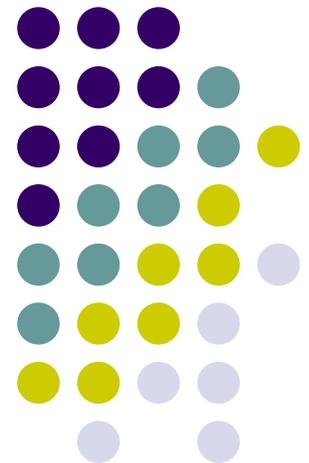

ОСНОВЫ ПРОГРАММИРОВАНИЯ В ОБОЛОЧКЕ Free Pascal





Язык программирования Паскаль (Pascal) был создан в **1970** г. **Никлаусом Виртом** – профессором информатики федерального технического университета в Швейцарии и назван в честь французского философа и математика XVIIв. Блеза Паскаля.

При создании этого языка Вирт большое внимание уделял структурному программированию, благодаря которому конструкции языка Паскаль позволяют создавать надёжные, легко проверяемые программы с ясной и чёткой структурой.

Позднее была создана интегрированная среда программирования **Turbo Pascal**.



Структура программы на языке Паскаль

1. Заголовок программы:

Program имя программы

В имени программы **используется только латинский алфавит, цифры и знак подчёркивания**; русские буквы, знаки препинания и пробелы между словами не допускаются!



2. Раздел описаний

описываются все идентификаторы объектов, используемые в данной программе. (Описать идентификатор значит указать его имя и тип.)

Uses <описание модулей> (Например, **uses Crt**)

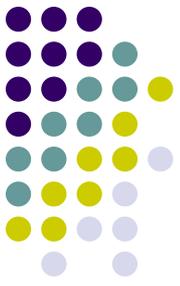
Const <описание констант> (Например, **const a=5**)

Type <определение типов данных>

Var <объявление переменных>

(Например, **var a, x: тип**)

Procedure (function) <описание процедур и функций>



3. Раздел операторов (тело программы)

Указывается последовательность действий, которые необходимо

выполнить над описанными объектами.

Оператор 1;

Оператор 2;

.....

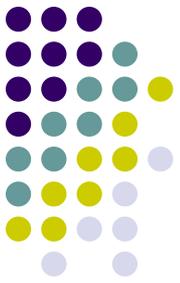
Оператор n;

End.

Программа записывается в форме отдельных предложений – операторов, которые разделяются ;

Каждый оператор представляет собой законченную фразу языка и определяет однозначно трактуемый этап обработки данных.

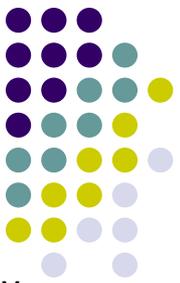
В конце программы ставится .



Стандартные типы данных:

- Целые числа – **integer**;
- Вещественные (действительные) числа – **real**;
- Символьный тип (константы и переменные, значения которых представляют символ, обрамлённый апострофами) – **char**;
- Строковый тип – **string**.

Операторы



(команды программы):

- **Оператор присваивания :=** - присваивает переменной конкретное значение; (например, $x := x * 8$ или $x := 5$)
- **Оператор условного перехода** (разветвляющийся алгоритм)

if условие **then** действие1 **else** действие2

- **Цикл с предусловием**

while условие **do begin** действие1; действие2; **end**

(**пока** условие сохраняет значение *true*, **выполнять** действие)

- **Цикл с постусловием**

repeat действие **until** условие

(**повторять** действие **до тех пор, пока** условие сохраняет значение *false*)

- **Цикл с параметром**

При возрастании параметра x **for** $x := x1$ **to** $x2$ **do** действие

При убывании параметра x **for** $x := x1$ **downto** $x2$ **do** действие

Процедуры (стандартные алгоритмы обработки информации)



- **Read**– ввод данных с клавиатуры;
- **Readln** – после ввода значений курсор переводится на следующую строку;
- **Write** – вывод указанных данных на экран (текст пишется в апострофах) ;
- **Writeln** - после вывода значений курсор переводится на следующую строку.

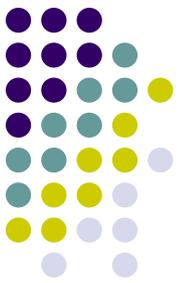
Например:

Write ('Введите значение x') – на экран будет выведена фраза **Введите значение x**

Read (x) – программа запомнит введённые с клавиатуры данные как переменную *x*

Write ('x:=', x) – после текста *x:=* подставится текущее значение переменной *x*

Модули Free Pascal

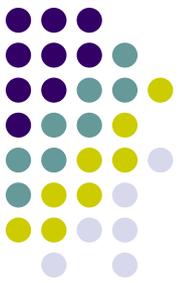


- Модуль Crt (*Cathod ray tube – электронно-лучевая трубка*) обеспечивает управление текстовым режимом работы экрана: перемещает курсор по экрану, изменяет цвет фона экрана и символов, создаёт окна, управляет звуком и др.

В любой программе модуль Crt используется в двух случаях:

- 1. Для очистки экрана – процедура **ClrScr**.***
- 2. Для задержки исполнения программы до нажатия любой клавиши – функция **ReadKey** (чтение символа с клавиатуры).***

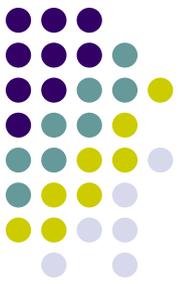
- Модуль Graph содержит набор процедур, позволяющих выполнять графические операции.



Процедуры модуля Crt:

- **ClrScr** - очищает экран или окно: экран заполняется цветом фона, а курсор устанавливается в левый верхний угол;
- **TextMode(Mode)** – выбор режима текстового экрана. По умолчанию установлен цветной режим Co80 из 25 строк по 80 символов в строке;
- **TextColor(farbe)** – определяет цвет текста;
- **TextBackground(farbe)** – определяет цвет фона;
- **Window(x1,y1,x2,y2)** – установка размеров окна для вывода текста;
- **Delay(x)** – задержка исполнения программы на x миллисекунд.

Алгоритм создания программы в оболочке Free Pascal



1. В горизонтальном меню выбрать **File**→**New**.
2. В экранном редакторе набрать текст программы.
3. Сохранить программу **File**→**Save** и набрать имя программы на английском языке.
4. Скомпилировать программу **Compile** →**Compile**. Если в тексте программы не обнаружено ошибок, то в диалоговом окне появится надпись **Compile Successful: press any key**. В противном случае – исправить ошибки и выполнить снова пункты 3 и 4.
5. Запустить программу на исполнение **Run** →**Run**.
6. Закрыть окно экранного редактора.
7. Выйти из программы **File**→**Exit**.