Программирование в Lazarus для школьников.

Занятие № 12.

Оператор ветвления.

Матыцин Игорь Владимирович

Учитель математики и информатики

МБОУ СОШ с. Девица

Цель: познакомить с оператором ветвления и научить решать простые задачи с не линейными алгоритмами.

Девица 2012.

Бывают такие задачи, в которых необходимо принять какое-либо решение. Это решение может повлиять на ход всего решения. Например при решении квадратных уравнений необходимо искать дискриминант, который говорит о том сколько у уравнения будет корней и будут ли они вообще. При составлении условий пишут :Если дискриминант больше нуля то у уравнения есть корни. В программировании точно так же можно написать слово если, только для языка паскаль оно будет писаться так: if. И после этого слова должно следовать условие относительно выполнения или не выполнения которого будет решаться какие действия выполнять. А после условия следует ключевое слово Then после которого следуют действия, которые должны выполняться при выполнении условия.

Целиком конструкция выглядит так: if (X>0) then X:=X\*23;

В этом ветвлении проверяется условие положительности переменной Х и если она положительна то мы умножаем эту переменную на 23. Но в этой задаче если переменная Х не положительно то не выполняются не какие действия. Это иногда не удобно. Поэтому существует дополненная конструкция оператор ветвления. Она называется полным ветвлением и выглядит так:

If (x>0) then x:=x\*23 else x:=x\*(-23);

В этой конструкции присутствует слово Else которое означает «Иначе» и все действия, которые происходят после этого слова выполнятся только тогда когда не выполнится условие. То есть в данном случае это произойдет, если x будет меньше или равен нулю.

Часто встречаются задачи, в которых одного условия недостаточно. Тут можно применять вложенные операторы. Вот так:

If (y>0) then if (X<0) then z:=x\*(-y);

Но такая схема не удобна, если очень много условий, да и просто кода получается больше и этот код сложнее читается.

Поэтому необходимо использовать логические операции, которые могут связать несколько условий в одно имеющее два значения – лож или истина.

Первая операция and. Это логическое умножение и оно будет истинно только тогда, когда все входящие в него части будут истины. Пример:

If (x>0) and (z>0) and (y>0) then Writeln (‘Первая четверть координат’);

Сообщение «Первая четверть координат» будет выведено на экран только в том случае, если все три условия будут истины, то ест переменные x, y, z будут положительны.

Следующая операция or. Это логическое сложение и оно будет истинно в том случае если хотя бы одно условие истинно. Пример:

If (x>0) or (z>0) or (y>0) then Writeln ‘Это не отрицательная область’;

Сообщение ‘Это не отрицательная область’ будет выведено на экран, если хотя бы одна переменная положительна.

Следующая операция унарная, то есть работает с одним условием, это not - логическое отрицание. Если условие истинно, то отрицание превращает его в лож и наоборот. Пример:

If not (a=b) then a:=b;

В данном случае присвоение произойдет в том случае если a не равно b.

Бывают ситуации, когда в программе встречаются несколько операторов if подряд и в конце только один else. И необходимо определить к какому if относится else. Если внутри операторов нет операторных скобок begin end, то else относится к последнему if. Если же присутствуют операторные скобки, необходимо тщательно отделить одни операторы от других.

Задания:

1. Дано целое число A. Проверить истинность высказывания: «Число A является положительным».

2. Дано целое число A. Проверить истинность высказывания: «Число A является нечетным».

3. Дано целое число A. Проверить истинность высказывания: «Число A является четным».

Boolean4◦. Даны два целых числа: A, B. Проверить истинность высказывания:

«Справедливы неравенства A > 2 и B ≤ 3».

Boolean5◦. Даны два целых числа: A, B. Проверить истинность высказывания:

«Справедливы неравенства A ≥ 0 или B < −2».

Boolean6◦. Даны три целых числа: A, B, C. Проверить истинность высказывания: «Справедливо двойное неравенство A < B < C».Логические выражения 13

Boolean7◦. Даны три целых числа: A, B, C. Проверить истинность высказывания: «Число B находится между числами A и C».

Boolean8◦. Даны два целых числа: A, B. Проверить истинность высказывания:

«Каждое из чисел A и B нечетное».

Boolean9◦. Даны два целых числа: A, B. Проверить истинность высказывания:

«Хотя бы одно из чисел A и B нечетное».

Boolean10◦. Даны два целых числа: A, B. Проверить истинность высказывания: «Ровно одно из чисел A и B нечетное».

Boolean11◦. Даны два целых числа: A, B. Проверить истинность высказывания: «Числа A и B имеют одинаковую четность».

Boolean12◦. Даны три целых числа: A, B, C. Проверить истинность высказывания: «Каждое из чисел A, B, C положительное».

Boolean13◦. Даны три целых числа: A, B, C. Проверить истинность высказывания: «Хотя бы одно из чисел A, B, C положительное».

Boolean14◦. Даны три целых числа: A, B, C. Проверить истинность высказывания: «Ровно одно из чисел A, B, C положительное».

Boolean15◦. Даны три целых числа: A, B, C. Проверить истинность высказывания: «Ровно два из чисел A, B, C являются положительными».

Boolean16◦. Дано целое положительное число. Проверить истинность высказывания: «Данное число является четным двузначным».

Boolean17◦. Дано целое положительное число. Проверить истинность высказывания: «Данное число является нечетным трехзначным».

Boolean18◦. Проверить истинность высказывания: «Среди трех данных целых

чисел есть хотя бы одна пара совпадающих».

Boolean19◦. Проверить истинность высказывания: «Среди трех данных целых

чисел есть хотя бы одна пара взаимно противоположных».

Boolean20◦. Дано трехзначное число. Проверить истинность высказывания:

«Все цифры данного числа различны».

Boolean21◦. Дано трехзначное число. Проверить истинность высказывания:

«Цифры данного числа образуют возрастающую последовательность».

Boolean22◦. Дано трехзначное число. Проверить истинность высказывания:

«Цифры данного числа образуют возрастающую или убывающую последовательность».

Boolean23◦. Дано четырехзначное число. Проверить истинность высказывания: «Данное число читается одинаково слева направо и справа налево».

Boolean24◦. Даны числа A, B, C (число A не равно 0). Рассмотрев дискриминант D = B2 − 4·A·C, проверить истинность высказывания: «Квадратное

уравнение A·x2 + B·x + C = 0 имеет вещественные корни».

Boolean25◦. Даны числа x, y. Проверить истинность высказывания: «Точка с

координатами (x, y) лежит во второй координатной четверти».

Boolean26◦. Даны числа x, y. Проверить истинность высказывания: «Точка с

координатами (x, y) лежит в четвертой координатной четверти».

Boolean27◦. Даны числа x, y. Проверить истинность высказывания: «Точка с

координатами (x, y) лежит во второй или третьей координатной четверти».

Boolean28◦. Даны числа x, y. Проверить истинность высказывания: «Точка с

координатами (x, y) лежит в первой или третьей координатной четверти».

Boolean29◦. Даны числа x, y, x1, y1, x2, y2. Проверить истинность высказывания: «Точка с координатами (x, y) лежит внутри прямоугольника, левая

верхняя вершина которого имеет координаты (x1, y1), правая нижняя —

(x2, y2), а стороны параллельны координатным осям».

Boolean30◦. Даны целые числа a, b, c, являющиеся сторонами некоторого треугольника. Проверить истинность высказывания: «Треугольник со сторонами a, b, c является равносторонним».

Boolean31◦. Даны целые числа a, b, c, являющиеся сторонами некоторого треугольника. Проверить истинность высказывания: «Треугольник со сторонами a, b, c является равнобедренным».

Boolean32◦. Даны целые числа a, b, c, являющиеся сторонами некоторого треугольника. Проверить истинность высказывания: «Треугольник со сторонами a, b, c является прямоугольным».

Boolean33◦. Даны целые числа a, b, c. Проверить истинность высказывания:

«Существует треугольник со сторонами a, b, c».

Boolean34◦. Даны координаты поля шахматной доски x, y (целые числа, лежащие в диапазоне 1–8). Учитывая, что левое нижнее поле доски (1, 1)

является черным, проверить истинность высказывания: «Данное поле является белым».

Boolean35◦. Даны координаты двух различных полей шахматной доски x1,

y1, x2, y2 (целые числа, лежащие в диапазоне 1–8). Проверить истинность

высказывания: «Данные поля имеют одинаковый цвет».

Boolean36◦. Даны координаты двух различных полей шахматной доски x1, y1,

x2, y2 (целые числа, лежащие в диапазоне 1–8). Проверить истинность высказывания: «Ладья за один ход может перейти с одного поля на другое».

Boolean37◦. Даны координаты двух различных полей шахматной доски x1,Условный оператор 15

y1, x2, y2 (целые числа, лежащие в диапазоне 1–8). Проверить истинность

высказывания: «Король за один ход может перейти с одного поля на другое».

Boolean38◦. Даны координаты двух различных полей шахматной доски x1,

y1, x2, y2 (целые числа, лежащие в диапазоне 1–8). Проверить истинность

высказывания: «Слон за один ход может перейти с одного поля на другое».

Boolean39◦. Даны координаты двух различных полей шахматной доски x1, y1,

x2, y2 (целые числа, лежащие в диапазоне 1–8). Проверить истинность высказывания: «Ферзь за один ход может перейти с одного поля на другое».

Boolean40◦. Даны координаты двух различных полей шахматной доски x1,

y1, x2, y2 (целые числа, лежащие в диапазоне 1–8). Проверить истинность

высказывания: «Конь за один ход может перейти с одного поля на другое».