Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**Пермский национальный исследовательский**

**политехнический университет**

ЭТФ

ИТАС

Факультет\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Кафедра\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КОБ

Специальность \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ОТЧЕТ**

* **лабораторной работе № 11**

Разработка программы вычисления координат критических точек

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(*наименование лабораторной работы*)

Выполнил:

студент группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КОБ-23-1С

Горкунов М. В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(*фамилия*, *инициалы*)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(*подпись*)

Проверил:

доцент кафедры ИТАС Тарутин А.В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(*оценка*)

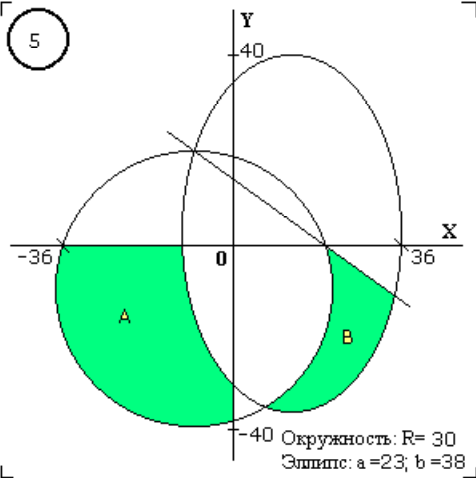
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(*подпись*)

Цель лабораторной работы: разработать алгоритм и программу, которая в ответ на вводимую с клавиатуры пару вещественных чисел, представляющих собой координаты X,Y точки А на плоскости, выдает сообщение о принадлежности этой точки к заданной закрашенной области.

Содержание лабораторной работы: В ходе работы предусмотреть в программе повтор описанных действий с новыми значениями координат X, Y в ответ на соответствующий запрос программы.

На рисунке изображены две закрашенные области. Необходимо узнать входит ли введённая точка, с координатами X, Y в закрашенные области.

Для решения задачи понадобятся следующие формулы:

Уравнение окружности:

Уравнение эллипса:

Уравнение прямой:

Алгоритм программы:

1. Вычесляем точки нужные для уравнений окружности, элипса и прямой.
2. Вычесляем коофиценты уравнений.
3. Ждем ввода пользователя X, Y точки для проверки ее на вхаждение в области.
4. Проверяем входит ли точка в закрашеные области, если да то выводим в какую именно, если нет выводим, что точка не принадлежит не одной области

Программный код:

const r = 30.0;

const a = 23.0;

const b = 38.0;

const circleYProjection = -40.0;

const circleXIntersection = -36.0;

const ellipseXIntersectin = 36.0;

const ellipseYProjection = 40.0;

const accuracy = 0.0001;

function GetSecondCoordinateOfCircle(coordinate, radius : real) : real; begin

Result := sqrt(sqr(radius) - sqr(coordinate));

end;

function GetSecondCoordinateOfEllipse(coordinate, radius1, radius2 : real) : real; begin

Result := radius1 \* sqrt(1 - sqr(coordinate / radius2))

end;

function GetPoint() : (real, real); begin

var x, y : real;

write('Enter the point (х y) : ');

while true do begin

var str := Console.ReadLine();

var numbers := str.Trim()

.Split(' ')

.Select(x -> x.Trim())

.Where(x -> x <> '')

.ToArray();

if (numbers.Length = 2) and

real.TryParse(numbers[0], x) and

real.TryParse(numbers[1], y)

then break;

write('Incorrect input, try again: ');

end;

Result := (x, y);

end;

begin

var circleY := circleYProjection + r;

var circleX := circleXIntersection + GetSecondCoordinateOfCircle(-circleY, r);

var ellipseY := ellipseYProjection - b;

var ellipseX := ellipseXIntersectin - GetSecondCoordinateOfEllipse(-ellipseY, a, b);

var circleLineXIntersection := circleX + GetSecondCoordinateOfCircle(-circleY, r);

var circleEllipseIntersectionX := circleXIntersection;

var circleEllipseIntersectionY := circleY + GetSecondCoordinateOfCircle(circleEllipseIntersectionX - circleX, r);

while sqr((circleEllipseIntersectionX - ellipseX) / a) + sqr((circleEllipseIntersectionY - ellipseY) / b) > 1 do begin

circleEllipseIntersectionX += accuracy;

circleEllipseIntersectionY := circleY + GetSecondCoordinateOfCircle(circleEllipseIntersectionX - circleX, r);

end;

var k := -circleEllipseIntersectionY / (circleLineXIntersection - circleEllipseIntersectionX);

var c := - k \* circleLineXIntersection ;

while true do begin

var (x, y) := GetPoint();

var isToTheRightOfYAxis := 0 <= x;

var isAboveXAxis := 0 <= y;

var isAboveTheLine := k \* x + c <= y;

var isInsideTheCircle := sqr(x - circleX) + sqr(y - circleY) <= sqr(r);

var isInsideTheEllipse := sqr((x - ellipseX) / a) + sqr((y - ellipseY) / b) <= 1;

if isInsideTheCircle and not isInsideTheEllipse and

not isToTheRightOfYAxis and not isAboveXAxis then

writeln('Point in the A area.')

else if not isAboveTheLine and

isInsideTheEllipse and not isInsideTheCircle then

writeln('Point in the B area.')

else writeln('Point outside of A and B areas.');

end;

end.

Вывод: во время выполнения данной работы был разротан алгоритм и написана программа, которая в ответ на введенные координаты X, Y точки, определяет принадлежность этой точки заданным областям.