Delphi и след това работи без да напомня за съществуването си. BDE съдържа редица сервизни инструменти с необходимите драйвери за работа с различни бази от данни (dBase, Paradox, FoxPro, Access и text databases). Посредством BDE, могат да се осъществява връзка и достъп до SQL сървъри за обработка на бази от данни, включително Informix, DB2, InterBase, Oracle и Sybase.

3. Програмен език Object Pascal

Преди създаването на Delphi, Borland вече бяха въвели някои подобрения в езика Pascal и всъщност бяха разработили Object Pascal. Съотнесен към стандартният език Pascal, Object Pascal е това което е C++ спрямо стандартният език C. Object Pascal добави класове към стандартният език и по този начин го пренесе в света на обектно-ориентираните езици за програмиране. Добавени бяха някои допълнителни ключови думи за работа с компонентния модел разработван от Borland.

Основата на обектните възможности на Object Pascal се съдържа в дефинирането на класове. Класовете са използват за разработване на компонентите от Visual Component Library (VCL). Те представляват съвкупност от полета и методи, работещи заедно за изпълнение на специфична програмна задача. По този начин, класовете капсулират (затварят) обработката и управлението на данните в едно собствено пространство (пространство на данни и методи за обработка) и освобождават потребителите от грижата за досадни подробности.

Обектно-ориентираното програмиране е разширение на структурното програмиране, доста добре застъпено в основните версии на Pascal. То изисква добра практика на програмиране и води до създаването на програми с ясен програмен код и лесно приложение на обекти и елементи в други програми.

В основата на обектно-ориентираното програмиране стои понятието “обект”. Обектът е програмен елемент, който затваря всичките си данни и код в един пакет. Много близко до понятието обект в Pascal е записът, който представлява съвкупност от полета съдържащи данни от различен тип. Записът позволява лесно обръщение към свързаните елементи в данните. Обектът е също съвкупност от данни и има полета с различен тип данни. За разлика от записа, обектът съдържа и програмен код (процедури и функции) който работи с данните съдържащи се в обекта. Тези процедури и функции се наричат методи.

Също за разлика от записите, обектите могат да съдържат и свойства. Към свойствата на обектите в Delphi се отнасят част от данните в записа, които имат начални стойности. Тези начални стойности могат да се променят по време на проектиране на приложението или по време на работа на програмата.

Когато се добавя нова форма към даден проект в Delphi, в редактора на код автоматично се декларира обектен тип (клас) – форма и програмен код, които създават този обект:

```
type
  TForm1 = class (Tform)
  private
    { Private declarations }
  public
    {Public declarations}
end ;
var Form1 : TForm1;
```

Новият обектен тип е TForm1 (клас) и произлиза от типа TForm, който също е обект и е един от основните обекти в Delphi. Конкретната форма (Form1) се декларира като обект, инстанция (екземпляр) на типа TForm1:

```
var Form1 : TForm1 ;
```

В горната декларация се описва една празна форма. Когато към формата се добавят компоненти, те се добавят към декларацията на класа и същевременно се отразяват в конкретната променлива Form1.