



# Принципы проектирования проектов в Delphi

*Не волнуйся, если что-то работает не так. Если бы все работало как надо, ты сидел бы без работы. Закон Мерфи*

## 6.1. Основные элементы интерфейса пакета

Создание прикладных программ (ПП), или приложений, delphi выполняет в интегрированной среде разработки IDE (integrated development environment). IDE служит для организации взаимодействия с программой и включает ряд окон, содержащих различные управляющие элементы. С помощью средств интегрированной среды разработчик может проектировать интерфейсную часть приложения, а также писать программный код и связывать его с управляющими элементами.

Запуск пакета выполняется из главного меню Windows. Создание проекта выполняется выбором из меню **File** команды **New** режима **VCL Forms Application** (рис. 6.1).

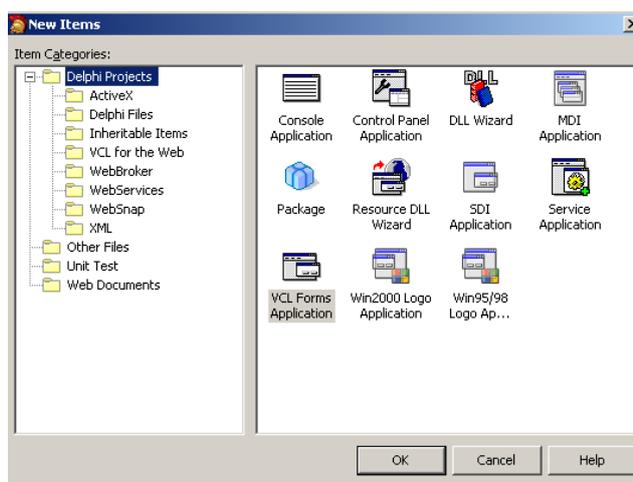


Рис. 6.1. Режимы создания проектов в Delphi.

После загрузки экран имеет вид, показанный на рисунке 6.2.

**Главное окно delphi.** Имя окна соответствует имени проекта, по умолчанию – **project1**. Главное окно содержит меню (1), панель инструментов меню (2) и окна проекта.

**Главное меню (1)** содержит все команды, необходимые для работы в среде визуального программирования Delphi.

**Панель компонентов (2)** – группа кнопок, которые дублируют наиболее часто используемые команды меню.

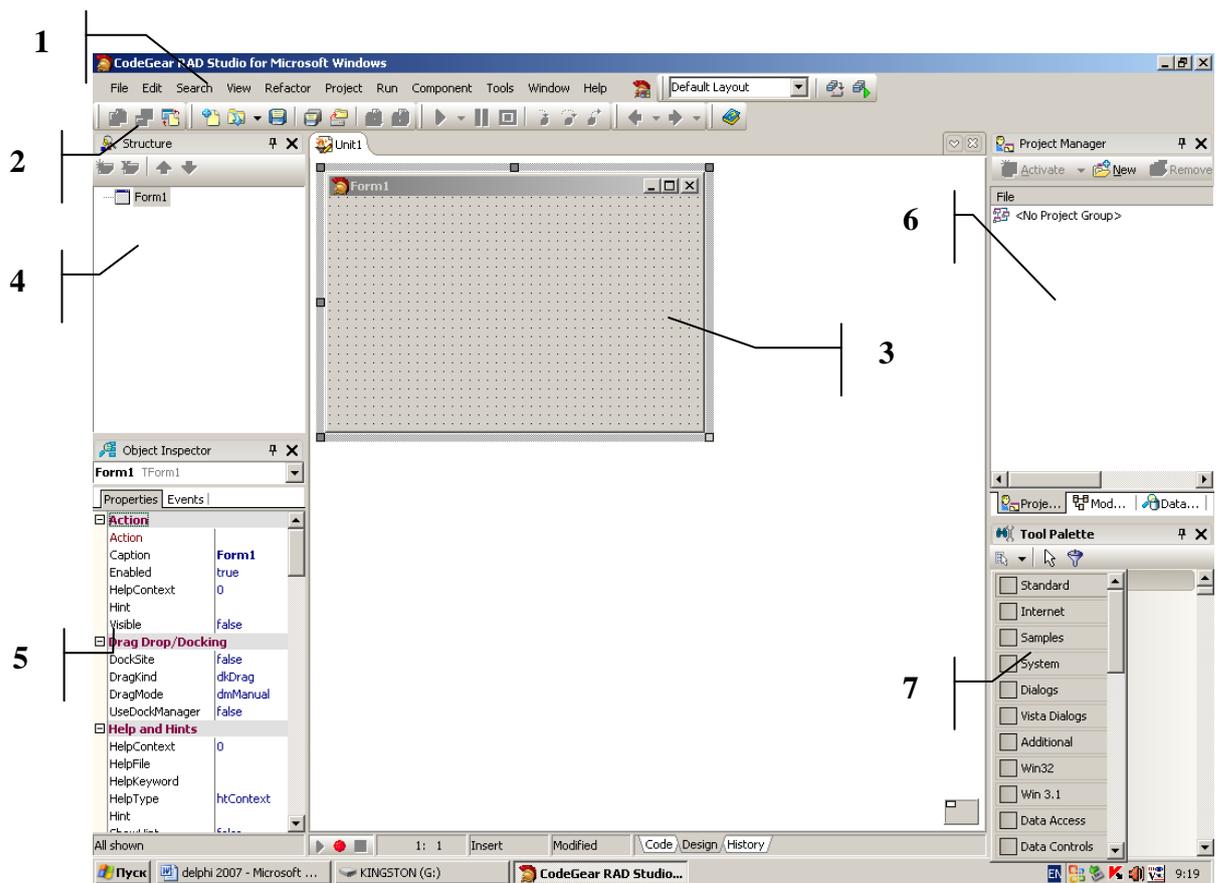


Рис.6.2. Вид экрана Delphi после загрузки.

**Окно конструктора формы (form1) (3).** Представляет собой прообраз будущего приложения. На форме располагаются все необходимые объекты интерфейса, определяемые программистом.

**Окно редактора кода (unit1.pas)** находится под окном конструктора формы (рис. 6.3.). Переключение в это окно выполняется или клавишей F12, или выбором вкладки code в нижней части окна конструктора форм.

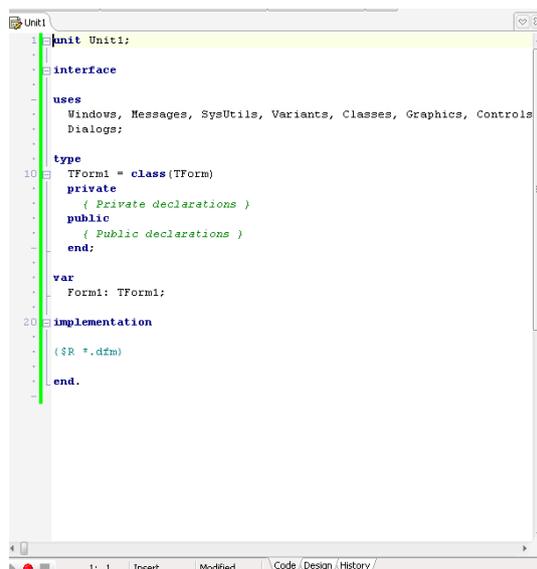


Рис. 6.3. Окно редактора кода.

**Окно структуры проекта** (4) заполняется информацией после активизации окна редактора. Данное окно служит для отображения разделов программы. Двойной щелчок по любому из разделов активизирует его в окне редактора.

**Окно инспектора объектов** (object inspector) (5). Основное назначение этого окна определять свойства объектов и события, которые будут с ними связаны. Окно содержит информацию о выделенном объекте. Оно имеет 2 вкладки. Свойства объекта определяются на вкладке **Properties** и представляют собой таблицу из 2-х столбцов. В первом столбце указывается название свойства, во втором – его значение. Если около названия свойства стоит знак «+», то в него входит набор свойств. Для их открытия стоит щелкнуть мышью на знак «+». Методы или события, связанные с объектом определяются на вкладке **Events**. Эта вкладка также состоит из двух столбцов: в первом указывается название события; во втором – название соответствующей процедуры. Вывод окна на экран выполняется клавишей F11 или из меню View выбирается команда Object inspector;

**Окно менеджера проекта** (6) содержит общую информацию о проекте в целом и обо всех файлах проекта в отдельности. Элементы списка можно просматривать и применить к ним команды контекстного меню.

**Палитра компонент (7)** состоит из набора групп, в которых находятся компоненты. **Компонент** – это некий функциональный элемент интерфейса, обладающий определенными свойствами. Размещая компоненты на форме, программист определяет ее интерфейс. Компоненты объединяются в группы по функциональному признаку.

Остановимся на некоторых командах меню Edit по редактированию (рис. 6.4).

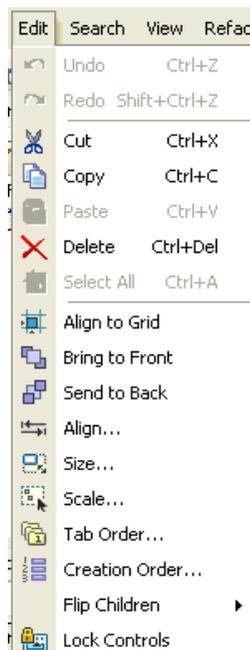


Рис. 6.4. Команды меню редактирования.

**Align to Grid** – выровнять по границе;

**Bring to Front** – разместить на переднем плане;

**Send to Back** – разместить на заднем плане;

**Align...** - открывает диалоговое окно, с помощью которого можно произвести выравнивание компонента (рис. 6.5.);

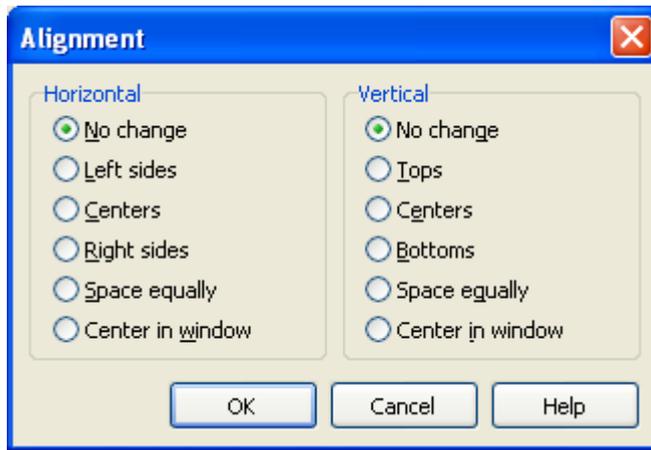


Рис. 6.5. Окно выравнивания компонента на форме.

Таблица 6.1.

Режимы выравнивания объектов на форме.

| По горизонтали           | По вертикали             |
|--------------------------|--------------------------|
| Не изменять              | Не изменять              |
| По левой стороне         | По верхнему краю         |
| По центру                | По центру                |
| По правой стороне        | По нижнему краю          |
| На одинаковом расстоянии | На одинаковом расстоянии |
| По центру окна           | По центру окна           |

**Size...** - открывает диалоговое окна с помощью которого можно изменять размер компонента (рис. 6.6.);

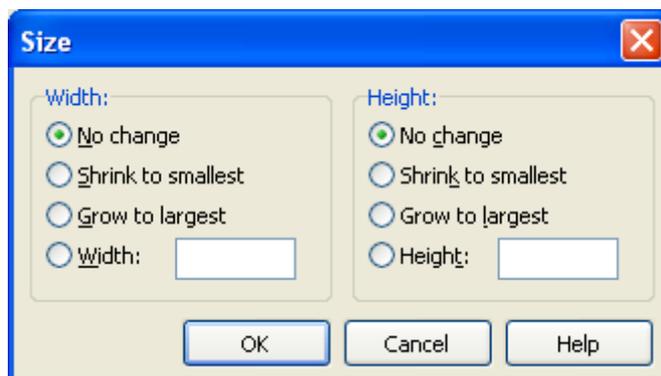


Рис. 6.6. Окно изменения размера компонентов.

Режимы изменения размеров компонентов.

| Ширина                      | Высота                      |
|-----------------------------|-----------------------------|
| Не изменять                 | Не изменять                 |
| Сократить до наименьшего    | По верхнему краю            |
| Увеличить до наибольшего    | По центру                   |
| Установить размер по ширине | Установить размер по высоте |

**Scale...** - осуществляет изменение масштаба формы и ее компонентов (рис. 6.7.);

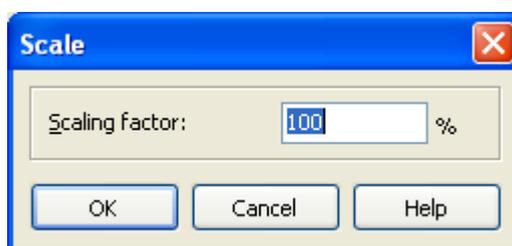


Рис. 6.7. Окно изменения масштабов формы.

**Tab Order...** - выводит список компонентов, размещенных на форме, позволяет менять порядок их обхода с использованием клавиши **Tab**;

**Creation Order...** - выводит список невидимых компонентов, размещенных на форме, позволяет менять порядок их обхода с использованием клавиши **Tab**;

**Flip Children** – выравнивает дочерние объекты в соответствии с родительскими;

**Lock Control** – контролирует редактирование компонента, размещенного на форме.

## 6.2. Характеристика проекта

Проект Delphi состоит из форм, модулей, установок параметров проекта, ресурсов и т.д. вся эта информация размещается в файлах. Многие из этих файлов создаются автоматически пакетом, когда вы строите свое

приложение. Ресурсы, например пиктограммы, находятся в файлах, которые вы получаете из других источников или создаете при помощи многочисленных инструментов и редакторов ресурсов. Кроме того, компилятор также создает файлы.

При проектировании приложения Delphi создает следующие файлы:

Таблица 6.3.

Файлы проекта.

| Расширение файла                          | Назначение файла   |
|---|--|
| Файл проекта (.dpr)                       | Этот текстовый файл используется для хранения информации о формах и модулях. В нем содержатся операторы инициализации и запуска программ на выполнение.  |
| Файл модуля (.pas)                        | Каждой создаваемой форме соответствует текстовый файл модуля, используемый для хранения кода программы. Можно создавать модули, не связанные с формами, например для библиотечных процедур и функций.  |
| Файл формы (.dfm)                         | Это двоичный или текстовый файл, который создает Delphi для хранения информации о содержании формы. Каждому файлу формы соответствует файл модуля (.pas).  |
| Файл параметров проекта (.dfo или .opt)   | В этом файле хранятся установки параметров проекта.  |
| Файл ресурсов (.res)                      | Этот бинарный файл содержит используемую проектом пиктограмму и прочие ресурсы   |
| Файлы резервных копий (.~dpr, .~df, .~pa) | Это файлы резервных копий для файлов проекта, формы и модуля соответственно. Если что-то безнадежно испорчено в проекте, можно соответственно изменить расширения этих файлов и таким образом вернуться к предыдущему не испорченному варианту.                    |
| Файл конфигурации окон (.dsk)             | Файл хранит конфигурацию всех окон среды разработки.   |
| Исполняемый файл (.exe)                   | Это исполняемый файл приложения. Он является автономным исполняемым файлом, для которого больше ничего не требуется, если только не используются библиотеки, не входящие в стандартную настройку Delphi, или не используется поддержка пакетов времени выполнения. |

|                              |   |
|------------------------------|---|
| Объектный файл модуля (.dcu) | Это откомпилированный файл модуля (.pas), который компонуется в окончательный исполняемый файл. |
|------------------------------|---|

Взаимосвязь между отдельными частями (файлами) проекта представлена на рисунке 6.8.

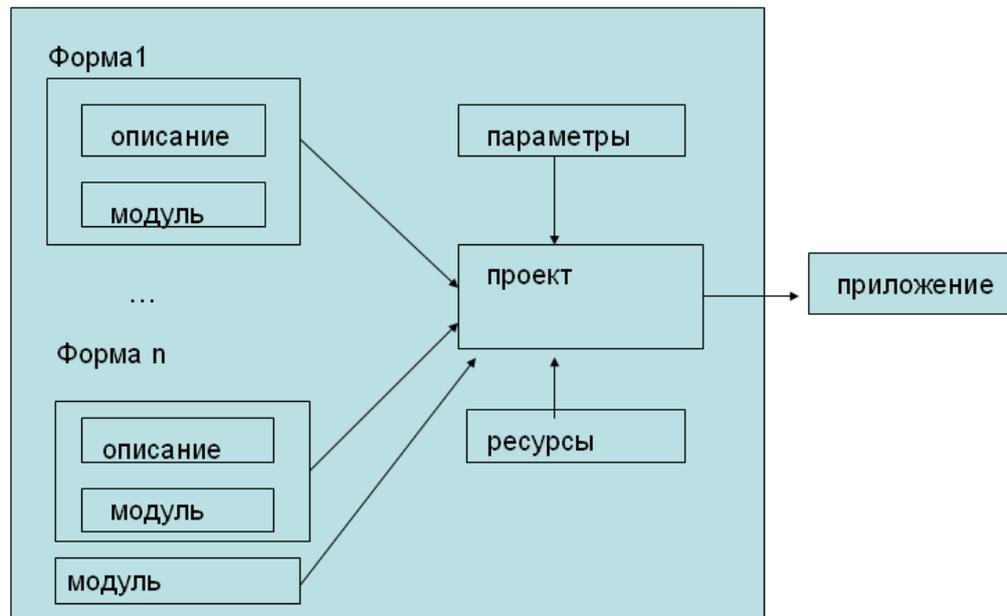


Рис. 6.8. Взаимосвязь элементов проекта.

Главной частью приложения является файл проекта, содержащий код на языке Object Pascal, с которого начинается выполнение программы и который обеспечивает инициализацию всех модулей. Он создается и модифицируется Delphi автоматически в процессе разработки приложения. Имя, которое дается файлу проекта при сохранении, становится именем исполняемого файла.

При сохранении все файлы проекта должны записываться в один каталог.

Файл проекта создается пакетом автоматически и имеет следующий вид:

***Program project1;***

***Uses***

***Form1,*** //имя формы

***Unit1 in @unit1.pas@ {form1};*** //имя программного модуля

***{SR\*.res}*** //ресурсы \* - имя как у проекта, например пиктограммы.

***begin***

```
application.initialize; //установка начальных параметров проекта  
application.createform(Tform1,form1); //создание формы  
application.run; //выполнение программы
```

*end.*

Просмотр этого файла **project – view source**. Просмотр форм **view – forms** или **shift +F12**.

### 6.3. Структура программного модуля стандартного проекта Delphi

Рассмотрим текст модуля с пустой формой. Все выделенные элементы модуля создаются в автоматическом режиме.

```
unit Unit1;  
interface //открытый интерфейс модуля  
uses // список подключаемых модулей  
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,  
Forms, Dialogs;  
//объявление класса формы  
type  
TForm1 = class(TForm)  
Private //закрытый раздел класса  
{ Private declarations }  
{ помещаются объявления переменных, функций и процедур,  
включаемых в класс формы, но не доступных для других модулей }  
public //открытый раздел класса  
{ Public declarations }  
{ помещаются объявления переменных, функций и процедур,  
включаемых в класс формы и доступных для других модулей }  
end;  
var  
Form1: TForm1;
```

{помещаются объявления типов, констант, переменных, функций и процедур, к которым будет доступ из других модулей, но которые не включаются в класс формы}

### ***implementation***

{помещаются объявления типов, констант, переменных, к которым не будет доступа из других модулей. Тут же должны быть реализации всех объявленных в разделе `interface` функций и процедур, а также могут быть реализации любых дополнительных, не объявленных ранее функций и процедур}

***{R \*.dfm}***

***end.***

## **6.4. Этапы разработки проекта**

Delphi относится к системам визуального программирования, которые называется также системами **RAD** (rapid application development – быстрая разработка приложений). Разработка приложений в этих средах состоит из двух этапов:

- **создание интерфейса приложения.** Определяется способ взаимодействия пользователя и приложения, т.е. внешний вид формы. Интерфейс приложения составляют компоненты, которые выбираются из палитры компонент и размещаются на форме. Компоненты делятся на:

1. **визуальные** – кнопки, списки, переключатели и т.д. Эти компоненты могут быть как управляющими, так и информационными. Информационные компоненты – это компоненты, которые служат для вывода или ввода данных, но при выборе их не выполняются никакие действия. Управляющие компоненты – это компоненты, при выборе которых выполняются процедуры обработки. Программист сам определяет какие компоненты какую роль будут играть, и какие процедуры обработки с ними связывать.

2. **вспомогательные** (невизуальные) – таймер (Timer), или таблица данных (table).

При создании интерфейса приложения для любого компонента выполняются следующие операции:

- выбор компонента в палитре компонент и размещение его на форме;
- изменение свойств компонента.

- определение функциональности приложения. **Функциональность приложения** определяется процедурами, которые выполняются при возникновении определенных событий, например, происходящих при действиях пользователя с управляющими элементами формы. Для этого на форме делают активным управляющий элемент. Для него в инспекторе объектов выбирают вкладку «Свойства» и двойным щелчком выбирают нужное свойство. В результате открывается окно редактора исходного текста с автоматически созданным заголовком процедуры обработки события.

Простейшее приложение представляет собой заготовку, обеспечивающую все необходимое для любого приложения.



## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Delphi - характеристика пакета.
2. Вид экрана Delphi после запуска. Характеристика элементов интерфейса.
3. Характеристика и состав проекта.
4. Этапы разработки проекта.