Урок 5. Циклы, массивы.

Наша программа хорошо вычисляет квадратные уравнения для одного набора констант. Однако, если таких наборов несколько, нам приходиться дважды писать практически один и тот же код программы. А если нам надо одновременно решить 10 различных примеров с квадратными уравнениями? Для того , чтобы несколько раз выполнять одну и ту же программу используются **циклы**.

Цикл имеет вид

**for** i:=0 **to** 2 **do**

Это означает, что *цикл* будет выполняется 3 раза. На первом шаге переменная i примет значение 0 , на втором шаге она примет значение 1, на третьем значение 2.

Та часть программы, которую нам надо использовать несколько раз с помощью операторов **for , to и do** я надо заключить между операторами begin и end.

Если написать.

**for** i:=0 **to** 2 **do**

**begin**

**Writeln(**'**Надпись**'**);**

**end;**

То слово '**Надпись**' будет выведено три раза.

Для удобной работы с *циклами* в языке Паскаль есть особый тип данных – **массив**.

Переменная типа массив задаётся так:

b\_arr : **array of** Integer = (2,6,1);

где оператор **array of** задаёт, что это массив . Далее мы пишем, какого типа это массив . Это может быть массив целых чисел, дробей или строк. Далее мы можем задать начальные значения этого массива. В данном случае мы создали массив из трёх элементов со значениями 2,6,1. Традиционно *массивы* начинаются не с первого , а с нулевого элемента. Затем идёт первый элемент *массива*, затем второй и т.д.

Теперь если мы хотим передать в функцию нулевой элемент *массива* , мы пишем Discriminant(b\_arr[0]…

Также мы можем использовать конструкцию вида

Discriminant(b\_arr[i]…

В этом случае мы передадим в функцию такой элемент *массива*, какое значение переменной i.

*Массивы* идеально сочетаются с *циклами* ведь при выполнении функции

**for** i:=0 **to** 2 **do**

мы будем последовательно передавать функции Discriminant сначала нулевой элемент нашего массива, потом первый и наконец второй. Но при этом нам надо написать только один цикл вместо того, чтобы три раза писать одно и то же. Файл lesson5\_1.pas иллюстрирует, как это работает. Изучите его и модифицируйте так, чтобы он выдавал решения уравнений во всех случаях. Модифицируйте программу так, чтобы в массиве было 10 элементов, и соответственно решалось 10 уравнений.