МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ШАХТИНСКИЙ ИНСТИТУТ (ФИЛИАЛ)

ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ЮЖНО-РОССАИЙСКИЙ ГОСУДАРСТВЕННЫЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

(НПИ) им.М.И. ПЛАТОВА»

ФОЗДО

(заочная форма обучение)

**Контрольная работа №** 1

по программированию на языках высокого уровня

(наименование дисциплины)

студент 1-го курса, шифр\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_Толстунов Александр Николаевич\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

Почтовый индекс и домашний адрес: 344020

г. Ростов-на-Дону, ул. Архангельская 7, а

2015 год

# Вариант 6

Задание 1

Треугольник задан координатами вершин. Составить программу для нахождения расстояний от его центра тяжести до вершин.

Решение.

Решение приведено в листинге 6.2.

**Листинг 6.1 Решение задания 1 варианта 6**

**program** var6\_ex1;

**var**

x1, y1, x2, y2, x3, y3: integer; //координаты вершин треугольника

ab, bc, ca: real; //длины сторон треугольника

xm, ym: real; //координаты центра тяжести

am, bm, cm: real; //расстояния от центра тяжести до вершин

**begin**

{ ввод данных }

write('Введите координаты точки A: ');

readln(x1, y1);

write('Введите координаты точки B: ');

readln(x2, y2);

write('Введите координаты точки C: ');

readln(x3, y3);

{ расчет длин сторон треугольника }

ab := sqrt(sqr(x2 - x1) + sqr(y2 - y1));

bc := sqrt(sqr(x3 - x2) + sqr(y3 – y2));

ca := sqrt(sqr(x1 - x3) + sqr(y1 - y3));

{ вывод данных }

writeln('Координаты точки A=[', x1, ', ', y1, ']');

writeln('Координаты точки B=[', x2, ', ', y2, ']');

writeln('Координаты точки C=[', x3, ', ', y3, ']');

writeln('Длина AB = ', ab:0:2);

writeln('Длина BC = ', bc:0:2);

writeln('Длина CA = ', ca:0:2);

{ расчет координат центра тяжести }

xm := (x1 + x2 + x3) / 3; //Кордината x центра тяжести

ym := (y1 + y2 + y3) / 3; //Кордината y центра тяжести

writeln('Координаты центра тяжести M=[', xm:0:2, ', ', ym:0:2, ']'); //вывод данных

am:=sqrt(sqr(xm - x1) + sqr(ym - y1)); //Расстояние до вершины А

bm:=sqrt(sqr(xm - x2) + sqr(ym - y2)); //Расстояние до вершины B

cm:=sqrt(sqr(xm - x3) + sqr(ym - y3)); //Расстояние до вершины C

{ вывод данных }

writeln('Расстояние от точки M до вершины А = ', am:0:2);

writeln('Расстояние от точки M до вершины B = ', bm:0:2);

writeln('Расстояние от точки M до вершины C = ', cm:0:2);

**end**.

# Вариант 6

Задание 2

Составить программу для решения уравнения .

Решение

Решение приведено в листинге 6.2.

**Листинг 6.2 Решение задания 2 варианта 6**

**program** var6\_ex2;

**var**

a, b, c, d, e, x1, x2, disc: real;

**begin**

writeln('Программа решает уравнения тип: (x-a)(x-b)=c.');

write('Введите коэффициент a: ');

readln(a);

write('Введите коэффициент b: ');

readln(b);

write('Введите коэффициент c: ');

readln(c);

**if**(c = 0) **then** //при c = 0

**begin**

x1:=a; x2:=b;

writeln('Корни уравнений: x1=', x1, ', x2=', x2);

**exit**; //выходим из программы

**end**

**else**

**begin**

{при раскрытии скобок получаем x^2 - bx - ax + ab - c = 0 }

e:=-b-a; //bx-ax

d:=a\*b-c; //ab - c

{ получаем x^2 - ex + d }

disc:= sqr(e) - (4 \* d);

**if** (disc < 0) **then** //если дискриминант меньше нуля

**begin**

writeln('Действительных корней нет.');

**exit**; //выходим из программы

**end**

**else**

**begin**

x1:= (-e - sqrt(disc)) / 2;

x2:= (-e + sqrt(disc)) / 2;

**end**;

**end**;

writeln('Корни уравнений: x1=', x1, ', x2=', x2); //вывод результатов

**end**.

# Вариант 6

Задание 3

Последовательность задана формулой общего члена:

.

Определить (из первых двадцати) члены последовательности, являю-щиеся нечетными числами.

Решение

Решение приведено в листинге 6.3.

**Листинг 6.3 Решение задания 3 варианта 6**

**program** var6\_ex3;

**var**

x, n: integer;

**begin**

**for** n:=1 **to** 20 **do**

**begin**

x:=(5 \* sqr(n)) - (4 \* n) - 1;

**if** (x **mod** 2 <> 0) **then**

writeln('В числовой последовательности член #', n,

' является нечетным и равен ', x);

**end**;

**end**.

Вывод программы будет следующим:

В числовой последовательности член #2 является нечетным и равен 11

В числовой последовательности член #4 является нечетным и равен 63

В числовой последовательности член #6 является нечетным и равен 155

В числовой последовательности член #8 является нечетным и равен 287

В числовой последовательности член #10 является нечетным и равен 459

В числовой последовательности член #12 является нечетным и равен 671

В числовой последовательности член #14 является нечетным и равен 923

В числовой последовательности член #16 является нечетным и равен 1215

В числовой последовательности член #18 является нечетным и равен 1547

В числовой последовательности член #20 является нечетным и равен 1919

# Вариант 6

Задание 4

Дан массив целых чисел А(12). Переписать в массив Х четные, а в массив Y нечетные элементы массива А.

Решение

Для решения данной задачи есть несколько вариантов. В одном из вариантов можно заполнить массив А непосредственно в программе, а в другом, который продемонстрирован ниже, массив заполняет пользователь, затем программа вычисляет четные и нечетные элементы и вводит их в соответствующие четности массивы. Решение представлено в листинге 6.4:

**Листинг 6.4 Решение задания 4 варианта 6**

**program** var6\_ex4;

**var**

A: **array**[1..12] **of** integer; { основной массив }

X: **array**[1..12] **of** integer; { массив четных чисел }

Y: **array**[1..12] **of** integer; { массив нечетных чисел }

i, j, k, z, num: integer; { счетчики для массивов }

**begin**

**for** i:=1 **to** 12 **do** { заполнение массива пользователем }

**begin**

write('Введите элемент массива А[', i, ']=');

readln(A[i]);

**end**;

k:=1; z:=1; { определение счетчиков }

{ вычисление четности и заполнение соответствующих массивов }

**for** j:=1 **to** 12 **do**

**begin**

num:= A[j];

**if** num **mod** 2 = 0 **then**

**begin**

X[k]:=num;

k:=k+1;

**end**

**else**

**begin**

Y[z]:=num;

z:=z+1;

**end**

**end**;

{вывод на экран элементов четного массива}

writeln('Массив из четных чисел:');

**for** i:=1 **to** k-1 **do**

write('A[', i, ']=', X[i], ', ');

writeln();

{вывод на экран элементов нечетного массива}

writeln('Массив из нечетных чисел:');

**for** i:=1 **to** z-1 **do**

write('A[', i, ']=', Y[i], ', ');

writeln();

**end**.

# Вариант 6

Задание 5

В массиве Х(5,4) найти наименьший из положительных элементов.

Решение

Решение приведено в листинге 6.4. Для данной программы применен randomize, который заполняет массив A случайными числами от -99 до 99 при запуске программы.

**Листинг 6.4 Решение задания 5 варианта 6**

**program** var6\_ex5;

**const**

n=5; { количество строк }

m=4; { количество столбцов }

**var**

A: **array**[1..n, 1..m] **of** integer;

i, j, min, ai, aj: integer;

contin: boolean;

**begin**

randomize; { заполнение массива случайными числами от -99 до 99 }

**for** i:=1 **to** n **do**

**for** j:=1 **to** m **do**

A[i,j]:=random(100)-random(100);

**for** i:=1 **to** n **do**  { Вывод на экран всех элементов массива }

**begin**

**for** j:=1 **to** m **do**

write('A[', i, ',', j, ']=', A[i,j], ', ');

writeln();

**end**;

contin:=true;

**for** i:=1 **to** n **do** { Нахождение первого положительного числа }

**begin**

**if** (contin = false) **then break**;

**for** j:=1 **to** m **do**

**if** (A[i,j] > 0) **then** //Если положительное число найдено,

**begin** //запоминаем его и прерываем проверку

ai:=i;

aj:=j;

min:=A[i,j];

contin:=false;

**break**;

**end**

**end**;

{ Нахождение минимального положительного элемента массива }

**for** i:=1 **to** n **do**

**for** j:=1 **to** m **do**

**if** (A[i,j] > 0) **and** (A[i,j] < min) **then**

**begin**

ai:=i;

aj:=j;

min:=A[i,j];

**end**;

{вывод результатов}

**if** (ai=0) **and** (aj=0) **and** (min=0) **then**

writeln('Положительные числа в массиве отсутствуют')

**else**

writeln('Минимальный положительный элемент массива: A[',

ai, ',', aj, ']=', min);

**end**.