**Открытая архитектура Delphi**

Компания Borland в развитии своих объектно-ориентированных средств разработки явно пришла к тому выводу, что повторное использование кода и объектная ориентация не являются единственными средствами повышения производительности программистов. С появлением Delphi разработчик может не только создавать и предоставлять своим коллегам готовые к использованию компоненты, но и расширять функциональные возможности среды, в которой он работает, с помощью так называемых "открытых интерфейсов". Такой подход позволяет использовать Delphi уже в роли общего ядра набора инструментальных средств на всех этапах создания прикладных систем - начиная с CASE-систем и заканчивая генерацией документации по создаваемым проектам, с полной их интеграцией в "святая святых" любой среды программирования - IDE. Рассмотрим основные возможности расширения функциональности среды Delphi для того, чтобы оценить степень "открытости" архитектуры этого инструмента.

**"Строительные блоки" приложений - компоненты**

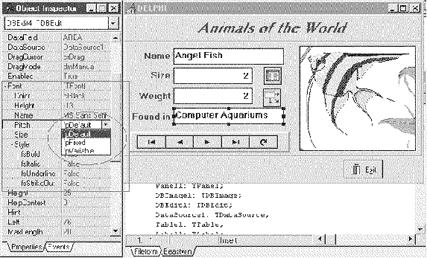
Как известно, фундаментальной основой визуальных средств Delphi является компонентный подход. В чем же он заключается?  
Delphi строится на базе компилятора объектно-ориентированного языка Object Pascal, продолжающего линию диалектов Pascal - Turbo Pascal и Borland Pascal. По мере своего развития, каждая очередная реализация Pascal компании Borland включала все новые расширения синтаксиса, отражающие последние достижения в области языков программирования. Если подходить к оценке качественных "ступеней" развития Pascal, особо следует отметить три из них, направленные на поддержку концепции повторного использования кода:

* модульная архитектура, с возможностью разделения интерфейсной и описательной частей (Turbo Pascal 4.0);
* средства объектной ориентации, со всеми, присущими ей характеристиками - наследованием, инкапсуляцией и полиморфизмом (Turbo Pascal 5.5);
* поддержка механизмов RTTI (Run-Time Type Information), позволяющих получать информацию о базовых характеристиках объектных типов (классов) и их экземпляров (объектов) с помощью языковых средств, непосредственно встроенных в системную библиотеку и структуру организации описаний классов (Delphi 1.0 - Object Pascal);

Следствием введения поддержки RTTI стала возможность создания *визуального* инструмента разработки приложений, каковым и является Delphi. На определенном уровне иерархии наследования базовой библиотеки классов Delphi появляется класс *TPersistent*, обеспечивающий необходимый уровень абстракции потокового ввода/вывода объектов (экземпляров классов). Его наследником выступает класс *TComponent*, определяющий основы поведения компонент Delphi VCL (Visual Component Library) в режиме *design-time* (этап "конструирования" приложения). На оконечных точках ветвей иерархии VCL находятся как таковые компоненты - готовые к визуальному использованию классы, непосредственно регистрируемые в рабочей библиотеке компонент и доступные из Палитры Компонент (Components Palette) IDE Delphi.  
Так как компоненты, используемые в разрабатываемой программе, написаны на том же языке, который используется при создании приложений, программист может достаточно легко создавать и регистрировать в Палитре свои компоненты, наследуя их от тех или иных представителей иерархии VCL или уже созданных программистом своих классов.   
С другой стороны, механизмы регистрации и дальнейшего наследования уже существующих стандартов динамического связывания (Windows DLL) и компонентной архитектуры (VBX в Delphi 1.0 и OCX - в 32- разрядной версии Delphi) позволяют использовать в Delphi доступные внешние инструменты (например, компиляторы C++) и, созданные с их помощью, программные блоки.  
*Самодостаточность Delphi для расширения набора доступных компонент является первым признаком открытости архитектуры этого инструмента.*

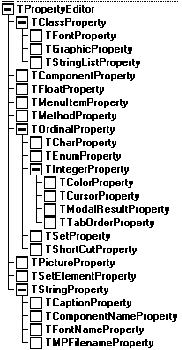
**Редакторы свойств и редакторы компонент - поведение IDE**

Логично, что при визуальном подходе к определению характеристик компонент (работа в design- time), необходимы средства определения редакторов специфических свойств в Инспекторе Объектов (Object Inspector).



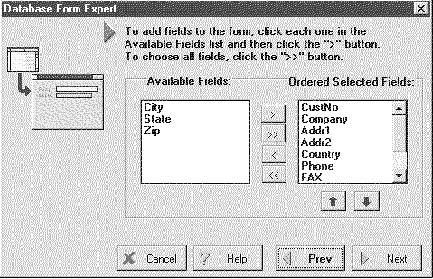
***Рис. 1***

Особенно остро встает для разработчиков компонент вопрос создания и использования редакторов свойств, когда свойства имеют сложный тип. Например, свойство может предоставлять ссылку на достаточно сложную структуру - запись или на строго определенных наследников одного из стандартных или пользовательских классов (возможные ситуации: 1) класс "множество данных" TDataSet - является предком и таблиц, и запросов, и хранимых процедур; можно сформулировать такую задачу, когда в качестве значения свойства в design-time должны выступать только запросы и таблицы, но, ни в коем случае - хранимые процедуры; 2) шрифт описывается рядом характеристик, представляемых вложенными записями).   
Delphi предоставляет разработчику ряд базовых классов, входящих в иерархию VCL, которые предназначены для создания редакторов свойств.



***Рис. 2***

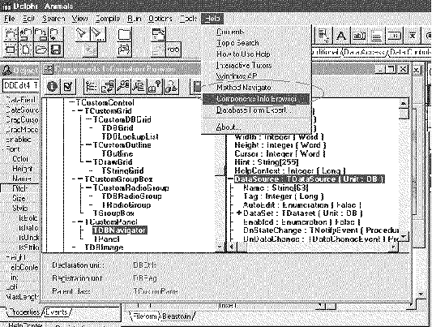
Стандартные редакторы свойств (более 20) являются наследниками базовых редакторов и, вместе с последними, доступны программисту для расширения/изменения функциональности, опять-таки, с использованием механизмов наследования и полиморфизма. Регистрация редакторов свойств и регистрации компонент аналогична регистрации самих компонент.   
*Так как редакторы свойств и редакторы компонент определяют design-time, существование таких редакторов и возможность расширения их функциональности являются вторым признаком открытости Delphi*.



***Рис. 3***

**Генерация кода - эксперты**

В процессе развития и, в том числе, визуализации средств разработки приложений, на фоне стандартизации пользовательского интерфейса в различных областях применения конечных систем, неотъемлемой частью таких инструментов стали генераторы кода и форм представления и ввода информации - эксперты.   
Кроме того, что Delphi включает ряд уже готовых к использованию экспертов (например, DataBase Form Expert, генерирующий формы и соответствующий код для простых приложений обработки баз данных с использованием запросов), эта среда программирования предоставляет разработчикам интерфейс для создания собственных экспертов, встраиваемых в IDE.  
Необходимо отметить, что функциональность таких экспертов может не ограничиваться на генерации кода, в силу того, что интерфейс экспертов дает возможность получения информации о внутренних объектах IDE, таких как палитра компонент. Вследствие этого, под общим названием "эксперты" могут фигурировать программные модули, позволяющие управлять повелением IDE, окна дизайнера и ее редактора исходных текстов, а также генерировать отчетную информацию о создаваемом проекте. (На приведенном выше рисунке вы можете увидеть эксперт, разработанный в Delphi и встроенный в IDE; функциональность этого эксперта заключается в предоставлении разработчику информации об иерархии наследования зарегистрированных компонент без компиляции; в данном случае доступ осуществляется через меню "Help", хотя возможна регистрация и в "галерее" шаблонов Delphi).

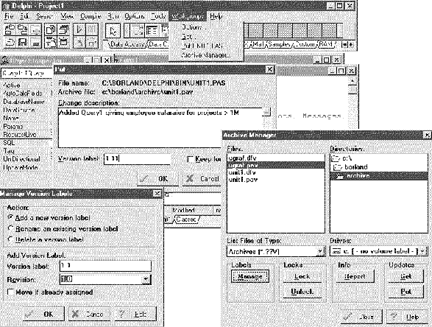


***Рис. 4***

*Наличие средств построения программных модулей генерации кода и обработки внутренней IDE- информации, называемых экспертами, являются третьим признаком открытости архитектуры Delphi.*

**Интеграция с внешними приложениями - открытые интерфейсы**

Как следствие возможности обмена информацией с IDE, реальным кажется и интеграция среды разработки Delphi с внешними инструментальными средствами - системами контроля версий, мониторами транзакций, CASE-системами и т.п.

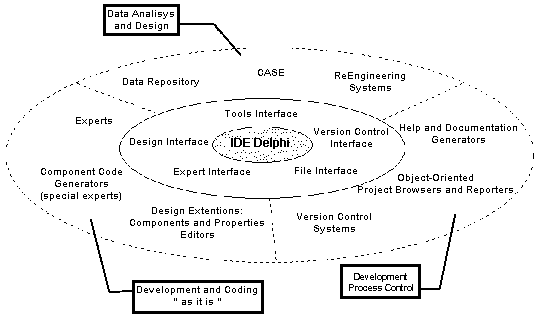


***Рис. 5***

И действительно, ряд производителей программных продуктов, относящихся к перечисленным категориям, заявил о поддержке ими Delphi на достаточно высоком уровне интеграции (подразумевая, например, для CASE-систем, не только генерацию кода в соответствии с синтаксисом Object Pascal, но и доступ к таким продуктам непосредственно из IDE). В качестве примера можно привести компанию Popkin Software (производителя CASE-средства System Architect), объявившую о поддержки Delphi в своих продуктах еще в августе 1995 года. Известен ряд систем контроля версий - Intersolv PVCS и MKS Source Integrity, способных работать с Delphi (32-разрядная версия PVCS входит в поставку Delphi Client/Server Suite 2.0, планируемого к выходу в первом квартале 1996 г.) и , например, мониторов транзакций (существует опыт взаимодействия с Novell Tuxedo и др.).  
*Описанные возможности интеграции с внешними приложениями на базе совокупности открытых интерфейсов, определяют четвертый признак открытости архитектуры Delphi.*

**Заключение**

Наконец, мы можем концептуально представить архитектуру открытых интерфейсов Delphi. Она приведена на следующей диаграмме:



***Рис. 6***

Вследствие такой открытости архитектуры Delphi, большое количество третьих компаний уже выбросило на рынок (или объявило о соответствующих планах) как различные расширения библиотеки компонент VCL (более 200 только коммерческих наборов компонент на октябрь 1995г.) так и средства интеграции своих продуктов (external-site interface).