

ОСНОВНЫЕ ПРИНЦИПЫ РЕАЛИЗАЦИИ ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ НА ЯЗЫКЕ PASCAL ПРИ АВТОМАТИЗАЦИИ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ

*К.Т.Н., доц. Терентьев В.П.; асс. Вардомацкая Е.Ю.,
асс. Черненко Д.В. (ВГТУ)*

Создавая прикладные программы, решающие задачи автоматизации технологических процессов, программист значительные усилия затрачивает на организацию ввода/вывода информации, т.е. создание интерфейса. Требования к оформлению интерфейса современных приложений постоянно растут. При этом усложняются и операционные системы. Уже усилия, затрачиваемые на программирование основного алгоритма, по которому решается конкретная задача, соизмеримы с работой по организации ввода/вывода. Проблема повторяется от программы к программе и требует выполнения трудоемких однотипных операций, отвлекающих внимание программиста от решения основной задачи.

Хорошим средством упрощения работы по организации ввода/вывода в операционной среде MS-DOS, его унификации с учетом современных требований к интерфейсу программ является разработанный фирмой Borland International пакет Turbo Vision (TV). Он представляет собой объектно-ориентированную библиотеку средств формирования пользовательского интерфейса.

TV позволяет создавать пользовательский интерфейс в текстовом режиме дисплея, опираясь на идеологию ООП - инкапсуляцию, наследование и полиморфизм.

При организации интерфейса решаются две задачи. Первая создание соответствующего образа на экране и вторая - посредством этого образа передача информации как от пользователя к ЭВМ, так и от ЭВМ к пользователю. Помимо решения этих задач при организации интерфейса приходится выполнять различные вспомогательные операции. В связи с этим объекты TV можно разбить на две группы: отображаемые и неотображаемые.

Все отображаемые объекты интерактивной программы представляют древовидную структуру, корнем которой является сама программа. Поскольку все объекты программы являются в некотором смысле обособленными друг от друга, возникают определенные трудности в передаче информации от объекта к объекту. В качестве механизма передачи информации в TV впервые вводится такое понятие как "обработка событий", когда любая передаваемая информация оформляется в виде т.н. события, представляющего на языке Pascal запись, отдельные поля которой характеризуют те или иные свойства передаваемой информации. Обработка событий начинается с текущего объекта, после чего передается тем или иным подэлементам объекта.

С появлением операционной среды Microsoft Windows пользователи ПЭВМ получили возможность работать в графической, интуитивно понятной среде. Графический интерфейс пользователя GUI (Graphical User Interface) не только значительно упростил обучение навыкам работы с приложениями, но и саму работу. Для выполнения команды достаточно выбрать из меню нужный пункт одним нажатием кнопки мыши. Кроме того, используемый многооконный режим позволяет работать сразу с несколькими программами. Однако при создании даже самого простого приложения необходимо создавать и программировать окна, меню, шрифты, диалоговые окна и множество других элементов.

С точки зрения программиста Windows представляет собой набор программ, обращаясь к которым можно относительно просто создавать эффективный интерфейс со всеми присущими Windows элементами. Все процедуры и функции операционной среды, вызываемые прикладными программами во время работы, назы-

ваются интерфейсом прикладной программы API (Application Programming Interface). Программный интерфейс Windows включает более 800 функций.

Продолжая принципы, заключенные в библиотеках TV, фирма Borland создала свой Windows API представленный в пакете BPW (Borland Pascal for Windows) совокупностью TPW-модулей, среди которых следует выделить следующие:

- WINPROCS - представляет доступ к стандартным процедурам и функциям Windows;
- WIN31 - предоставляет доступ к расширенным возможностям Windows 3.1.

Создание любых Windows-приложений с помощью пакета BPW базируется на объектной технологии. Прикладные программы, пользуясь услугами Windows, создают различные объекты и управляют ими. Типичными объектами являются окна, меню, пиктограммы, диалоговые блоки, перья и кисти для рисования, таймер и т.д. Даже выполняемая пользовательская программа рассматривается и обрабатывается как объект системы Windows.

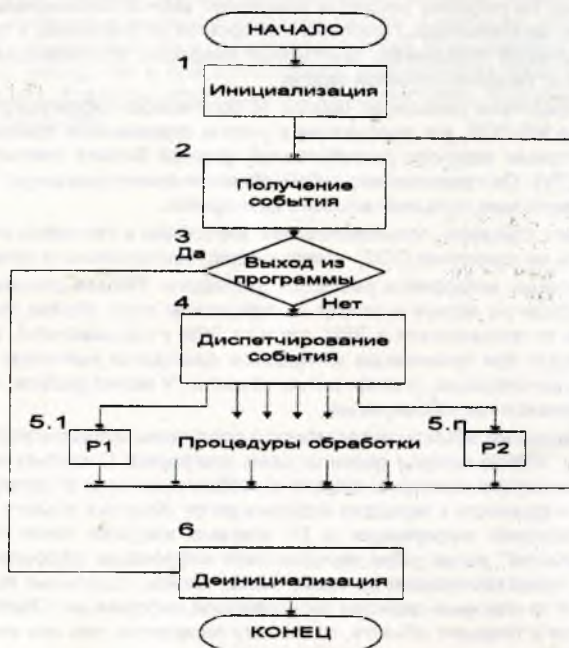


Рис.1. Алгоритмическая схема программы с событийно-управляемой архитектурой.

Управление окнами и множеством объектов программы - задача не из простых. Для ее решения в Windows используется механизм обработки событий, принципы которого были заложены ещё при разработке библиотек TV. Под событием понимают факт совершения элементарного действия, от которого может зависеть ход выполнения программы. Программа, ориентированная на обработку событий, имеет примерно следующую алгоритмическую структуру, представленную на рис.1. Достоинство такой схемы управления в ее универсальности. Чтобы в существующую программу добавить реакцию на новое событие, не нужно менять исходный код. Достаточно определить новую процедуру, которая при возникновении события будет осуществлять необходимые действия.

Написать просто и понятно Windows-приложение без использования объектно-ориентированного программирования (ООП) крайне сложно. По этой причине приложения для Windows достаточно долго создавала узкая группа программистов-профессионалов высшего класса. Для упрощения разработки программ фирма Borland предложила библиотеку OWL (Object Windows Library), которая фактически является объектно-ориентированной надстройкой над Windows API. Эта модульная и удобная в эксплуатации библиотека настолько упростила процесс написания Windows-программ, что теперь каждый желающий может писать программы, по виду ничем не уступающим изделиям Microsoft.

Существенно новым принципом программирования, позволяющим создавать Windows-приложения практически любому человеку, знакомому с ПЭВМ, явилось визуальное программирование (конструирование). Первыми эти принципы программирования были использованы в среде Visual Basic (VB). Этот мощный программный продукт относится к системам скоростной разработки приложений RAD (Rapid Application Development), что позволяет сделать процесс программирования более наглядным и интуитивно понятным. Новый стиль позволяет разработчику программы наглядно конструировать пользовательский интерфейс с помощью мыши, а не обычным для прежних времен путем: написанием кодов, их трансляцией и запуском программы, чтобы посмотреть, как это выглядит на экране. Однако VB не свободен от рода проблем, испытывает недостаток в механизмах, обеспечивающих хорошее структурирование, компактность и прозрачность программ. В нем отсутствует строгость объектно-ориентированного языка.

Delphi - следующий после VB шаг в развитии среды RAD. Интегрированная среда разработчика IDE (Integrated Development Environment) Delphi на первый взгляд выглядит похожей на VB 4.0. Однако между двумя продуктами существенное различие. Delphi базируется на языке Object Pascal, компиляторы которого были одними из самых быстрых. Наличие объектно-ориентированных расширений поддерживают хороший стиль программирования и более эффективный код. К тому же в отличие от VB Delphi генерирует действительно исполняемый exe-файл.

Delphi имеет большой набор стандартных объектов Windows, представленных в виде визуальных компонентов VC (Visual Component). VC можно создавать непосредственно в среде Delphi и заносить в библиотеку визуальных компонентов VCL (Visual Component Library). VCL является хранилищем VC, которые могут использоваться при создании Delphi-приложений. Все компоненты VCL отображены на экране в виде кнопок и легко доступны для пользователя.

Кроме этого, Delphi продолжает и развивает принципы событийного программирования, которые для сферы Windows являются наиболее естественными.

Из вышеизложенного материала видно, что создание программ с интерфейсом, соответствующим современным требованиям - задача, посильная для широкого круга специалистов. Однако, несмотря на кажущуюся простоту создания Windows-приложения, в Delphi нет замены основательной программистской подготовке. Человек, освоивший принципы объектно-ориентированного программирования и проектирования CUI сгенерирует с помощью Delphi лучший код, чем неопытный новичок. RAD - это всего лишь инструмент, который можно использовать и во благо и во зло. Для того чтобы научиться грамотно, создавать Windows-приложения, необходимо иметь следующие базовые знания:

- Знать основные принципы, программирования на языке Pascal;
- Понимать основы объектно-ориентированного программирования и уметь создавать объектно-ориентированные программы;
- Знать общую структуру и порядок использования объектных библиотек (например, таких как OWL).