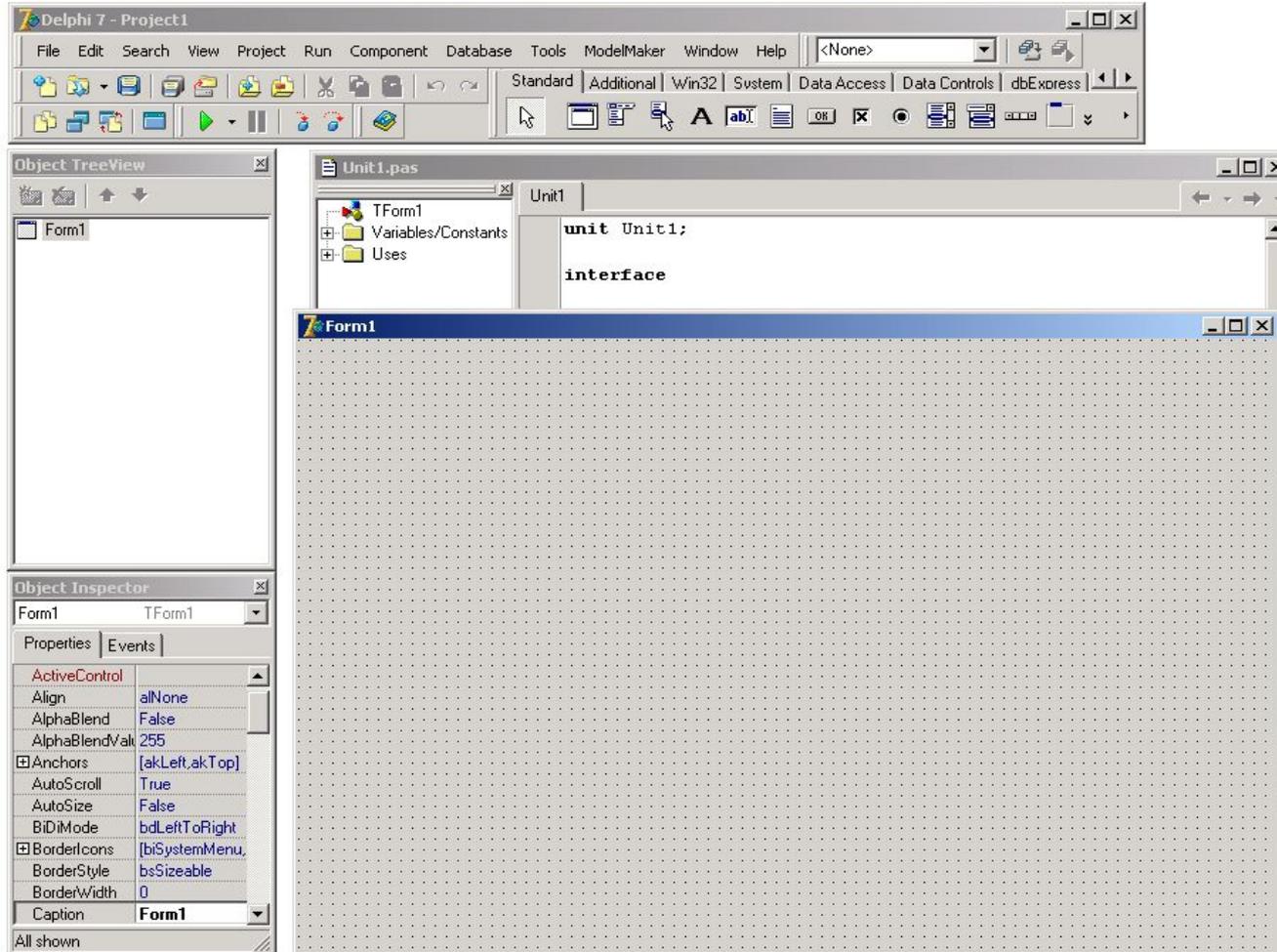

Создание приложений в Delphi

Особенности программирования в оконных операционных средах

Окна среды программирования используются для выполнения действий, связанных с формированием в окнах создаваемого приложения видимых элементов (кнопок, списков, надписей, меню и т.д.), и для создания текстов программных процедур обработки событий, связанных с этими элементами (таких, например, событий, как нажатие на кнопку, выбор строки из списка или команды из меню и т. д.).

Среда визуального программирования

на примере среды программирования Delphi 7



Система окон, команд, меню и панелей среды разработки

Окна среды разработки, например, окна **среды программирования Delphi 7** (главное окно **Delphi**, окна **Object TreeView**, **Object Inspector**, а также окно формы и окно программного модуля – окна с заголовками типа **Form1** и **Unit1.pas**) используются для создания проекта программного приложения типа **Application**, а именно, для формирования видимых в окнах программ **элементов** (кнопок, списков, надписей, меню и т.д.) и для создания **процедур обработки событий**, связанных с этими элементами (таких, например, событий, как нажатие на кнопку, выбор строки из списка или команды из меню и т.д.).

Главное окно **среды программирования Delphi** содержит главное меню **Delphi** (команды File, Edit, View, Run, Help и другие), **панели инструментов** с кнопками и **Палитру компонентов** (обычно справа вверху).

Команды главного меню **Delphi** открывают доступ к меню с более конкретными командами второго уровня.

Быстрый доступ к наиболее важным и употребительным командам можно получать также с использованием **контекстных меню** (например, после щелчка правой кнопкой "мыши").

Панели инструментов содержат кнопки, которые могут быть использованы для более быстрого выполнения тех же действий по сравнению с использованием команд главного меню **Delphi** или **контекстных меню**.

Палитра компонентов обычно занимает правую часть главного окна **Delphi**. С помощью компонентов создаются видимые в окнах программ элементы (кнопки, списки, надписи, меню и т.д.).

Прочие окна среды программирования (**Object TreeView**, **Object Inspector**, а также окно формы и окно программного модуля – это окна с заголовками типа **Form1** и **Unit1.pas**) используются для создания проекта программного приложения.

Использование компонентов

Компоненты (кнопки, текстовые поля, надписи, списки и прочие) при создании приложения должны быть размещены в окне формы (или в окнах форм) этого приложения.

При выполнении действий, связанных с выбором, размещением и использованием **компонентов**, используются **Палитра компонентов** (с нее выбираются компоненты с помощью указателя "мыши"), а также окно **Object Inspector** (предназначенное для отображения и установки **значений свойств компонентов** и для **определения реакции компонентов на все доступные события**) и окно **Object TreeView** (в этом окне отображаются связи между **компонентами**).



Свойства, события, методы

Значения **свойств** компонентов определяют расположение, размеры, вид и прочие характеристики компонентов.

Значения **свойств** компонентов можно устанавливать в окне **Object Inspector** на вкладке **Properties** на стадии проектирования приложения.

Значения **свойств** компонентов можно изменять и программным путем во время выполнения приложения

Событие наступает при выполнении каких-либо действий, связанных с компонентом.

Если при наступлении события должны быть выполнены действия, являющиеся реакцией приложения на это **событие**, то для выполнения этих действий должна быть создана **процедура обработки этого события**. Затем в эту процедуру должны быть добавлены операторы, выполняющие действия, определяющие реакцию программного приложения на **событие**.

В общем случае для создания **процедуры обработки того или иного события**, связанного с выделенным компонентом или даже с несколькими выделенными компонентами, можно в окне **Object Inspector** открыть вкладку **Events (События)**, поместить указатель "мыши" в правую часть строки с названием выбранного события и выполнить двойной щелчок левой кнопкой "мыши".

После выполнения этих действий в окне **Object Inspector** в строке с названием выбранного события появится название **процедуры обработки этого события**, а в окне программного модуля появится "пустая" заготовка **процедуры обработки этого события**.

В созданную заготовку **процедуры обработки события** должны быть добавлены операторы, выполняющие действия, определяющие реакцию приложения на выбранное **событие**.

Метод – это процедура или функция, принадлежащая объекту. **Методом**, в том числе, является и процедура обработки события, связанного с объектом.

В среде программирования **Delphi** разработано множество **методов**, которые могут быть использованы для разнообразных целей.

Создание проектов

Для создания **проекта** нового программного приложения (типа **Application**) можно запустить среду программирования **Пуск=>Программы=>Borland Delphi 7=>Delphi 7** или, если в среде программирования уже выполнялась работа с каким-либо **проектом**, можно выполнить команды главного меню среды программирования **File=>New=>Application**

Сохранение файлов проектов

После формирования видимых элементов и создания программных текстов приложения (а еще лучше до этого) следует **сохранить все файлы проекта** приложения в папке, специально созданной или создаваемой при сохранении для **файлов именно этого проекта**. При **сохранении файлов проекта** рекомендуется **изменить стандартные имена проекта и модуля (или модулей)** типа **Project1** и **Unit1**, предлагаемые средой программирования на имена, соответствующее назначению проекта (буквы в имени должны быть только латинскими).

Кроме того, **файл проекта и файлы модулей должны быть сохранены с разными именами.**

Для **сохранения файлов проекта** приложения может быть использована в зависимости от конкретной ситуации любая из соответствующих команд из меню команды **File** (**Save, Save As..., Save Project As..., Save All**), причем для **сохранения всех файлов**, связанных с проектом приложения, следует сохранить как **файл проекта** (с расширением **dpr**), так и **файлы модулей** (с расширением **pas**). **Остальные файлы**, связанные с **файлом проекта** или с **файлами модулей**, сохраняются автоматически.

При **сохранении файлов проекта** следует обращать внимание на то, что, несмотря на возможность **сохранения файла проекта (и связанных с ним файлов)** и **файлов модулей (и связанных с ними файлов)** в разных папках (командами, соответственно, **Save Project As...** и **Save As...**), очень рекомендуется **сохранять** все эти **файлы** в одной папке, **специально созданной или создаваемой при сохранении для файлов именно этого проекта**. В противном случае становится возможным неправильное использование **файлов проекта**, и как следствие, неправильное выполнение приложения.

Запуск созданных приложений

Для запуска созданного приложения на выполнение в среде программирования можно выполнить команды меню **Run => Run** или соответствующую кнопку **Run** на панели инструментов **Debug** (можно также использовать клавишу **F9**).

При запуске приложения на выполнение в среде программирования сначала происходит **трансляция** (точнее, **компиляция**) программы, в результате которой создается **выполняемый файл** приложения с расширением **exe** (этот файл часто называют **exe-модулем**), который затем и выполняется под управлением среды программирования.



Создание и редактирование приложений в среде программирования

Особенности создания проекта, сохранения файлов проекта, открытия проекта в среде программирования.

Структура программного модуля.

Способы создания обработчиков событий.

Использование окна кода при создании и редактировании текста программы.

Компиляция и запуск приложения.

Особенности создания проекта, сохранения файлов проекта, открытия проекта в среде программирования

При создании проекта приложения создается несколько файлов: файл проекта (с расширением `dpr`), файлы модулей (`pas`), файлы форм (`dfm`), файл ресурсов проекта (`res`), файлы для сохранения параметров настройки компилятора (`cfg`), проекта (`dof`), среды программирования (`dsk`) и, возможно, другие файлы, связанные с проектом.

После формирования видимых элементов и создания программных текстов приложения (а еще лучше до этого) следует **сохранить все файлы проекта** приложения в папке, специально созданной или создаваемой при сохранении для **файлов именно этого проекта**. При **сохранении файлов проекта** рекомендуется **изменить стандартные имена проекта и модуля (или модулей) типа Project1 и Unit1**, предлагаемые средой программирования на имена, соответствующее назначению проекта (буквы в имени должны быть только латинскими).

Кроме того, **файл проекта и файлы модулей должны быть сохранены с разными именами.**

При **сохранении файлов проекта** следует обращать внимание на то, что, несмотря на возможность **сохранения файла проекта (и связанных с ним файлов)** и **файлов модулей (и связанных с ними файлов)** в разных папках (командами, соответственно, **Save Project As...** и **Save As...**), очень рекомендуется **сохранять все эти файлы** в одной папке, **специально созданной или создаваемой при сохранении для файлов именно этого проекта.**

В противном случае становится возможным **неправильное использование файлов проекта**, и как следствие, **неправильное выполнение приложения.**

При **создании нового проекта** или **открытии другого проекта**, если до этого еще **не были сохранены все файлы ранее открытого проекта** могут появляться окна с вопросами о сохранении изменений в тех или иных **файлах ранее открытого проекта.**

На эти вопросы следует ответить в зависимости от необходимости **сохранять или не сохранять эти изменения.**

Структура программного модуля

unit <имя модуля>;

interface

<интерфейсная часть>

implementation

<исполняемая часть>

end. <= признак конца модуля

Между исполняемой частью и признаком конца модуля могут также находиться инициализирующая (**initialization**) и завершающая (**finalization**) части модуля.

Способы создания обработчиков событий

В общем случае для создания **обработчика событий** или, иначе говоря, **процедуры обработки того или иного события**, связанного с выделенным компонентом или даже с несколькими выделенными компонентами, можно в окне **Object Inspector** открыть вкладку **Events (События)**, поместить указатель "мыши" в правую часть строки с названием выбранного события и выполнить двойной щелчок левой кнопкой "мыши".

Создать процедуру обработки события, наиболее характерного для данного компонента, или получить доступ к этой процедуре можно, поместив указатель "мыши" в область этого компонента в окне формы или на его значок в окне **Object TreeView и затем выполнив двойной щелчок левой кнопкой "мыши".**

Использование окна кода при создании и редактировании текста программы

Первоначально **окно кода** (с заголовком типа **Unit1.pas**) содержит минимальный **исходный текст**, обеспечивающий нормальное функционирование пустой формы в качестве окна Windows-программы.

Текст программы обычно вставляется между строками **{ \$R *.dfm }** и **end**.

Описания переменных и других объектов можно вставлять в области описаний выше строки **implementation**.

Окно кода обладает всеми свойствами текстового редактора, необходимыми для ввода и редактирования текстов программ, а также некоторыми дополнительными возможностями, связанными с контролем текста программы по мере его ввода.

Компиляция и запуск приложения

Для **компиляции созданного приложения** и его **запуска** на выполнение в среде программирования можно выполнить команды меню **Run => Run** или нажать соответствующую кнопку **Run** на панели инструментов **Debug** (можно также использовать клавишу **F9**).

В результате компиляции создается **выполняемый файл** приложения с расширением **exe** (этот файл часто называют **exe-модулем**), который затем и **выполняется под управлением среды программирования**.

Созданное приложение может выполняться **независимо от среды программирования** (для простого приложения достаточно запустить на выполнение **exe-модуль**).