

## Практическая работа № 1, Элементы интерфейса

### Общие положения

Delphi – интегрированная среда разработки (**IDE**– Integrated Development Environment), используемая для создания и поддержки приложений, которые могут работать под любой 32 разрядной операционной системой типа Windows.

Delphi дает возможность создавать программы в стиле визуального конструирования, т.е. пользователь оформляет свою будущую программу, и видит результаты своей работы еще до запуска самой программы.

Это довольно легкая в изучении среда, и в то же время довольно сложная.

Delphi имеет пользовательский графический интерфейс, подобный Visual Basic и C++. На данный момент множество фирм приняло за стандарт данный интерфейс для собственных приложений.

Весь исходный текст программ на Delphi пишется на языке Object Pascal, практически ничем не отличающемся от принципов, заложенных в Turbo Pascal. Процесс написания приложения разделяется на две части:

**Первая часть** – конструирование. Программист располагает на рабочей форме своей программы необходимые элементы объектов Delphi, называемых *компонентами*. Он позиционирует эти элементы, устанавливает нужные размеры, меняет свойства.

**Вторая часть** – написание программного кода. Программист описывает свойства элементов, доступных только во время работы приложения, реакцию на событие появление окна, нажатия на кнопку и т.п.

### Элементы интерфейса

После запуска Delphi откроется главное окно среды разработки. Оно может отличаться в зависимости от версии Delphi, но основа остается не изменной. Мы будем рассматривать Delphi 2006.

Главное окно (рис.1) выполнено в классическом стиле Windows программ, и содержит множество панелей. С помощью разворачивающегося списка, находящегося в верхней строке меню, можно выбрать созданный вами или определяемый средой разработки вид окна. Возможны следующие варианты:

**Default Layout** – размещение по умолчанию.

**Classic Undocked** – классическое размещение; панели находятся не внутри главного окна, а выделены в самостоятельные окна.

**Debug Layout** – размещение отладки; включается автоматически, если вы запустили программу, и выключается автоматически, после выполнения программы.

Если вы изменили расположение окон и хотите, чтобы это изменение осталось при следующем запуске среды разработки, то его нужно сохранить, щелкнув по кнопке **Save current desktop (Сохранить текущий рабочий стол)**, которая находится справа от разворачивающегося списка возможных размещений.

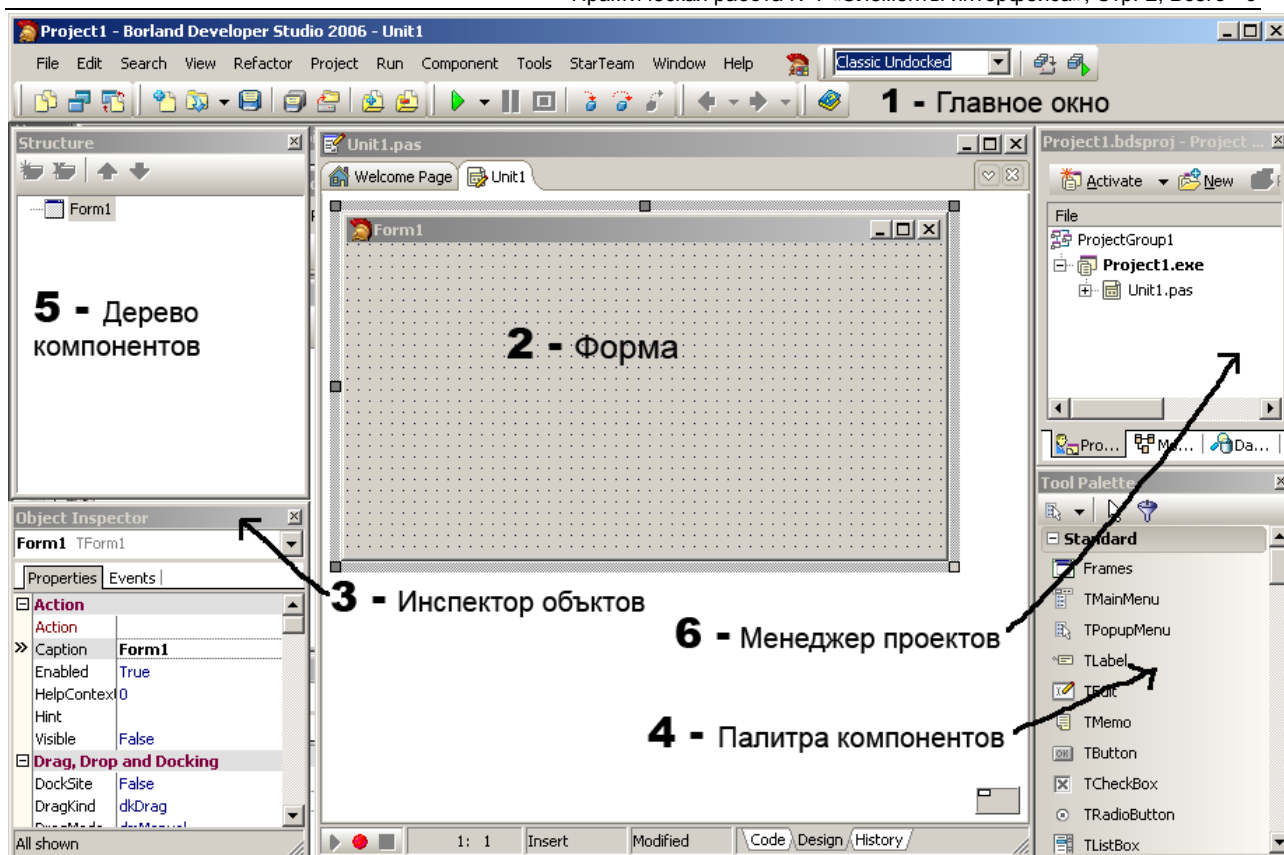


Рис.1

На главном окне представлены следующие панели:

1. Главное окно программы.
2. Форма.
3. Инспектор объектов.
4. Палитра компонентов.
5. Дерево компонентов.
6. Менеджер проектов.

### Примечание

При работе никогда не закрывайте эти окна! Их нужно просто «сворачивать».

### Главное окно

**Главное окно** осуществляет основные функции управления проектом создаваемой программы. В нем находится основное меню и панели инструментов.

### Примечание

Главное окно остается открытым все время работы **IDE**. Закрывая его, вы тем самым закрываете Delphi и все открытые в нем окна.

**Основное меню** содержит команды, необходимые для разработки и тестирования приложений и используется так же, как любое стандартное Windows-меню. Основное меню содержит несколько вложенных меню, в которых расположены различные команды.

**Панель инструментов** представляет собой набор кнопок, который дублирует различные команды основного меню. Эти кнопки служат для ускорения доступа к наиболее часто выполняемым командам. На этой панели есть, в частности, кнопка сохранения проекта на диске, кнопка открытия проекта, кнопка запуска программы на выполнение.

Пользователь может изменить конфигурацию панели инструментов. Для этой цели используется окно пользовательских настроек, с помощью которого можно отображать, скрывать или менять положение на панели инструментов тех или иных инструментов.

Окно пользовательских настроек можно открыть, щелкнув правой кнопкой мыши на панели инструментов. С помощью появившегося контекстного меню выполняются необходимые настройки.

## Форма

**Окно проектировщика формы** – главное место, где происходит сборка программы из компонентов, содержащихся в палитре компонентов. Компоненты можно перемещать по форме, зажимая левую кнопку мыши. Также можно изменять размер, как самой формы, так и компонентов, перемещая мышью специальные маркеры размера.

Сама форма – это уже готовая к исполнению программа. По ходу работы система формирует в окне **Code** (Редактора кодов) текст программы на языке Object Pascal, связанной с формой.

С помощью вкладок вы можете переключаться между файлами исходного кода (**Code**), окном редактирования формы (**Design**) и окном истории изменений (**History**).

## Окно редактора кодов

**Code Editor** (Редактор кода) представляет собой текстовый редактор с подсветкой синтаксиса языка программирования – различные элементы языка окрашиваются различными цветами для удобства чтения кода.

## Палитра компонентов

**Tool Palette** (Палитра компонентов) представляет собой множественные тематические страницы, на которых располагаются компоненты. Все компоненты разделены на категории, которые можно для удобства сворачивать и разворачивать.

Работа с палитрой компонентов очень проста. В режиме редактирования формы можно дважды нажать левой кнопкой мыши на интересующем компоненте, и он будет добавлен на форму, или перетащить объект на форму, зажав левую кнопку мыши.

Как и прочие элементы среды разработки, окно палитры компонентов может быть настроено под требования разработчика. Для этого необходимо установить курсор на окно палитры компонентов и нажать правую кнопку мыши – появится всплывающее меню, через которое можно редактировать категории, настраивать цвета и прочее.

### Примечание

Пока не открыт или не создан новый проект, в списке отображаются файлы и приложения, которые разработчик может создать.

## Инспектор объектов

**Object Inspector** (Инспектор объектов) - отображает свойства активного компонента или самой формы. Имя активного компонента находится под заголовком панели.

Эта панель имеет две вкладки – **Properties** (Свойства) и **Events** (События).

На первой вкладке (**Properties**) постоянно отображаются все доступные свойства выбранного компонента. В левой колонке содержится список, а в правой – текущие значения по умолчанию.

На второй закладке (**Events**) отображаются возможные обработчики событий для выбранного компонента. В левой колонке – названия, а в правой – соответствующие свойства или процедуры.

### Что же такое свойства и реакции на события?

Каждый объект в Delphi, даже само окно разрабатываемого приложения, имеют определенные свойства: цвет, размер, отображаемый текст и т.п. Эти свойства можно менять еще до запуска проектируемой программы на выполнение. В зависимости от изменяемого свойства результат можно просматривать уже сразу.

Реакции на события, это результат произошедшего системного события, например, щелчок мыши, нажатие на кнопку, открытие окна и т.п. Реакцию на событие назначают программно, указывая список действий, которые необходимо произвести. Например, если пользователь щелкает по кнопке, появляется графическое изображение.

### Дерево компонентов

На панели **Structure** (Дерево компонентов) представлен полный перечень всех компонентов вашего проекта. Здесь легко найти компонент даже, если он перекрывается другим компонентом.

### Менеджер проектов

Файлы, образующие приложение, формы и модули – собраны в проект. **Project Manager** (Менеджер проектов) показывает списки файлов и модулей приложения и позволяет осуществлять навигацию между ними.

Структура проекта выполнена в виде иерархического дерева. Корнем дерева проектов является **Project Group** (Группа проектов). Она объединяет в себе один или более проектов. Каждый проект, в свою очередь, содержит файлы кода, заголовочные файлы, файлы ресурсов и прочее.

### Создание и сохранение проекта и файла модуля

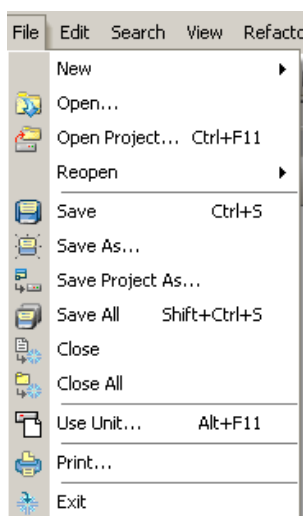


Рис.2

Каждый пункт **Меню** содержит набор команд. На рис.2 изображено открытое меню пункта **File** (Файл). Рассмотрим некоторые команды.

Команда **File/New/VCL Forms Application** создает новый проект. После выполнения этой команды в менеджере проектов (см. рис.3) можно увидеть все составляющие проекта.

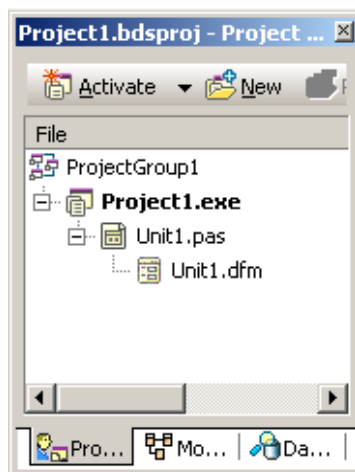


Рис.3

ProjectGroup1 (Заголовок дерева) – имя группы проектов.

Project1.exe – имя проекта (приложения). Когда создается новое приложение, Delphi дает ему имя Project, прибавляя порядковый номер.

Unit1.pas – модуль. Проект состоит из отдельных модулей. Каждое окно программы хранится в отдельном модуле. Файлы с расширением pas содержат исходный код модуля.

Unit1.dfm – визуальная форма. Она сохраняется в файле с таким же именем, как и у модуля, но с расширением dfm.

Если у вас в проекте несколько приложений, то только одно из них является активным, и только с ним вы можете работать. Имя активного приложения выделено жирным шрифтом. Чтобы изменить активное приложение, достаточно дважды щелкнуть по его имени левой кнопкой мыши.

После создания нового проекта рекомендуется сохранить его, для этого выполним команду **File/Save all** (Сохранить все). Если файл не был ранее сохранен, Delphi открывает диалоговое окно **Save As**, в котором нужно указать имя файла. По умолчанию указано текущее имя – Unit1.pas. Это имя можно изменить. После Delphi запросит имя проекта. По умолчанию указано текущее имя – Project1.dpr. Имя проекта задает имя будущего исполняемого файла (с расширением exe).

### Примечание

1. Сохранить проект можно так же с помощью команды **File/Save Project As..**
2. Для каждого нового проекта лучше отводить отдельную папку, т.к. проекты включают множество файлов. Это позволит вам упростить процесс копирования проекта на другой компьютер.
3. Названия файлов должны состоять из букв английского алфавита, символа подчеркивания и арабских цифр, первым символом не может быть цифра. Пробелы – недопустимы.

### Перечень сохраняемых файлов на диске

**DPR (Delphi Project)** – файл проекта содержит основной код программы, ссылки на все окна (формы) проекта и относящиеся к ним модули. В нем также содержится код инициализации. Имеет одноименное название с проектом.

**PAS (Pascal)** – файл модуля программы содержит текст, который вы видите в окне редактора кода, так называемого модуля программы.

**DFM (Delphi Form)** – файл формы представляет собой файл с полными данными о проектировщике формы: позиция, размер, расположенные компоненты и т.п. Форма приложения является неотъемлемой частью модуля PAS и имеет то же название.

**RES (Resource)** – ресурсный файл содержит в себе иконки, значки указателя мыши, картинки, звуки.

**DOF (Delphi Options File)** – содержит (в текстовом виде) текущие опции проекта

**DSK (DeSKtop)** – файл с настройками рабочей среды Delphi.

**CFG** – содержит настройки конфигурации проекта.

**DCU (Delphi Compiled Unit)** – двоичный файл модуля имеет одноименное название с модулем и представляет собой откомпилированный модуль проекта в промежуточном формате. Когда компилируется программа, все модули компилируются в файлы формата dcu, а потом собираются в один и получается исполняемый файл.

**EXE** – откомпилированная программа сохраняется автоматически при запуске проекта на выполнение. Обновляется в момент компиляции. Имеет одноименное название проекта. Полностью самостоятельное приложение.

По ходу работы в среде Delphi могут автоматически создаваться файлы, имеющие расширение, начинающееся символом ~. Это резервные копии файлов, которые создаются при их повторном сохранении.

### Задание

1. Создайте новый проект. Для этого выполните команду **File/New/VCL Forms Application**.
2. Сохраните код программы и проект под именами, например, **Unit1.pas** и **Pr1.dpr**. Для этого выполните команду **File/Save all**, предварительно создав папку, в которую вы будете записывать свои проекты.
3. Откомпилируйте и выполните «пустую» программу, созданную в п.1 – на клавиатуре нажмите кнопку **F9**. В результате компиляции создается exe файл, который можно будет запускать на выполнение.

#### Примечание

Компиляцию можно выполнить еще двумя способами:

- а) выбрать команду основного меню **Run/Run**,
- б) щелкнуть «зеленый треугольник» на панели инструментов.

На экране исчезнут все вспомогательные окна, у формы пропадет координатная сетка. Форма примет вид стандартного Windows-окна с заголовком Form1, со стандартными кнопками: свернуть, на весь экран, закрыть.

4. Закройте запущенную программу, щелкнув мышью на стандартной кнопке «X». Снова на экране появляется инспектор объектов и редактор формы.

#### Примечание

Особое внимание обратите на п.2.; самая распространенная ошибка – выполнение команды **File/Save** вместо **File/Save all**.

5. Закройте Delphi.
6. Загрузите Delphi.
7. Загрузите свой проект, выполнив команду **File/Open Project** и указав правильно имя (п. 2).
8. Повторите п. 3 и 4.