

Введение

Цель:

Разработать программу, которая на основе введенных целочисленных координат трех точек на плоскости вычисляет основные геометрические характеристики треугольника (длины сторон, длину биссектрисы из наименьшего угла), определяет тип треугольника (тупоугольный или нет), и проверяет принадлежность произвольной точки этому треугольнику.

Задачи:

1. Разработать алгоритм для вычисления расстояний между заданными точками на плоскости, представляющими вершины треугольника.
2. Реализовать алгоритм для вычисления углов треугольника и нахождения наименьшего угла.
3. Реализовать формулу для вычисления длины биссектрисы, исходящей из вершины с наименьшим углом.
4. Определить, является ли треугольник тупоугольным, на основе вычисленных углов.
5. Реализовать алгоритм для проверки принадлежности произвольной точки заданному треугольнику (метод площади или метод векторного произведения).
6. Организовать корректный ввод и вывод данных, обеспечивая возможность проверки корректности всех этапов выполнения программы.

▪ Аналитическая часть

Структура представлена на рисунке 1 в виде IDEF0-нотации. Задача разделяется на несколько блоков, представленных на рисунке 2:

- Ввод данных с клавиатуры
- Вчисление длин сторон
- Проверка на тупоугольность треугольника
- Вычисление биссектрисы
- Проверка, находится ли новая точка внутри треугольника

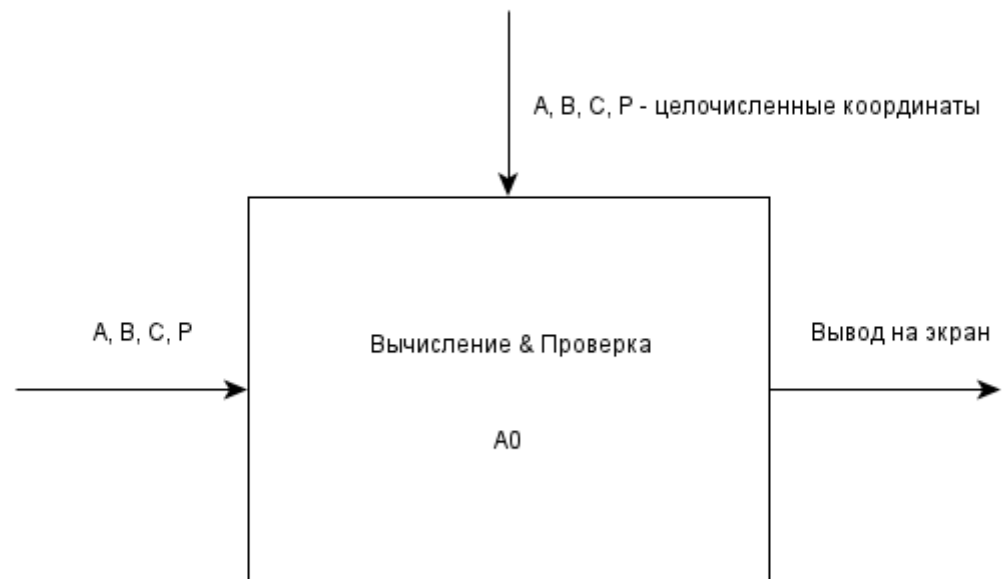


Рисунок 1 - Общая IDEF0-Нотация

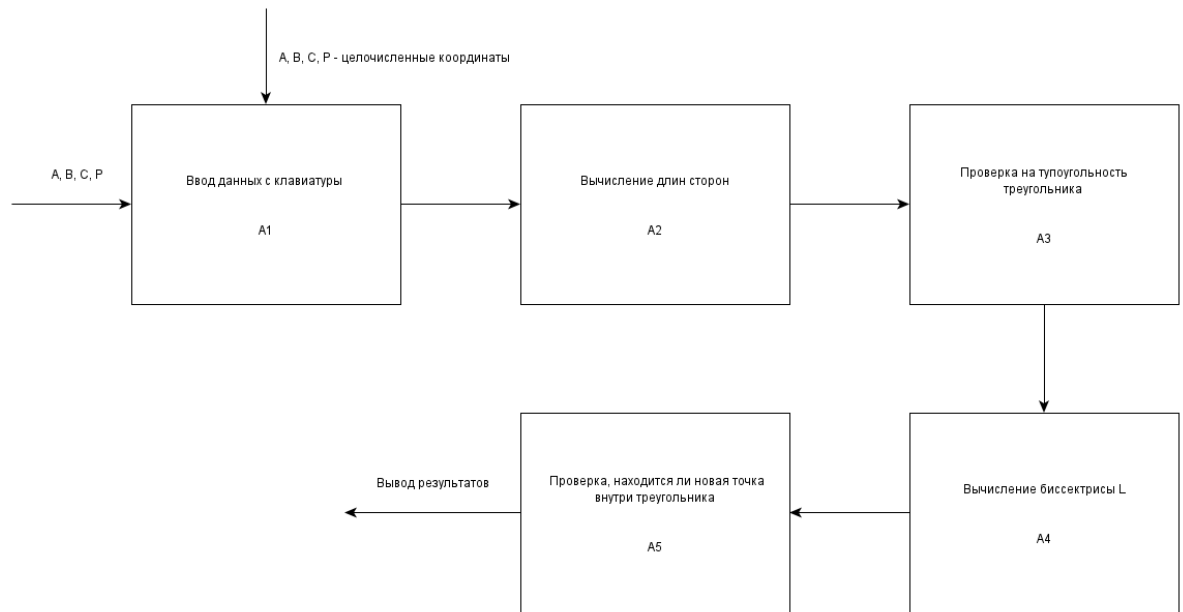


Рисунок 2 - Подробная IDEF0-Нотация

■ Конструкторская часть

Блок-схема алгоритма представлена на рисунке 1. Элементы, отвечающие за интерфейс пользователя, на блок-схеме не отображены; текстовые сообщения, ввиду малозначимости их дословного приведения, представлены сокращенно.

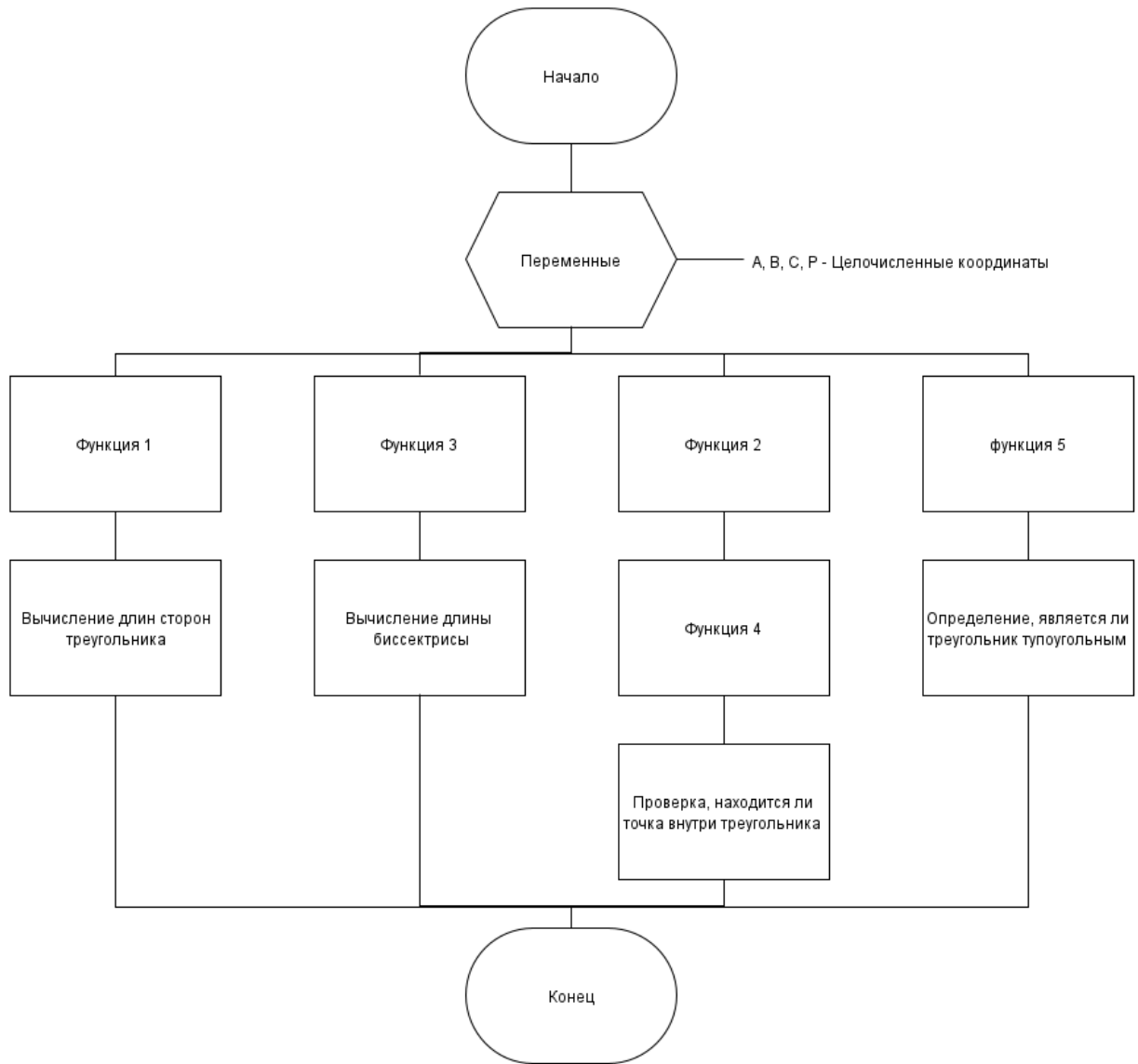


Рисунок 1 - Блок-схема алгоритма программы

■ Технологическая часть

Реализация алгоритма

В настоящем разделе представлена реализация алгоритма, чья блок-схема представлена на рисунке 1. Реализация была произведена с помощью языка программирования Pascal и представлена в листинге 1.

Листинг 1 - Программа алгоритма

```
program TriangleTask;

uses Math;

type
  TPoint = record
    x, y: Real;
  end;

var
  A, B, C, P: TPoint;
  AB, BC, CA: Real;
  bisectorA: Real;
  areaABC, areaABP, areaBCP, areaCAP: Real;
  isObtuse: Boolean;

// Функция для вычисления расстояния между двумя точками
function Distance(p1, p2: TPoint): Real;
begin
  Distance := Sqrt(Sqr(p2.x - p1.x) + Sqr(p2.y - p1.y));
end;

// Функция для вычисления площади треугольника по координатам трех точек
function TriangleArea(p1, p2, p3: TPoint): Real;
begin
  TriangleArea := Abs((p1.x*(p2.y - p3.y) + p2.x*(p3.y - p1.y) + p3.x*(p1.y - p2.y)) /
2.0);
end;

// Вычисление длины биссектрисы из вершины
function Bisector(a, b, c: Real): Real;
begin
  Bisector := Sqrt(a * b * (1 - Sqr(c) / Sqr(a + b)));
end;

// Проверка принадлежности точки треугольнику
function IsPointInTriangle(P, A, B, C: TPoint): Boolean;
begin
  areaABC := TriangleArea(A, B, C);
  areaABP := TriangleArea(A, B, P);
  areaBCP := TriangleArea(B, C, P);
```

```

    areaCAP := TriangleArea(C, A, P);
    IsPointInTriangle := (Abs(areaABC - (areaABP + areaBCP + areaCAP)) < 1e-6);
end;

// Проверка, является ли треугольник тупоугольным через длины сторон
function IsObtuseTriangle(a, b, c: Real): Boolean;
begin
    if (Sqr(a) > Sqr(b) + Sqr(c)) or (Sqr(b) > Sqr(a) + Sqr(c)) or (Sqr(c) > Sqr(a) + Sqr(b))
    then
        IsObtuseTriangle := True
    else
        IsObtuseTriangle := False;
    end;
end;

begin
    // Ввод координат вершин треугольника
    WriteLn('Введите координаты вершин треугольника:');
    Write('A (x, y): '); ReadLn(A.x, A.y);
    Write('B (x, y): '); ReadLn(B.x, B.y);
    Write('C (x, y): '); ReadLn(C.x, C.y);

    // Вычисление длин сторон треугольника
    AB := Distance(A, B);
    BC := Distance(B, C);
    CA := Distance(C, A);

    WriteLn('Длины сторон треугольника:');
    WriteLn('AB = ', AB:0:2);
    WriteLn('BC = ', BC:0:2);
    WriteLn('CA = ', CA:0:2);

    // Определение, является ли треугольник тупоугольным
    isObtuse := IsObtuseTriangle(AB, BC, CA);
    if isObtuse then
        WriteLn('Треугольник является тупоугольным.')
    else
        WriteLn('Треугольник не является тупоугольным.');
```

```

    // Вычисление длины биссектрисы, проведенной из стороны с наименьшей суммой
    if (AB <= BC) and (AB <= CA) then
        bisectorA := Bisector(BC, CA, AB)
    else if (BC <= AB) and (BC <= CA) then
        bisectorA := Bisector(CA, AB, BC)
    else
        bisectorA := Bisector(AB, BC, CA);

    WriteLn('Длина биссектрисы: ', bisectorA:0:2);

    // Ввод координат произвольной точки
    WriteLn('Введите координаты произвольной точки:');
    Write('P (x, y): '); ReadLn(P.x, P.y);

```

```
// Проверка, находится ли точка внутри треугольника
if IsPointInTriangle(P, A, B, C) then
    WriteLn('Точка находится внутри треугольника.')
else
    WriteLn('Точка находится вне треугольника.');
```

end.

Тестирование реализации

1. Метод эквивалентного разбиения

Метод предполагает разбиение множества входных данных на классы эквивалентности.

Ключевыми данными являются:

1. Координаты точек треугольника A, B, C .
2. Координаты произвольной точки P .

Разбиение:

1. Треугольник может быть:
 - Равносторонним.
 - Равнобедренным.
 - Разносторонним.
 - Вырожденным (площади = 0).
2. Точка P :
 - Находится внутри треугольника.
 - Лежит на стороне треугольника.
 - Находится вне треугольника.

2. Метод комбинаторного покрытия условий

Метод предполагает комбинацию различных условий, чтобы охватить все возможные варианты работы программы.

Условия:

1. Треугольник может быть тупоугольным, остроугольным или прямоугольным.
2. Точка может быть внутри, на границе или вне треугольника.

Результаты тестирования

№	Входные данные	Ожидаемый результат
1	$A(0,0), B(1,0), C(0,1), P(0.1, 0.1)$	Точка внутри треугольника, треугольник остроугольный
2	$A(0,0), B(4,0), C(0,3), P(5,5)$	Точка вне треугольника, треугольник остроугольный
3	$A(0,0), B(2,0), C(1,1), P(1,0)$	Точка на стороне треугольника, треугольник остроугольный
4	$A(0,0), B(5,0), C(2.5,5), P(3,2)$	Точка внутри треугольника, треугольник тупоугольный
5	$A(0,0), B(2,0), C(1,0), P(1,1)$	Вырожденный треугольник, площадь = 0
6	$A(0,0), B(3,0), C(0,4), P(1,1)$	Треугольник остроугольный, точка внутри
7	$A(0,0), B(6,0), C(3,5), P(10,10)$	Треугольник тупоугольный, точка вне
8	$A(0,0), B(4,0), C(2,4), P(2,0)$	Треугольник прямоугольный, точка на границе

Заключение

В этом проекте была написана программа для решения следующих задач:

1. После ввода трех координат, вычисление длины сторон треугольника, образованного этими координатами.
2. Вычисление длины биссектрисы противолежащей стороны от наименьшего угла треугольника.
3. Проверка, является ли треугольник тупоугольным.
4. После ввода произвольной координаты, проверка, находится ли эта точка внутри треугольника.