

Компонентный Паскаль и система BlackBox

В программировании, да и не только, есть принцип KISS, согласно которому простота провозглашается как важная цель проектирования. Аббревиатура принципа расшифровывается как *keep it simple, stupid* — «делай это проще, дурачок» или *keep it short and simple* — «делай короче и проще». Этот принцип можно распространить не только на программы, но и на системы программирования.

Блэкбокс можно считать образцом простой и в то же время эффективной системы программирования.

Система Блэкбокс имеет много интересных возможностей, отличающих ее от других именно простотой и эффективностью.

Простота архитектуры. Приложение, созданное в Блэкбоксе, состоит из подсистем, которые в свою очередь состоят из модулей. Это могут быть и подсистемы, реализующие функции приложения, и подсистемы самого Блэкбокса. Модуль представляет собой совокупность данных и процедур. Если модуль использует данные или процедуры другого модуля, в его начале помещается оператор `IMPORT`, в котором указывается полное имя модуля, состоящее из двух частей: имени подсистемы и имени модуля. Данные и процедуры, которые должны быть доступны из других модулей, помечаются символом «*».

Подсистема просто отображается на файловую систему. Все файлы подсистемы находятся в подпапке с именем подсистемы. В папке подсистемы создаются подпапки со стандартными именами: `Mod` – исходные коды, `Sym` – символьные файлы, `Code` – кодовые файлы, `Docu` – документация, `Rsrc` – ресурсные файлы. Результат трансляции модуля – символьный файл `*.osf` и кодовый файл `*.ocf` с теми же именами, что и модуль. Они размещаются в подпапки `Sym` и `Code` соответственно, автоматически создаваемые, если их не было. Так же автоматически создается папка подсистемы. Исходный код модуля сохраняется в файл с расширением `*.odc` в подпапке `Mod`. Если создать файл с документацией модуля (с тем же именем) и поместить его в подпапку `Docu`, ваша подсистема становится частью всей системы. Информация о ней становится частью общего репозитория, доступного через пункт главного меню «Инфо-Обзор подсистем».

В зависимости от конфигурации система содержит 60-80 модулей. Для сравнения система `MS Visual Studio 9.0` имеет более 13 тысяч файлов и использует 130 типов файлов. Отметим и немалое число архитектурных понятий: модуль, сборка, класс, файл, пространство имен.

Простота интерфейса. Папка `Rsrc` подсистемы содержит различные файлы, необходимые для работы подсистемы: формы, файлы с данными и другие. Один из таких файлов со стандартным именем `Menus.odc` содержит информацию о меню подсистемы (перечень функций подсистемы). Вот начало системного файла-меню:

```
MENU "Файлы"
  "Новый"      "N"      "StdCmds.New"    ""
  "Открыть..." "O"      "HostCmds.Open" ""
```

В нем указывается название каждого пункта меню, горячая клавиша (если она есть) и вызываемая процедура. Когда система стартует, она собирает файлы-меню всех подсистем и формирует общее меню. Таким образом, для подключения новой подсистемы достаточно разместить в рабочей папке новую подпапку. Меню открывается для редактирования, так что в него легко добавить новый пункт и убрать лишние.

Для создаваемых подсистем доступны все возможности, как и для системы в целом: поиск по документации, анализ модулей, определение интерфейсов модулей и другие.

Документы «три в одном». Программы в системе хранятся в файлах с расширением «.odc» (`Oberon DoCument`). Это не обычные текстовые документы. Они больше похожи на документы `Word`'а и могут хранить вместе с программой данные и элементы диалоговых форм. На рисунке 1 приведен пример программы вычисления произведения двух переменных, для которых встроены элементы ввода.

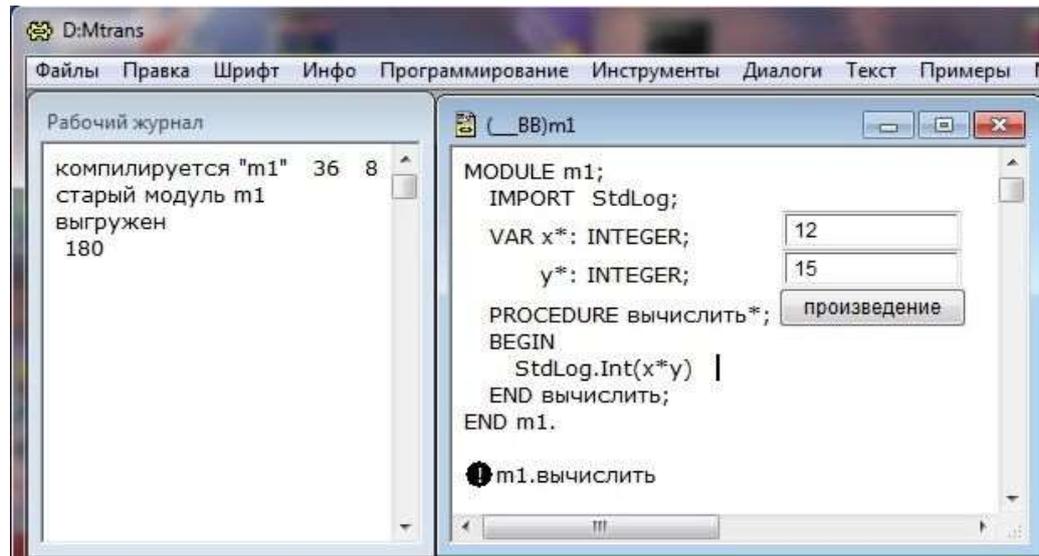


Рис. 1. Программа + командер + форма в одном документе

Программа запускается нажатием либо кнопки «произведение», либо специального символа-кнопки, называемого *командером* (или *командиром*). Результат выводится в рабочий журнал. Элементы диалоговой формы добавляются через меню «Диалоги». Для элемента достаточно указать, с каким внешним объектом он связывается. *Внешние объекты* – это те, которые помечены знаком «*». Без объяснений понятно, в чем тут удобство.

Создание формы в три приема. В большинстве случаев поступают по-другому: строят независимые формы. И это делается очень просто! Всего три простейших действия (см. рис.2): 1) выбор пункта меню «Диалоги – Новая форма»; 2) ввод имени модуля в окне «Новая форма...» или перенос имени из текста программы клавишами Ctrl+C и Ctrl+V; 3) нажатие на кнопку «Ладно». Система автоматически просматривает текст модуля и строит элемент формы для каждого внешнего объекта. При этом учитывается его тип и имя. Например, для логических переменных строится элемент-флажок, для имен процедур – кнопка. При необходимости форму можно подредактировать, меняя размеры и позиции элементов. Сгенерированную форму необходимо сохранить (в примере – в папку Rsrc).

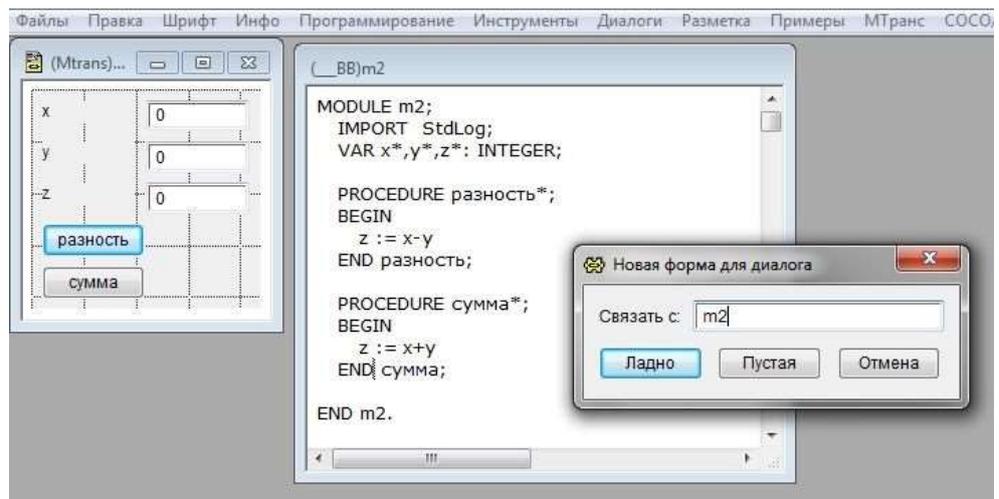


Рис. 2. Создание формы модуля

Для запуска формы в программу вставляется (Ctrl+Q) командер, вызывающий процедуру OpenAuxDialog стандартного модуля StdCmds (стандартные команды). При вводе значений в форме происходит изменение связанных с элементами переменных. Для изменения выходных значений на форме используется процедура «Update». Самое интересное, что диалоговых форм может быть несколько, и все они будут работать. При этом в самом модуле ничего менять не надо. Попробуйте это сделать в Delphi или Visual Studio.

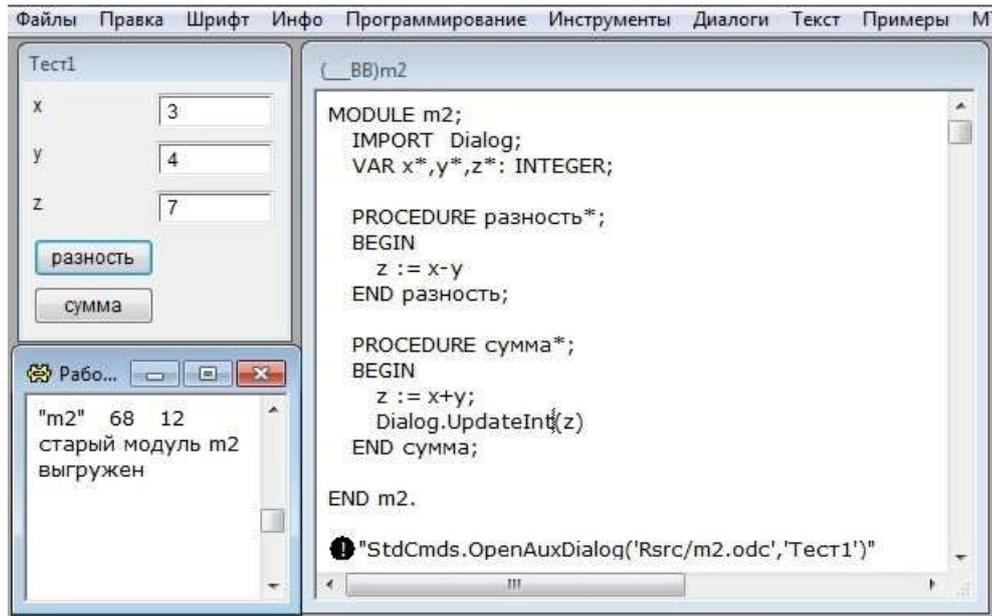


Рис. 3. Работающая форма модуля

Следующая возможность, реализованная в модуле `i21sysIn`, позволяет вводить информацию непосредственно из того же файла, где находится программа! На рисунке 4 процедура «тест» запускалась дважды. Сначала по командеру, в котором после имени находятся значения, считанные процедурой «`In.Int`». При втором запуске по командеру предварительно был выделен текст со значениями 6 и 7, которые и были считаны.

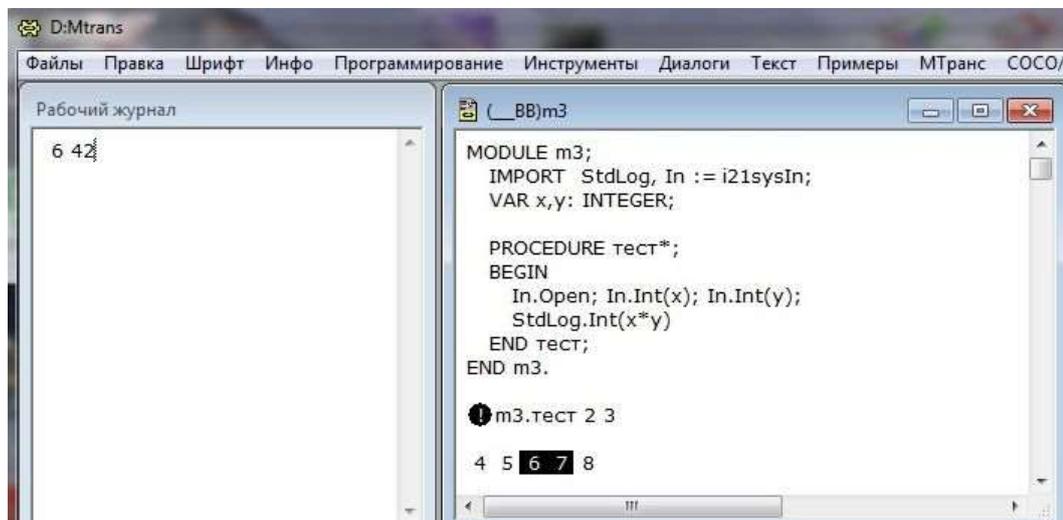


Рис. 4. Работающая форма модуля

Способ с выделением части текста особенно удобен, если процедура считывает несколько чисел, количество которых может быть разным. Также процедура может считывать из файла. Для этого необходимо создать пункт меню, вызывающий эту процедуру. Файл с данными открывается в окне, когда оно активно, инициируется пункт меню с вызовом процедуры. В модуле может быть несколько процедур, каждая со своим набором тестовых данных.

Шаблоны редактирования. Если вы уже заинтересовались системой и попробовали её запустить, сделайте следующее. Откройте новый файл и наберите в открывшемся окне слово «`module`». Нажмите клавишу F5. Вы получите заготовку для модуля. Сдвиньте курсор две строки ниже, наберите «`прос`», опять нажмите F5. То, что получится, изображено на рис. 5. Теперь достаточно добавить имя модуля и имя процедуры и продолжить набор тела процедуры. Такие шаблоны

разворачиваются и для других конструкций: IF, WHILE. Набирать можно лишь несколько первых букв (даже одну!).

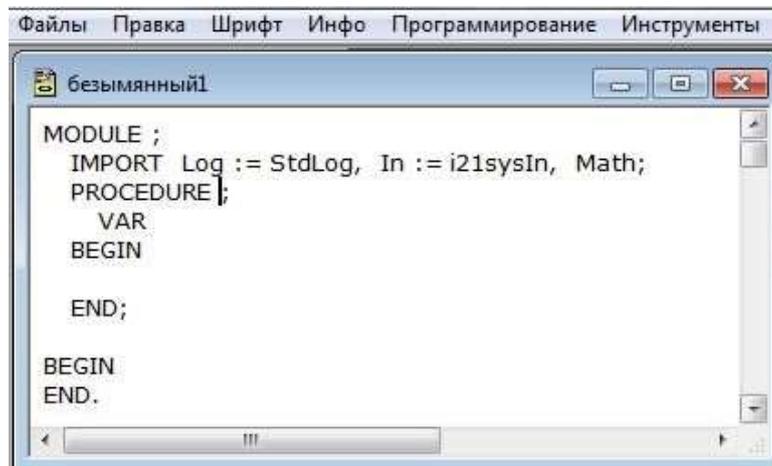


Рис. 4. Макрорасширения конструкций по F5

Этот механизм называется *макрорасширениями* и реализован в модуле i21sysEdit. Разобравшись в его исходном тексте можно добавить и свои макрорасширения. Кстати этот модуль поддерживает и другой набор ключевых слов – русских. Попробуйте набрать «модуль» или «проц».

Ошибки по месту. Модуль компилируется по клавише F5. На месте каждой зафиксированной ошибки появляется специальный маркер в виде перечеркнутого квадратика (см. рис. 5). Это достаточно удобный и оригинальный способ. Курсор перемещается по маркерам по клавише Ctrl+E, при этом в нижней части основного окна отображается текст ошибки. Если нажать кнопку мыши на маркере, текст появится непосредственно в тексте программы.

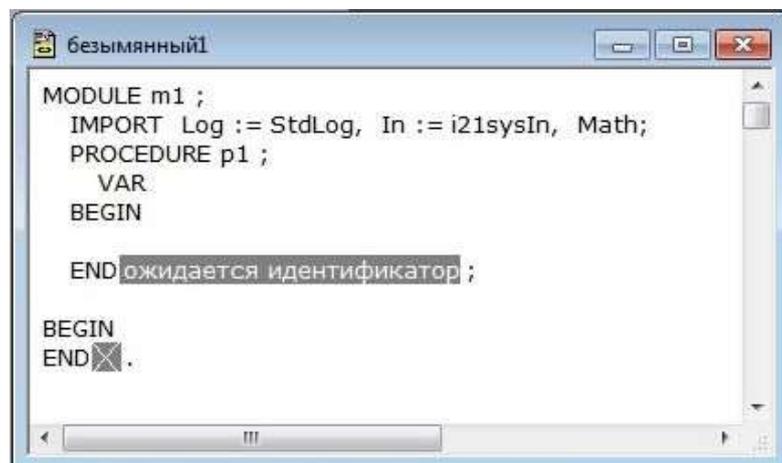


Рис. 5. Макрорасширения конструкций по F5

Окна с вьюшками. На рисунке 5 показан один из примеров использования встроенных в документ картинок. В оригинале такая картинка называется view, а в переведенной документации вьюшкой. У авторов перевода были разные варианты перевода: картинки, окошки, отображения... Слово «вьюшка» вообще-то имеет несколько значений в русском языке. Картинки можно было бы назвать «интервьюшками», так как они позволяют организовать интерактивное взаимодействие. Если на картинку с вращающимися кубиками навести мышку и нажать кнопку с одновременным нажатием клавишей «Ctrl», появится диалоговое окно, в котором можно задать цвета граней куба (это и показано на рисунке).

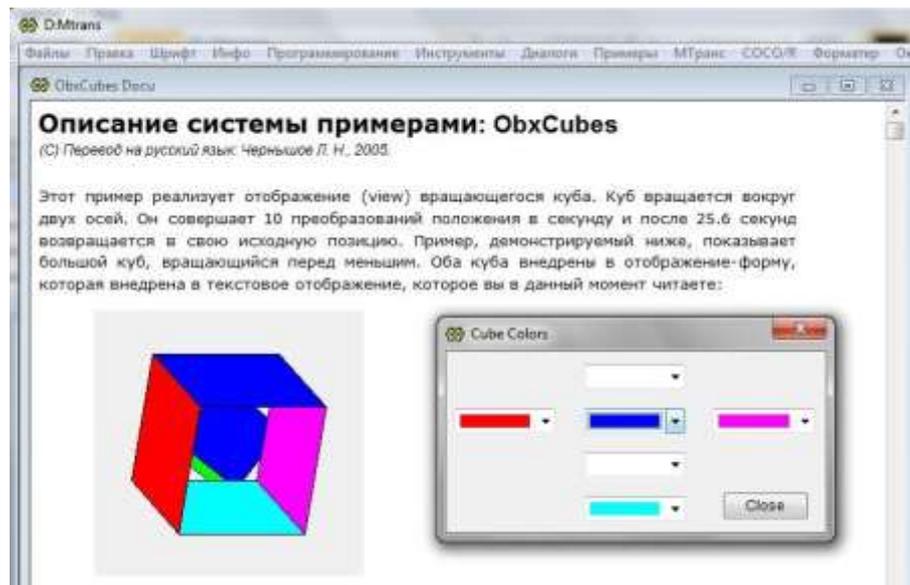


Рис. 6. Динамическая трехмерная картинка в окне

Авторы надеются, что приведенные сведения привлекут внимание тех читателей, которые до этого не были знакомы с черным ящиком.

1. Н. Вирт. Долой «жирные» программы. Открытые системы, №6, 1996.
2. Сайт «Информатика-21» -- международный общественный научно-образовательный проект. <http://www.inr.ac.ru/~info21/>