Лабораторная работа №2

# Ввод и вывод значений простых переменных и одномерных массивов. Создание функциональных тестов

Вариант 16

Выполнил: Татарников Максим группа А-07-22

**1. Постановка задачи (ПЗ)**.

**Задание**: Разработать нисходящим способом алгоритм, отделив ввод и вывод от её решения, и написать программу на Delphi, создав консольное приложение для MS Windows, для решения задачи из нижеприведенного перечня задач.

**Условие**: При заданных А1, А2, …, Аn, В1, В2, …, Вn и С1, С2, …, Сn для каждой из n троек вида (Аi, Вi, Сi) проверить, может ли быть построен треугольник со сторонами Аi, Вi, Сi, при этом подсчитать число треугольников и сумму их периметров.

**2. Уточненная постановка задачи**

Даны три *одномерных вещественных положительных массива* *A, B, C* из *n* (0<*n*≤20) элементов – длины сторон треугольников.

Найти:

*num\_triangle* *–* количество треугольников, которые возможно составить из тройки Ai, Bi, Ci.

(Сумма любых двух сторон должна быть больше третьей)

Если *num\_triangle = n, то вывести “Из всех троек можно составить треугольники!”.*

Если *num\_triangle = 0, то вывести “Ни из одной тройки нельзя составить ни один треугольник!”*

*perimeter –* Сумма всех значений Ai, Bi, Ci, из которых возможно составить треугольник.

Если невозможно вычислить периметр (num\_triangle=0), то вывести «Вычислить периметр невозможно»

**3. Пример**

Возьмем ***n***= 10. И длины разных величин (положительные, отрицательные, нулевые):

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***A*** | 1 | 3 | 8 | 4 | 2 | 7 | 10 | 7 | 6 | 7 |
| ***B*** | 1 | 4 | 5 | 7 | 3 | 8 | 5 | 11 | 2 | 7 |
| ***C*** | 1 | 5 | 3 | 3 | 5 | 9 | 6 | 5 | 9 | 1 |

Из условия только из 4 троек (3,4,5), (1,1,1), (7,8,9), (10,5,6) можно составить треугольник, следовательно:

num\_triangle = 4

perimeter = 3+4+5+1+1+1+7+8+9+10+5+6=60

**4. Таблица данных**

(начало таблицы данных)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Описание  (смысл, диапазон, точность)** | **Тип** | **Структура** | **Формат в/в** |
| Входные данные | ***n*** | число троек длин,  0 < *n* ≤ 20 | цел | простая переменная | XX (:2) |
| ***A*** | Первая сторона треугольника,  0<Ai<=20, точн. 0.1 | вещ | одномерный массив (20) | XX.X (:3:1) |
| ***B*** | Вторая сторона треугольника,  0<Bi<=20, точн. 0.1 | вещ | одномерный массив (20) | XX.X (:3:1) |
| ***C*** | Третья сторона треугольника, 0<Ci<=20, точн. 0.1 | вещ | одномерный массив (20) | XX.X (:3:1) |

(продолжение таблицы данных)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Описание  (смысл, диапазон, точность)** | **Тип** | **Структура** | **Формат в/в** |
| Выходные данные | ***num\_triangle*** | Количество возможных треугольников  0 <= num\_triangle <= 20 | цел | простая переменная | XX (:2) |
| ***perimeter*** | Сумма всех длин троек, которые прошли условие  perimeter>=0 | вещ | простая переменная | XXX.X (:4:1) |
| Промежу-точные  данные\* | ***i*** | индекс текущего элемента,  0 ≤ i ≤ 21 | цел | простая переменная | --- |
|  |  |  |  | --- |
|  |  |  |  | --- |
|  |  |  |  | --- |

**5. Входная форма**:

1.1 Количество троек n:

обр1

1.2 <n>

Обр2.1 Длины сторон треугольника:

<A[1]> <B[1]> <C[1]>

Обр2.2 <A[2]> <B[2]> <C[2]>

..........

<A[n]> <B[n]> <C[n]>

***6.* Выходная форма:**

Обр3 Лаб.2

Обр4 Количество троек n = <n>

Обр5 Длины сторон треугольника:

<A[1]> <B[1]> <C[1]>

Обр6 <A[2]> <B[2]> <C[2]> ......

<A[n]> <B[n]> <C[n]>

Обр7 Количество треугольников =

<num\_triangle>

Обр8 Сумма периметров = <perimeter>

Обр9 Из всех троек можно составить треугольник (num\_triangle = n)

*Обр10 Сумма периметров =* <perimeter>

*Обр11 Ни из одной тройки нельзя составить ни одного треугольника*

*Обр12 Периметр = 0*

**7. Аномалии**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Описание** | **Условие возникновения\*\*** | **Реакция на аномалию** |
| 1 | ***n*** меньше минимально допустимого значения | *n*<1 | Сообщение: «Некорректное *n*: *n*<1» (обр.1.2)  Действие: Завершение работы программы |
| 2 | **Ai**, **Bi**, **Ci** меньше минимально допустимого значения | **Ai** или **Bi** или **Ci** < 0 | Сообщение: «Некорректная длина треугольника: A, B, C *<0*» (обр.2.2)  Действие: Завершение работы программы |

**8. Функциональные тесты**

num\_triangle as n\_t

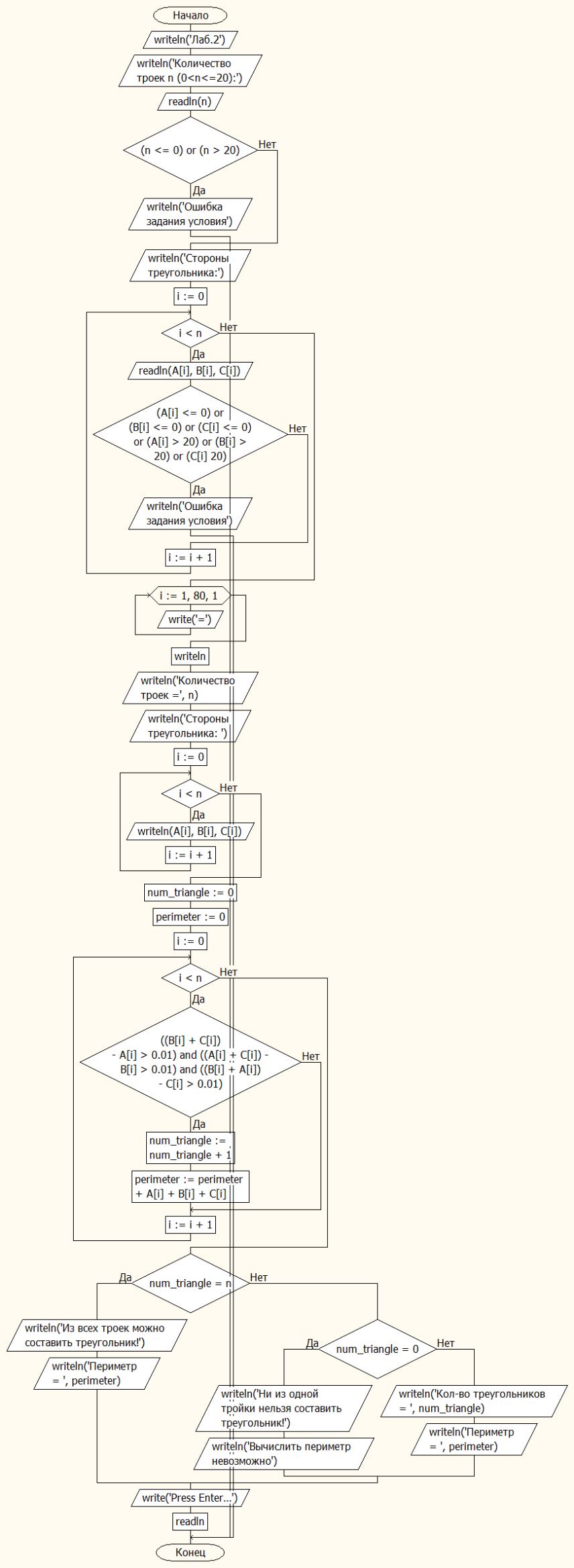
perimeter as p

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Исходные данные** | | | | | | | | | | | | | **Результаты** | | **Тест №** |
|  | *аном* | *граница* | | | *Средние значения* | | | | | *граница* | | *аном* | ***n\_t*** | *макс* = 20 | 2 |
|  |  | | | | | | | | | | | | *мин* = 1 и не ноль | 3 |
| ***n*** | <1 | 1 | | | [2 , 19 ] | | | | | 20 | | >20 |
| Тест№ | --- | | 3 | | 1 | | | | | 2 | --- | | *сред* = (2,19) | 1 |
|  |  | | | | | | | | | | | | *0* | 4 |
| **A[*i*]** | <-20 | -20 | | | (-20,0) | | 0 | (0.1,20) | | 20 | | >20 | *не сущ* = не возможно | --- |
| Тест№ | --- | 2 | | | 3 | 1 | | 1 | | 2 | | --- | *Макс*.*выч*.*нагрузка* =  20 точек из 20 попали | 2 |
| **B[*i*]** | <-20 | -20 | | | (-20,0) | 0 | | (0.1,20) | | 20 | | >20 |
| Тест№ | --- | | | 2 | 3 | 1 | | 1 | 2 | | | --- | **p** | *макс* = 1200 | 2 |
| *мин* = 0.3 | 3 |
| **C[*i*]** | <-20 | -20 | | | (-20,0) | 0 | | (0.1,20) | | 20 | | >20 |
| Тест№ | --- | 2 | | | 3 | 1 | | 1 | | 2 | | --- | *сред* = (0.3, 1200) | 1 |
|  |  |  | | |  |  | |  | |  | |  | *0* | … |
|  |  |  | | |  |  | |  | |  | |  | *не сущ, когда n\_t=0* | 4 |
|  | *Аном3* | *аном3* | | | *Аном3* | Аном2 | | *сред* | | *граница* | | *Аном1* | *Макс*.*выч*.*нагрузка* = 1200 | 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| **№ теста** | **Входные данные** | **Ожидаемый  результат** | **Смысл теста** |
| 1 | *n*=10   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | *A* | 1 | 3 | 8 | 4 | 2 | 7 | 10 | 7 | 6 | 7 | | *B* | 1 | 4 | 5 | 7 | 3 | 8 | 5 | 11 | 2 | 7 | | *C* | 1 | 5 | 3 | 3 | 5 | 9 | 6 | 5 | 9 | 1 | | N\_t = 7  *P = 115* | Нормально-средние значение для общей проверки программы |
| 2 | *n* =20   |  |  |  |  |  | | --- | --- | --- | --- | --- | | *A* | 20 | 20 | … | 20 | | *B* | 20 | 20 | … | 20 | | *C* | 20 | 20 | … | 20 | | *N\_t = 20*  P = 1200 | **Максимальное** значение всех исходных данных и результатов.  **Максимальная вычислительная нагрузка** для n\_t и p. |
| 3 | *n*=1   |  |  | | --- | --- | | *A* | 0.1 | | *B* | 0.1 | | *C* | 0.1 | | *N\_t = 1*  P = 0,3 | **Минимальные** значения n.  **Минимальные** значения A,B,C,n\_t,p |
| 4 | P – не сущ  Num\_triangle = 0 | Сообщение по обр.А1 | Аномалия №1  … |

|  |  |  |
| --- | --- | --- |
| **Результаты** | | **Тест №** |
| N\_t | *макс* =20 | 2 |
| ***мин* и не 0:** 1 | 3 |
| *сред* = (0,20) | 1 |
| *не сущ* = невозможно | --- |
| ***0*** | **4** |
| *Макс.выч.нагрузка* = 20 | 2 |
| p | *макс* = 1200 | 2 |
| ***мин*  и не 0:**  *n* =20   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | *A* | 0,1 | 0 | … | 0 | 0 | | *B* | 0.1 | 0 | … | 0 | 0 | | *C* | 0.1 | 0 | … | 0 | 0 |   P = 0.1+0.1+0.1 = 0.3 | 3 |
| *сред* = (0.3, 1200) | 1 |
| *не сущ, когда num\_triangle = 0* | 4 |
| ***0, не сущ*** | … |
| *Макс*.*выч*.*нагрузка* = 1200 | 2 |

***9. Блок-схема***

******

***10 Программа***

**program** Lab2;

**const**

Nmax = 20;{верхняя граница индексов массива – максимальное количество точек}

{раздел описания переменных:}

**var**

n, i, num\_triangle: integer;

perimeter: real;

A, B, C: **array** [0..Nmax] **of** real;

{раздел операторов:}

**begin**

//вывод заголовка в выходной документ:

writeln('Лаб.2':40); {вывод c переходом на следующую строку}

{ввод исходных данных: }

writeln('Количество троек n (0<n<=20):'); readln(n); {обр 1.1,1.2}

{ввод массивов A, B, C:}

**if** (n <= 0) **or** (n > 20) **then**

**begin**

writeln('Ошибка задания условия');

**exit**;

**end**;

writeln('Стороны треугольника:'); {обр2.1}

i := 0;

**while** i<n **do**

**begin**

readln(A[i], B[i], C[i]);

**if** (A[i]<=0) **or** (B[i]<=0) **or** (C[i]<=0) **or** (A[i]>20) **or** (B[i]>20) **or** (C[i]>20) **then**

**begin**

writeln('Ошибка задания условия');

**exit**;

**end**;

{обр2.2:ввод 3 элементов через пробел и переход на след. строку}

i := i+1;

**end**;

{вывод входных данных в выходной документ для подтверждения: }

**for** i := 1 **to** 80 **do** write('='); { отделим визуально чертой и строкой введенные и выводимые значения }

writeln;

writeln('Количество троек =', n:3); {обр4}

{вывод исходных массивов A, B, C}

writeln('Стороны треугольника: ');

i := 0;

**while** i < n **do** {обр4,5}

**begin**

writeln(' ':5, A[i]:3:1, ' ':8, B[i]:3:1, ' ':8, C[i]:3:1);

i := i + 1;

**end**;

num\_triangle := 0;

perimeter := 0;

i := 0;

**while** i < n **do** // перебираем по-очереди все n точек

**begin**

**if** ((B[i] + C[i]) - A[i] > 0.01) **and** ((A[i] + C[i]) - B[i] > 0.01) **and** ((B[i] + A[i]) - C[i] > 0.01) **then** // Проверка усл на неравенство треугольника

**begin**

num\_triangle := num\_triangle + 1; // и количество увеличиваем

perimeter := perimeter + A[i] + B[i] + C[i];

**end**;

i := i + 1;

**end**;

**if** num\_triangle = n **then**

**begin**

writeln('Из всех троек можно составить треугольник!':45);

writeln(' Периметр = ', perimeter:3);

**end**

**else if** num\_triangle = 0 **then**

**begin**

writeln('Ни из одной тройки нельзя составить треугольник!');

writeln('Вычислить периметр невозможно');

**end**

**else**

**begin**

writeln(' Кол-во треугольников = ', num\_triangle:2); {обр.7}

writeln(' Периметр = ', perimeter:3); {обр.8}

**end**;

write('Press Enter…'); readln; {задерживаем экран до нажатия ENTER}

**end**.