Л А Б О Р А Т О Р Н А Я Р А Б О Т А N6

Комплексная геометрическая задача с использованием графических средств для представления результата

ЦЕЛЬ РАБОТЫ: научиться применять знания аналитической геометрии для решения практических задач машинной графики,
осуществлять построение изображения (в СКУ) объектов, расположенных в МСК.

Решение комплексной задачи с геометрическим содержанием требует от студента использования в совокупности сведений, полученных при изучении аналитической геометрии, знаний и навыков, приобретенных в ходе изучения дисциплины "Вычислительная
техника и информационная технология".

При решении подобной задачи студент должен выбрать математический метод решения задачи (выбрать как собственно путь (последовательность) решения, так и наиболее рациональные математические соотношения для вычисления требуемых характеристик геометрических объектов), разработать алгоритм решения задачи, модульную структуру комплекса программ, алгоритмы отдельных модулей, провести отладку отдельных модулей и комплексную отладку всего программного продукта.

При этом необходимо разработать структуру используемых данных и определить их типы, определить множество допустимых и недопустимых исходных данных и предусмотреть контроль вводимых данных.

Для работы с программой ввода исходных данных, ответа на запросы программы желательно предусмотреть меню.

Во второй части задания требуется в графическом режиме вывести на экран изображение полученных геометрических объектов (сделать иллюстрацию (рисунок) полученного решения задачи).

Это позволяет проверить практические навыки студентов при работе с графическим модулем, а также их умение выполнять масштабирование, учитывать особенности вывода изображения на экран (например, разные разрешающие способности экрана).

В качестве примера рассмотрим решение следующей задачи:

На плоскости задано два множества точек. Найти такую окружность (окружность должна проходить хотя бы через три различные точки первого множества) и такой треугольник (вершины треугольника должны располагаться в точках второго множества), что прямая, соединяющая центр окружности и точку пересечения высот треугольника, образует минимальный угол с осью ординат.

Предполагается, что количество точек в каждом множестве ограничено и задача может быть решена простым перебором, то есть рассматриваются все возможные пары окружностей и треугольников, для которых определяется угол, образованный двумя прямыми: первая соединяет центр окружности с точкой пересечения высот треугольника, вторая - ось ординат. Из всех рассматриваемых пар выбирается та, для которой искомый угол будет иметь минимальное значение.

Таким образом, при решении стоящей задачи необходимо выполнить следующие операции:

1) определить координаты центра окружности, проходящей через три заданные точки; при этом попутно находится ответ на вопрос о возможности построения окружности, проходящей через три заданные точки;
2) найти координаты точки пересечения высот треугольника, при этом необходимо выполнить следующие действия: - проверить существование треугольника с вершинами в трех заданных точках;
   - найти уравнение прямой, являющейся высотой в треугольнике;
   - найти точку пересечения двух прямых;
3) найти уравнение прямой, проходящей через две заданные точки (центр окружности и точка пересечения высот треугольника);
4) найти угол между двумя прямыми.

Рассмотрим решение каждой из выделенных задач.

Определить координаты центра окружности по известным координатам трех точек можно несколькими способами.

Один из способов основан на использовании уравнения окружности

$$(X-Xo)S + (Y-Yo)S = RS \qquad (1)$$

где X,Y - координаты точки, принадлежащей окружности;
Xo,Yo - координаты центра окружности;
R - радиус окружности.

Записав это соотношение для трех точек (X1,Y1,X2,Y2,X3,Y3 -координаты этих точек), лежащих на окружности, получим систему трех уравнений с тремя неизвестными, решив которую найдем координаты центра (Xo,Yo) и радиус окружности R:

$$(X1-Xo)S + (Y1-Yo)S = RS$$

(X2-Xo)S + (Y2-Yo)S = RS (2)

(X3-Xo)S + (Y3-Yo)S = RS

Перед определением координат центра окружности, проходящей через очередные три точки, целесообразно убедиться в том, что такая окружность может быть построена. Для этого рассматриваемые три точки не должны лежать на одной прямой. Три точки не лежат на одной прямой, если выполняется соотношение:

(Y2-Y1)*(X3-X1)<>(Y3-Y1)*(X2-X1) (3)

Если неравенство (3) справедливо, то окружность существует и можно найти координаты ее центра и величину радиуса из (2). Для решения этой системы можно поступить следующим образом: приравнять друг другу левые части первых двух уравнений

(X1-Xo)S + (Y1-Yo)S = (X2-Xo)S + (Y2-Yo)S

и получить соотношение:

A*Xo+B*Yo=C где A=2*(X2-X1)

B=2*(Y2-Y1)

C=X2S-X1S+Y2S-Y1S

Затем аналогично приравняем левые части второго и третьего уравнений и получим:

D*Xo+E*Yo=F где D=2*(X3-X2)

E=2*(Y3-Y2)

F=X3S-X2S+Y3S-Y2S

В итоге получаем простейшую систему двух уравнений с двумя неизвестными:

A*Xo+B*Yo=C

D*Xo+E*Yo=F (4)

где коэффициенты A,B,C,D,E,F вычисляются по приведенным формулам через известные координаты трех точек.

Если A<>0, то выражаем Xo из первого уравнения Xo=(C-B*Yo) /A и подставляем его значение во второе уравнение, откуда получаем

Yo*(E-B*D/A)=F-DC/A

Если E-B*D/A<>0, то Yo=(F-D*C/A)/(E-B*D/A)

Поскольку A<>0, то должно выполняться соотношение A*E<>B*D, т.е. (X2-X1)*(Y3-Y2)<>(Y2-Y1)*(X3-X2).

Это соотношение выполняется, так как три точки не лежат на одной прямой. Если же A=0, то Yo=C/B. Здесь B<>0, так как если одновременно $A=0$ и $B=0$, то точки 1 и 2 фактически являются одной точкой (в нашем случае тест дал бы результат, что все три точки лежат на одной прямой). Из второго уравнения получим для $Xo=(F-E*C/B)/D$.

Можно решить уравнение и классическим путем (через определители):

$$Xo=\frac{\begin{vmatrix} C & B \\ F & E \end{vmatrix}}{\begin{vmatrix} A & B \\ D & E \end{vmatrix}}=\frac{C*E-B*F}{A*E-D*B}$$

$$Yo=\frac{\begin{vmatrix} A & C \\ D & F \end{vmatrix}}{\begin{vmatrix} A & B \\ D & E \end{vmatrix}}=\frac{A*F-D*C}{A*E-D*B} \qquad (5)$$

Это дает тот же результат.

В заключение из любого уравнения системы (2) определяется величина радиуса окружности.

Другой способ определения координат центра окружности может заключаться в следующем. Известно, что центр окружности лежит на пересечении перпендикуляров, проведенных через середины отрезков.

Координаты середины отрезка определяется из сотношений: $Xc=(X1+X2)/2$

$$Yc=(Y1+Y2)/2 \qquad (6)$$

где $X1,X2,Y1,Y2$ - координаты концов отрезка;

Xc,Yc - координаты центра отрезка.

Чтобы найти уравнение перпендикуляра, проведенного к отрезку, надо знать уравнение прямой, проходящей через две точки, являющиеся концами отрезка.

Уравнение прямой имеет вид

$$A*X+B*Y+C=0 \qquad (7)$$

где $A=Y1-Y2$

$B=X2-X1$

$C=(X1-X2)*Y1+(Y2-Y1)*X1$

(Следует подчеркнуть, что уравнение (7) описывает любую прямую на плоскости. Например, уравнение прямой Y=k*X+b не описывает вертикальную прямую (прямая имеет параметрическое число три. Поэтому следует использовать всегда уравнение (7)).

Уравнение прямой, перпендикулярной рассматриваемому отрезку и проходящей через точку (Xc,Yc), имеет вид

$$-B*X+A*Y+D=0 \qquad (8)$$

где D=B*Xc-A*Yc

Таким образом, выбрав второй способ определения координат центра окружности, надо также прежде всего убедиться в том, что точки не лежат на одной прямой. Затем определить координаты центров отрезков, соединяющих, например, точки 1 и 2, 2 и 3. Далее определить из (8) коэффициенты перпендикуляров, проведенных к отрезкам через их середины.

Затем найти координаты точки пересечения двух перпендикуляров, для чего решить систему двух уравнений типа (8) с двумя неизвестными. Найденная точка является центром окружности.

Радиус окружности определяется как расстояние от центра до любой из искомых точек:

$$R=» \overline{(Xo-X1)S+ (Yo-Y1)S} \qquad (9)$$

Рассмотрим решение второго пункта поставленной задачи. Для проверки существования треугольника следует воспользоваться соотношением (3). Если три точки не лежат на одной прямой, то есть неравенство (3) выполняется, то треугольник существует.

Для нахождения уравнения высоты треугольника надо 1)определить коэффициенты уравнения стороны треугольника, к которой перпендикулярна высота. Для этого используется соотношение (7). 2)Определить коэффициенты уравнения высоты по известным коэффициентам уравнения стороны треугольника и известным координатам вершины треугольника. Для этого используется соотношение (8).

Точка пересечения двух высот находится при совместном решении системы двух уравнений с двумя неизвестными, каждое из которых представляет уравнение высоты. Координаты точки пересечения находятся из соотношений:

$$Xp=\frac{B1*C2-B2*C1}{A1*B2-A2*B1}; \qquad Yp=\frac{C1*A2-C2*A1}{A1*B2-A2*B1}$$

где A1,A2,B1,B2,C1,C2 - коэффициенты уравнений прямых, имеющих вид

A1*X+B1*Y+C1=0 и A2*X+B2*Y+C2=0.

Третий пункт решения задачи выполняется на основе соотношения (7), то есть в результате получаем коэффициенты уравнения прямой, соединяющей центр окружности и точку пересечения высот треугольника.

Четвертый шаг задачи, заключающийся в нахождении угла между двумя прямыми, основан на использовании соотношения (12):

$$tgf=\frac{A1*B2-A2*B1}{A1*A2-B1*B2} \qquad (12)$$

где A1, B1, C1 - коэффициенты уравнения первой прямой;

A2, B2, C2 - коэффициенты уравнения второй прямой.

В данном случае уравнение второй прямой имеет вид X=0, то есть A2=1, B2=C2=0. Коэффициенты A2,B2,C2 можно определить с помощью процедуры, задав любые две точки, принадлежащие оси ординат, например, (0,0) и (0,1).

Из (12) $f=Arctg\left(\frac{A1*B2-A2*B1}{A1*A2-B1*B2}\right)$.

При вычислении (12) следует иметь в виду, что знаменатель может быть равен нулю, поэтому в проограмме перед делением следует произвести его анализ, сравнив с заранее заданным маленьким числом. Равенство нулю означает угол в 90 градусов.

В соответствии с алгоритмом и рассмотренными математическими методами, используемыми при решении задачи, можно разработать следующую модульную структуру комплекса:

MAIN

Управляющая программа

|

-- -

| | | | | | |

```
|      OKR       |  |  |      KOEFPR      |  |  |      KOEFPERP      |  |  | PERESECH         |
| Определение | | | Определение  | | |   Определение   | | | Определение |
| параметров  | | | коэффициентов| | |  коэффициентов  | | |  координат  |
| окружности  | | |   уравнения  | | |                   уравнения          | | |           точки          |
--------------- | |                          прямой         | | | перпендикулярной| | | пересечения |
                    | ---------------- | |                              прямой            | | |            двух           |
                    |                          | ------------------- | |   прямых                                  |
                    |                          |                                      | ---------------
                    |                          |                                      |
        ----------------  ------------------                       ------------------
        |          UGOL        | |             TRIT              |           |          MASST          |
        ----------------           ------------------              ------------------
        | Определение  |           |  Определение  |              |  Определение  |
        |  угла между  |           | факта располо- |              |  координат про-|
        | двумя прямыми|           | жения трех то- |              |  масштабирован-|
        ----------------           |  чек на одной  |              |   ной  точки   |
                                 |        прямой          |         ------------------
                                 ------------------
```

Функции управляющей программы будут состоять в следующем:

1) ввод и анализ исходных данных;
2) выбор очередных трех точек первого множества, которые могут принадлежать окружности, и обращение к процедурам определения параметров окружности;
3) выбор очередных трех точек второго множества, которые могут являться вершинами треугольника, и обращение к процедурам, вычисляющим угол между прямыми;
4) сравнение полученного угла с текущим значением и запоминание минимального угла и номеров точек, образующих искомые окружность и треугольник;
5) выдача результатов;
6) выдача графического изображения результата.

Исходными данными в рассматриваемой программе являются:

1)количество точек первого множества и координаты точек (X,Y) этого множества;

2)количество точек второго множества и координаты точек этого множества. Количество точек - переменная целого типа.

После ввода количества точек следует проверить кооректность введеных данных. Здесь можно выделить следующие варианты:

1) N<=0;
2) 0<N<3;
3) 3<=N<=NM;
4) N>NM,

где N - количество введенных точек,
NM - размерность массивов для хранения координат точек. Поскольку количество - натуральное число, то отрицательное или нулевое значение некорректно. Если количество введенных точек меньше трех, то нельзя построить окружность (во втором случае треугольник). Третий случай является корректным. В четвертом случае при последующем вводе координат точек не хватит памяти, отведенной для соответствующих массивов.

Программа в случае ввода некорректных исходных данных заправшивает ответ пользователя на повторный ввод данных или прекращение работы.

Для размещения координат точек можно выбрать тип данных массив, причем можно выделить одномерные массивы для хранения по-отдельности X- и Y-координат точек, а можно выделить двумерный массив, в котором первый столбец отводится для хранения X-координат, а второй - Y-координат. Следует отметить, что координаты должны быть величинами действительного типа. Студенты часто рассматривают только целочисленные координаты (причем расположенные в заранее оговоренном диапазоне), объясняя это последующим отображением в графическом режиме, при котором координаты пикселов задаются неотрицательными целыми значениями. Однако такой подход является неверным.

Поскольку выделение памяти под массивы производится в языке Паскаль статическим способом, то заранее требуется объявить размерность массивов координат точек. Поскольку рассматриваемые задания носят учебный характер (при решении реальных задач и размерность будет определяться из реальных потребностей), то задавать большое количество точек нет смысла, так как в этом случае резко увеличивается количество переборов и, соответственно, время вычислений.

Например, если задать количество точек каждого множества равным 5, то может существовать C =5!/(3!*2!)=10 различных окружностей и столько же различных треугольников. А поскольку должны быть рассмотрены варианты сочетания каждой окружности с каждым треугольником, то всего получается 100 комбинаций.

Чтобы не рассматривать повторяющиеся комбинации точек, следует поступить следующим образом: индекс первой точки (I) изменять от 1 до N-2, индекс второй точки изменять от I+1 до N -1, индекс третьей точки от I+2 до N. Перебирая во вложенных циклах все варианты, студенты часто изменяют все три индекса от 1 до N, что ведет к лишним затратам времени: в этом случае либо встречаются повторяющиеся варианты геометрических объектов, либо студенты вставляют анализ индексов на совпадение, что также требует дополнительного расхода времени.

Наконец, рассмотрим заключительный этап - построение изображения в графическом режиме на экране дисплея. Прежде всего необходимо осуществить масштабирование, то есть перейти от мировой системы координат (в которой задавались координаты точек) к системе координат устройства (они имеют неотрицательные целые значения, ограниченные максимальными значениями, зависящими от монитора и режима работы графического адаптера).

Для проведения масштабирования надо определить минимальные и максимальные значения координат (МСК). При этом надо иметь в виду, что эти минимальные и максимальные координаты не совпадают с соответствующими значениями координат введенных точек. Для треугольника минимальное и максимальное значения можно определить, рассматривая координаты введенных точек, а для окружности следует знать абсциссы концов горизонтального диаметра, для ординат - ординаты вертикального диаметра, то есть рассматривать величины Xc-R, Xc+R, Yc-R, Yc+R. Если же изображать и точку пересечения высот в треугольнике, то также следует учесть, что эта точка в тупоугольном треугольнике лежит за его пределами.

Для вычисления коэффициентов масштабирования Kx,Ky следует использовать следующие соотношения:

$$Kx=\frac{KXmax-Kxmin}{Xmax-Xmin} \qquad Ky=\frac{KYmax-KYmin}{Ymax-Ymin}$$

где KXmax, KYmax - координаты правого нижнего угла поля вывода;

KXmin, KYmin - координаты левого верхнего угла поля вывода;

Xmin, Xmax - минимальное и максимальное значения абсциссы изображаемых точек в МСК;

Ymin, Ymax - минимальное и максимальное значения ординаты изображаемых точек в МСК.

Коэффициенты масштабирования в общем случае будут иметь неодинаковые значения (неоднородное масштабирование), что приведет при их использовании к искажению пропорций геометрических объектов. Для исключения искажений пропорций следует взять один единый коэффициент масштабирования для обеих осей - минимальное из двух значений Kx, Ky. В этом случае изменение размеров изображения вдоль координатных осей будет одинаковым, правда, не все пространство экрана будет использовано в этом случае. С учетом направления координатных осей в графической системе координаты точек на экране рассчитываются из соотношений:

XKi=E(Xn+(Xi-Xmin)*Kx)

YKi=E(Yn+(Ymax-Yi)*Ky)

где Xn - абсцисса начальной точки поля вывода;

Yn - ордината начальной точки поля вывода;

E - операция округления до ближайшего целого;

XKi, YKi - координаты i-ой точки на экране;

Xi, Yi - координаты i-ой точки в МСК.

ТЕКСТ ПРОГРАММЫ ГЕОМЕТРИЧЕСКОЙ ЗАДАЧИ

```
program lab_61;
Найти окружность (построенную на трех точках заданного множества)
и треугольник (построенный на трех точках другого множества)
такие, что прямая, соединяющая центр окружности и точку пересечения
высот треугольника образовывала минимальный угол с осью ординат
   Uses Crt,Graph; label 1,2,3,4,5;
   const ko=10; размерность массива для хранения координат точек первого множества
                  kt=10; размерность массива для хранения координат точек
                             второго множества
                  delx=10; расстояние по оси абсцисс между краем экрана и
```

```
                              полем вывода изображения
                           dely=10; расстояние по оси ординат между краем экрана и
                              полем вывода изображения
type mas=array[1..ko]  of real;
     mas1=array[1..3]  of real;
     mask=array[1..3] of integer;
var x1,y1,   массивы координат точек первого множества
    x2,y2:   массивы координат трочек второго множества
         mas;
           kxr,kyr:  массивы экранных координат вершин результирующего
                                 треугольника
           mask;
    i,k,l,j,i1,j1,  параметры циклов
    kod,         признак распололжения трех точек на одной прямой
    k1,          фактическое количество точек первого множества
    k2,          фактическое количество точек второго множества
    xn,yn,       координаты верхнего левого угла поля вывода
    xk,yk,       координаты нижнего правого угла поля вывода
    kr,          величина радиуса окружности в СКУ
    kx0r,ky0r, координаты центра окружности в СКУ
    kol,   общее количество просмотренных вариантов
    i1r,j1r,k1r, номера трех точек первого множества, на которых
                             построена результирующая окружность
    i2r,j2r,k2r, номера трех точек второго множества, на которых
                             построен результирующий треугольник
    gd,gm,                    тип и режим работы адаптера
    kx1,ky1,kx2,ky2,kx3,ky3, координаты трех точек результирующей
                                           окружности в СКУ
    kxpr,kypr:координаты точки пересечения высот треугольника в СКУ
                   integer;
    ch:char; признак повторения ввода данных при некорректных данных
    xr,yr:mas1;массивы координат вершин результирующего треугольника
    xa,ya:word;коэффициенты учета разных разрешающих способностей
    ugolmin,   величина искомого угла
```

r, радиус текущей окружности
p, знаменатель в выражении для определения координат точки пересечения двух прямых
x0,y0, координаты центра текущей окружности в МСК
a,b,c, коэффициенты уравнения прямой
x0r,y0r, координаты центра результирующей окружности
a1,b1,c1, коэффициенты уравнения одной стороны треугольника
a2,b2,c2, коэффициенты уравнения второй стороны треугольника
xp,yp, координаты точки пересечения высот треугольника
rr, радиус результирующей окружности
xmin,xmax, минимальные и максимальные значения координат
ymin,ymax, точек изображаемых объектов
xpr,ypr, координаты точки пересечения высот в результирующем треугольнике
x1r,x2r,x3r,координаты трех точек первого множества, через которые
y1r,y2r,y3r, проходит результирующая окружность
x12r,x22r,x32r, координаты трех точек второго множества, которые
y12r,y22r,y32r, являются вершинами результирующего треугольника
fi, текущее значение угла образованного прямой проходящей через центр окружности и точку пересечения высот и осью ординат
km, итоговый коэффициент масштабирования изображения
kx,ky:real; коэффициенты масштабирования вдоль осей

Процедура определения координат промасштабированной точки

```
procedure masst(x,y,xmin,ymax,k:real;xn,yn:integer;var kx,ky:integer);
begin
        kx:=round(xn+(x-xmin)*k);
        ky:=round(yn+(ymax-y)*k);
end;
```

Процедура определения принадлежности трех точек одной прямой procedure

```
trit(x1,x2,x3,y1,y2,y3:real;var kod:integer); begin
                if (y2-y1)*(x3-x1)<>(y3-y1)*(x2-x1) then kod:=0
         else kod:=-1;
```

```
end;
```

Процедура вычисления параметров окружности по трем точкам procedure okr(x1,x2,x3,y1,y2,y3:real; var x0,y0,r:real); var a,b,c,d,e,f,p:real;

```
begin
        a:=2*(x2-x1);
        b:=2*(y2-y1);
        c:=x2*x2-x1*x1+y2*y2-y1*y1;
        d:=2*(x3-x2);
        e:=2*(y3-y2);
        f:=x3*x3-x2*x2+y3*y3-y2*y2;
        p:=a*e-d*b;
        x0:=(c*e-b*f)/p;
        y0:=(a*f-c*d)/p;
        r:=sqrt((x1-x0)*(x1-x0)+(y1-y0)*(y1-y0));
end;
```

Процедура вычисления коэффициентов уравнения прямой procedure koefpr(x1,x2,y1,y2:real;var a,b,c:real); begin

```
    a:=y1-y2;
    b:=x2-x1;
    c:=(x1-x2)*y1+(y2-y1)*x1;
 end;
```

Процедура вычисления коэффициентов уравнения прямой, перпендикулярной заданной

```
 procedure koefperp(a,b,xc,yc:real;var a1,b1,c1:real);
 begin
     a1:=-b;
     b1:=a;
     c1:=b*xc-a*yc;
 end;
```

Процедуора определения координат точки пересечения двух прямых

```
 procedure peresech(a,b,c,d,e,f:real;var p,x0,y0:real);
 begin
    c:=-c;
    f:=-f;
```

```
          p:=a*e-d*b;
          x0:=(c*e-b*f)/p;
          y0:=(a*f-c*d)/p;
end;
      Процедура вычисления угла между двумя прямыми
procedure ugol(a1,b1,a2,b2,p:real; var fi:real);
const del=1e-8;
var c,s,t:real;
begin
            s:=a1*a2+b1*b2;
            c:=a1*b2-a2*b1;
            if (p<=del)and(c>100*s)or(s=0) then fi:=90
            else
            begin
                      t:=abs(c/s);
                      fi:=arctan(t)*180/pi;
            end;
end;
              Основная программа
begin
                Ввод исходных данных
          1: writeln(' Введите количество точек первого множества');
                readln(k1);
                if (k1>=3)and(k1<=ko) then
                  writeln(' количество введено верно')
                else
                begin
                                 if k1<0 then writeln('количество точек должно быть натуральным');
                                  if (k1>0)and(k1<3) then writeln ('количество точек недостаточно');
              if (k1>ko) then writeln('введено много точек');
              writeln ('если хотите повторить ввод, нажмите Y(y),',
                                'если завершить ввод - N(n)');
                                  ch:=UpCase(ReadKey);
                                  if ch='Y' then goto 1;
```

```
                    if ch='N' then goto 2;
          end;
3:            writeln(' Введите количество точек второго множества');
           readln(k2);
           if (k2>=3)and(k2<=kt) then
             writeln(' количество введено верно')
           else
           begin
                    if k2<0 then writeln('количество точек должно быть натуральным');
                     if (k2>0)and(k2<3) then writeln ('количество точек недостаточно');
          if (k2>kt) then writeln('введено много точек');
          writeln ('если   хотите   повторить  ввод,  нажмите Y(y),',
                    'если завершить ввод - N(n)');
                     ch:=UpCase(ReadKey);
                     if ch='Y' then goto 3;
                     if ch='N' then goto 2;
                end;
           for i:=1 to k1 do
           begin
                    writeln('введите координаты очередной точки первого множества ');
           readln(x1[i],y1[i]);
           end;
           for i:=1 to k2 do
           begin
                    writeln('введите координаты очередной точки второго множества ');
           readln(x2[i],y2[i]);
           end;

           ugolmin:=1000;
           kol:=0;
      Цикл перебора всех комбинаций по три точки первого множества
           for i:=1 to k1-2 do
              for j:=i+1 to k1-1 do
                 for k:=j+1 to k1 do
```

```
                 begin
Определение возможности построения окружности, проходящей через
                    три текущие точки
                         trit(x1[i],x1[j],x1[k],y1[i],y1[j],y1[k],kod);
                         if kod=-1 then goto 4;
                         if kod=0 then
Определение параметров текущей окружности
                         begin
                              okr(x1[i],x1[j],x1[k],y1[i],y1[j],y1[k],x0,y0,r);
Цикл перебора всех комбинаций по три точки второго множества
                              for i1:=1 to k2-2 do
                                 for j1:=i1+1 to k2-1 do
                                    for l:=j1+1 to k2 do
                                    begin
Определение возможности построения треугольника на трех точках
                            trit(x2[i1],x2[j1],x2[l],y2[i1],y2[j1],y2[l],kod);
                                       if kod=-1 then goto 5;
                                       if kod=0 then
                                       begin
Определение коэффициентов уравнений прямых-двух сторон треугольника
                                       kol:=kol+1;
                                       koefpr(x2[i1],x2[j1],y2[i1],y2[j1],a1,b1,c1);
                                       koefpr(x2[j1],x2[l],y2[j1],y2[l],a2,b2,c2);
Определение коэффициентов уравнений перпендикулярных прямых-высот
                                       koefperp(a1,b1,x2[l],y2[l],a1,b1,c1);
                                       koefperp(a2,b2,x2[i1],y2[i1],a2,b2,c2);
Определение координат точки пересечения высот
                                       peresech(a1,b1,c1,a2,b2,c2,p,xp,yp);
Определение коэффициентов уравнения прямой, проходящей через
             центр окружности и точку пересечения высот
                                       koefpr(x0,xp,y0,yp,a,b,c);
Определение угла между построенной прямой с осью ординат
                                       ugol(a,b,1,0,p,fi);
                                       if fi<ugolmin then
```

```
          begin
                    x0r:=x0;
                    y0r:=y0;
                    rr:=r;
                          x1r:=x1[i];
                          x2r:=x1[j];
                          x3r:=x1[k];
                    ugolmin:=fi;
                          y1r:=y1[i];
                          y2r:=y1[j];
                          y3r:=y1[k];
                    xpr:=xp;
                    ypr:=yp;
                    i2r:=i1;
                    j2r:=j1;
                    k2r:=l;
                    i1r:=i;
                    j1r:=j;
                    k1r:=k;
                    x12r:=x2[i1];
                    x22r:=x2[j1];
                    x32r:=x2[l];
                    y12r:=y2[i1];
                    y22r:=y2[j1];
                    y32r:=y2[l];
                    xr[1]:=x2[i1];
                    xr[2]:=x2[j1];
                    xr[3]:=x2[l];
                    yr[1]:=y2[i1];
                    yr[2]:=y2[j1];
                    yr[3]:=y2[l];
          end;
     end;
5: end; end;
```

```
                4:end;
    Вывод результатов в аналитическом виде
        writeln('Всего рассмотрено ', kol:4,' вариантов');
        if kol=0 then
        begin
                writeln('Нет окружностей или треугольников');
                goto 2;
        end;
writeln('Окружность с точками', i1r:4,j1r:4,k1r:4);
writeln('Их координаты',x1r:8:2,y1r:8:2,x2r:8:2,y2r:8:2,x3r:8:2,y3r:8:2);
writeln('Треугольник с точками', i2r:4,j2r:4,k2r:4);
writeln('Их координаты',x12r:8:2,y12r:8:2,x22r:8:2,y22r:8:2,
            x32r:8:2,y32r:8:2);
writeln('Угол равен', ugolmin:8:4,' градусов');
readln;
        поиск минимальных и максимальных координат
        xmin:=1e10; xmax:=-1e10;
        ymax:=-1e10; ymin:=1e10;
        if x0r-rr<xmin then xmin:=x0r-rr; для координат точек
        if x0r+rr>xmax then xmax:=x0r+rr;                 окружности
        if y0r-rr<ymin then ymin:=y0r-rr;
        if y0r+rr>ymax then ymax:=y0r+rr;
        for i:=1 to 3 do
        begin                                              для координат точек
                if xr[i]>xmax then xmax:=xr[i];                треугольника
                if xr[i]<xmin then xmin:=xr[i];
                if yr[i]>ymax then ymax:=yr[i];
                if yr[i]<ymin then ymin:=yr[i];
        end;                                               для координат точки
        if xpr<xmin then xmin:=xpr;                          пересечения высот
        if xpr>xmax then xmax:=xpr;
        if ypr<ymin then ymin:=ypr;
        if ypr>ymax then ymax:=ypr;
        gd:=detect;
```

```
    InitGraph(gd,gm,' ');
    GetAspectRatio(xa,ya);
  Определение поля вывода и его очерчивание
    xn:=0+delx;
    xk:=Getmaxx-delx;
    yn:=0+dely;
    yk:=GetMaxY-dely;
    Rectangle(xn,yn,xk,yk);
  Вычисление коэффициентов масштабирования
    kx:=(xk-xn)/(xmax-xmin);
    ky:=(yk-yn)/(ymax-ymin);
    if kx<=ky then km:=kx
                else km:=ky;
  Вычисление экранных координат центра окружности
    masst(x0r,y0r,xmin,ymax,km,xn,yn,kx0r,ky0r);
    kr:=round(rr*km);
    SetColor(12);
    Circle(kx0r,ky0r,kr); Вывод окружности
  Вычисление координат точек, через которые проходит окружность
    masst(x1[i1r],y1[i1r],xmin,ymax,km,xn,yn,kx1,ky1);
    masst(x1[j1r],y1[j1r],xmin,ymax,km,xn,yn,kx2,ky2);
    masst(x1[k1r],y1[k1r],xmin,ymax,km,xn,yn,kx3,ky3);
    SetColor(10);
    Circle(kx0r,ky0r,3);
  Вывод трех точек в виде малых окружностей,
  через которые проходит окружность
    Circle(kx1,round(ky0r+(ky1-ky0r)/ya*xa),3);
    Circle(kx2,round(ky0r+(ky2-ky0r)/ya*xa),3);
    Circle(kx3,round(ky0r+(ky3-ky0r)/ya*xa),3);
 Вычисление экранных координат трех вершин треугольника
    for i:=1 to 3 do
    begin
      masst(xr[i],yr[i],xmin,ymax,km,xn,yn,kxr[i],kyr[i]);
      kyr[i]:=round(ky0r+(kyr[i]-ky0r)/ya*xa);
```

```
    end;
      masst(xpr,ypr,xmin,ymax,km,xn,yn,kxpr,kypr);
  Вывод треугольника и его вершин в виде малых окружностей
    for i:=1 to 3 do
    begin
          Circle(kxr[i],kyr[i],3); if i<3 then
             Line(kxr[i],kyr[i],kxr[i+1],kyr[i+1]); end;
    Line(kxr[1],kyr[1],kxr[3],kyr[3]); SetColor(15);
  Рисование искомой прямой
    Line(kx0r,ky0r,kxpr,round(ky0r+(kypr-ky0r)/ya*xa));
    SetColor(13);
  Рисование прямых - высот треугольника
    for i:=1 to 3 do
       Line(kxr[i],kyr[i],kxpr,round(ky0r+(kypr-ky0r)/ya*xa)); readln;
    CloseGraph;
2:
end.
```

При решении приведенных задач могут оказаться полезными следующие формулы:

- угол между двумя направленными отрезками P1P2 и P3P4:

(X2-X1)*(X4-X3)+(Y2-Y1)*(Y4-Y3) cost=---

____________________ ___________________ »(X2-X1)S+(Y2-Y1)S

* »(X4-X3)S+(Y4-Y3)S

- Уравнение прямой:

Y=K*X+B

K=tgf X   Y

- + - =1 (A<>0, B<>0) A   B

- уравнение прямой, проходящей через две точки: Y-Y1   X-X1   ! X Y 1 !

----- = -----   или   ! X1 Y1 1 ! = 0

Y2-Y1   X2-X1   ! X2 Y2 1 !

- расстояние от точки Po(Xo,Yo) до прямой A*X+B*Y+C=0:

A*Xo + B*Yo +C
d=----------------- (знак противоположен знаку С);

» AS + BS

- уравнения прямых, проходящих через точку (X1,Y1) под углом f к прямой A1*X+B1*Y+C1=0

B1*tgf-A1
Y-Y1=--------- * (X-X1)
A1*tgf+B1

B1*tgf-A1
Y-Y1=--------- * (X-X1)
A1*tgf-B1