

ЦЕНТР ДЕТСКОГО (ЮНОШЕСКОГО) НАУЧНО-ТЕХНИЧЕСКОГО ТВОРЧЕСТВА



## «ПРОГРАММИРОВАНИЕ В СИСТЕМЕ TURBO PASCAL»

### *Часть 1*

### *Базовые конструкции языка Pascal*

#### УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

для учащихся заочной школы юных техников, обучающихся по направлению  
«программирование»

Автор - педагог дополнительного  
образования ЦНТТ Андреева И.Ю.

г. Армавир  
2011

## Пояснительная записка

Предлагаемое методическое пособие представляет собой сборник заданий к первой части учебного курса «Программирование в системе Turbo Pascal» и предназначено для детей и молодежи в возрасте от 14 до 18 лет, обучающихся в заочной школе юных техников по направлению «Программирование».

Цель данного пособия - помочь учащимся в проведении самостоятельной работы по изучению основ программирования и разработке компьютерных программ в системе программирования Turbo Pascal.

В методической разработке содержатся **учебный план** «Базовые конструкции языка Pascal» - первого модуля курса «Программирование в системе Turbo Pascal», **порядок выполнения контрольных заданий и варианты заданий** к тематическим блокам модуля. Каждое задание содержит теоретический материал, разбор характерных примеров и задач, методические рекомендации по их выполнению. В конце каждого задания дается перечень задач для самостоятельного решения и список вопросов для самоподготовки.

## Учебно-тематический план

### Часть 1. Базовые конструкции языка Pascal

- ◆ Работа в среде Turbo Pascal на примере разработки программ линейной структуры. Вычисление арифметических выражений.
- ◆ Организация ветвлений. Условный оператор IF. Оператор выбора Case ..of.
- ◆ Циклы for, while, repeat ... until.
- ◆ Работа с одномерными массивами.
- ◆ Работа с двумерными массивами.
- ◆ Символьные и строковые типы данных в Pascal.
- ◆ Использование процедуры и функции в Pascal.
- ◆ Комбинированный тип Запись.
- ◆ Файловый тип данных. Работа с файлами.

### Порядок выполнения контрольных заданий

1. Внимательно прочитать и уяснить условие задачи, которую предстоит решить.
2. Ознакомиться с необходимым теоретическим материалом, примерами решения задач. Использовать список рекомендуемой литературы.
3. Разработать алгоритм и написать текст программы. Отладить программу на компьютере.
4. Подготовить отчет.

Каждый **отчет оформляется** в виде пояснительной записки и должен содержать следующие элементы:

- ◆ титульный лист;
- ◆ текст пояснительной записки;
- ◆ список использованной литературы;

Содержание **пояснительной записки отчета**:

1. Постановка задачи.
2. Краткие сведения об особенностях применяемых операторов и методов (теоретическое введение).
3. Описание программы:
  - ◆ описание алгоритма решения задачи (в виде блок-схемы);
  - ◆ описание входных и выходных данных программы;
  - ◆ перечень входных данных (тестовые примеры)
  - ◆ программный код на языке Pascal.

## ЗАДАНИЕ № 3

### Циклы **for**, **while**, **repeat ... until**

**Цель работы.** Разработать алгоритмы и программы решения задач. Исходные данные – целые числа (тип **integer**). Использовать операторы цикла **for**, **while**, **repeat ... until**. Исходные данные должны вводиться с проверкой на область допустимых значений.

#### **Теоретическая часть:**

Алгоритм циклической структуры - это алгоритм, в котором происходит многократное повторение одного и того же участка программы. Такие повторяемые участки вычислительного процесса называются циклами. Программа циклической структуры содержит один или несколько циклов. Различают детерминированные циклы с заранее известным числом повторений и итерационные циклы, в которых число повторений заранее неизвестно. Изменяющаяся в цикле переменная называется параметром цикла.

Для организации цикла необходимо выполнить следующие действия:

- 1) задать перед циклом начальное значение параметра цикла;
- 2) изменять параметр перед каждым новым повторением цикла;
- 3) проверять условие повторения цикла;
- 4) управлять циклом, т.е. переходить к его началу, если он не закончен, или выходить из него по окончании.

В языке Паскаль существует 3 вида циклов:

- 1) цикл с параметром или цикл типа **for**,
- 2) цикл с предусловием или цикл типа **while**,
- 3) цикл с постусловием или цикл типа **repeat ... until**.

В цикле типа **for** число повторений известно заранее, в циклах типа **while** и **repeat ... until** число повторений цикла заранее неизвестно, производится проверка условия повторения цикла: в цикле типа **while** - перед циклом, в цикле типа **repeat ... until** - после его окончания.

В циклах типов **for** и **while** повторяющаяся часть (тело цикла) состоит из одного оператора, если требуется выполнить в цикле несколько операторов, они заключаются в операторные скобки **begin ... end**, образуя составной оператор. В цикле типа **repeat ... until** тело цикла помещается между зарезервированными словами языка **repeat** и **until**, операторные скобки не требуются, в названии цикла его тело условно обозначается тремя точками.

С помощью цикла типа **for** удобно находить суммы, произведения, искать максимальные и минимальные значения и т.п. При нахождении суммы некоторой переменной, например **S** присваивается значение 0, затем в цикле к этой переменной прибавляется соответствующий член заданной последовательности. При нахождении произведения переменной присваивается значение 1, затем в цикле эта переменная умножается на общий член последовательности.

#### **Пример цикла типа for**

Вычисление **n** чисел Фибоначчи:

$F_1=1; \quad F_2=1; \dots; \quad F_n=F_{n-1}+F_{n-2}$ ,  
например  $F_3=F_2+F_1=1+1=2$ ;  $F_4=2+1=3$  и т.д.

```
program fib; {Нахождение чисел Фибоначчи}
  var x, y, z, i, n : integer;
begin
  writeln ('Введите n');
  read (n);
  x:=1; y:=0;
```

```

for i:=1 to n do
  begin
    z:=x; x:=x+y; y:=z;
    write ( ' ', x );
  end;
readln;
end.

```

### Пример цикла типа while

Дано натуральное число n. Подсчитать количество цифр данного числа.

Подсчет количества цифр начнем с последней цифры числа. Увеличим счетчик цифр на единицу. Число уменьшим в 10 раз, тем самым избавляемся от последней цифры числа. Далее с получившимся числом проделываем ту же последовательность действий и т.д. , пока число не станет равным нулю.

```

program cycl_while;
uses crt; {вызов модуля Crt для управления режимом экрана}
var
  n, m, k : integer;
begin
  clrscr ; {процедура очистки экрана из модуля Crt}
  writeln ('Введите натуральное число ');
  readln (n);
  m:=n;
  k:=0;
  while m<>0 do
    begin
      k:=k+1; {увеличиваем счетчик на 1}
      m:=m div 10; {удаляем в числе последнюю цифру }
    end;
  writeln('В числе ',n,' - ',k,' цифр');
  readln;
end .

```

### Пример цикла типа repeat ... until

Вычислить сумму целых чисел вводимых с клавиатуры. Числа вводить до тех пор, пока не будет введен 0.

Вычисляем сумму s до тех пор, пока n <> 0, где n - очередное вводимое число.

```

program sum;
uses crt; {вызов модуля crt для управления режимом экрана}
var
  s, n : integer;
begin
  clrscr ;
  s:=0;
  repeat
    write( 'Введите число ');
    readln (n);
    s:=s+n;
  until n = 0;
  writeln ('s= ', s : 5 );
  readln;
end.

```

Цикл повторяется до тех пор, пока условие записанное после ключевого слова `until`, будет ложным (не выполняется). Как только это условие выполнится, происходит выход из цикла. После окончания цикла производится печать результата (оператор `writeln`). Отметим, что цикл с предусловием (типа `while`) может не выполниться ни разу, цикл с постусловием `repeat ... until` выполнится по крайней мере 1 раз. Когда число повторений цикла неизвестно заранее, применяются циклы с предусловием или с постусловием. Когда число повторений цикла известно заранее, как правило, применяется цикл типа `for`. Но любой цикл типа `for` можно заменить циклом с предусловием или постусловием.

### **Задачи для самостоятельного решения:**

1. Создайте программу, которая вычисляет сумму чисел от 1 до  $N$ . Значение  $N$  ( $N$  должно быть меньше 100) вводится с клавиатуры.
2. Напишите программу печати таблицы перевода расстояний из дюймов в сантиметры (1 дюйм - 2,5 см) для значений длин от 1 до 20 дюймов.
3. Используя цикл `while`, напишите программу вывода всех четных чисел от 2 до 100 включительно.
4. Создайте и произведите отладку программы, вычисляющей сумму квадратов чисел от 1 до введенного целого числа  $n$ .
5. Используя цикл `while`, напишите программу определения суммы всех нечетных чисел от 1 до 99 включительно.
6. Используя цикл `while`, напишите программу определения идеального веса для взрослых людей по формуле; Идеальный вес = рост - 100. Выход из цикла - значение роста 250.
7. Используя цикл `repeat`, напишите программу-фильтр, которая вводит любые символы, но комментирует только буквы русского алфавита. Завершение работы программы — по нажатию буквы «Я».
8. Используя цикл `repeat`, напишите программу, которая требует ввод пароля, например, числа 111, и, если пароль правильный, заполняет все строки экрана сообщением «Молодец!!!». Если после пятой попытки пароль неверен, выйти из программы.
9. Создайте программу получения в порядке убывания всех делителей данного числа.
10. Создайте программу определения наибольшего общего делителя двух натуральных чисел.
11. Создайте программу определения наименьшего общего кратного двух натуральных чисел.
12. Создайте программу, подсчитывающую количество цифр введенного целого неотрицательного числа. (Можно использовать операцию целочисленного деления для последовательного уменьшения числа на один разряд.)
13. Создайте и произведите отладку программы, определяющей максимальное из всех введенных чисел. (Пусть признаком конца ввода чисел служит число 0.)
14. Найдите наибольшее и наименьшее значения функции  $y = 3x^2 + x - 4$ , если в заданном интервале  $[a, b]$   $x$  изменяется с шагом 0,1.
15. Вычислите сумму квадратов  $N$  четных натуральных чисел.
16. Вычислите:
  - 1)  $1 + 2 + 4 + 8 + \dots + 210$ ;
  - 2)  $(1 + 2)(1 + 2 + 3)\dots (1 + 2 + \dots + 10)$ .

17. В бригаде, работающей на уборке сена, имеется  $n$  косилок. Первая из них работала  $t$  часов, а каждая следующая на 10 минут больше, чем предыдущая. Сколько часов проработала вся бригада?

### **Контрольные вопросы:**

1. Каково назначение операторов повтора (цикла)?
2. Какие требования предъявляются к выражениям, управляющим повторениями?
3. В чем различия операторов повтора `while` и `repeat`?
4. В каких случаях предпочтительнее использовать для организации циклон! оператор повтора `for`? Что записывается в заголовке этого оператора?
5. Каким образом в операторе цикла `for` описывается направление изменения значения параметра цикла?
6. Какие ограничения накладываются на использование управляющей переменной (параметра цикла) в цикле `for`?
7. Какие правила пунктуации необходимо соблюдать при записи операторов;
8. Что такое вложенные циклы? Какие дополнительные условия необходимо соблюдать при организации вложенных циклов?

### **Литература:**

1. Бабушкина И.А. Практикум по Турбо Паскалю.- М.: АБФ, 1998.
2. Емелина Е.И. Основы программирования на языке Паскаль. - М.: Финансы и статистика,1997.
3. Зуев Е.А. Turbo Pascal. Практическое программирование.- М.: ПРИОР, 1997 .
4. Немнюгин С., Перколаб Л. Изучаем Turbo Pascal. – СПб.: Питер, 2001 .
5. курс, С-Петербург, 2002
6. Фаронов В.В. Турбо-Паскаль 7.0. Начальный курс. Учебное пособие.- М.: "Нолидж",1998.