Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

им. В.И. Ульянова (Ленина)»

Кафедра вычислительной техники

Отчет по лабораторной работе №1

по дисциплине «Программирование»

На тему «Условные операторы»

Студент гр. 9308 Семенов Алексей Иванович

Проверил доцент Сискович Тамара Ильинична

Санкт-Петербург,

2019

**Содержание**

Введение………………………………………………………………..3

1. Задание………………………………………………………………3

2. Уточнение задания…………………………………………………..4

3. Описание переменных………………………………………………4

4. Контрольные примеры……………………………………………...5

5. Краткое описание алгоритма……………………………………….6

6. Схема алгоритма…………………………………………………….7

7. Текст программы…………………………………………………….11

8. Результаты работы программы.………………………………...…12

Заключение……………………………………………………………..12

**Введение**

Целью выполнения лабораторной работы является приобретение практических навыков разработки разветвляющихся алгоритмов и использование условных операторов для программирования

**1. Задание**

Разработать алгоритм, написать программу, проверяющую местоположение точки заданной координатами x,y на плоскости. Программа должна проверить, находится ли точка в заштрихованной области, на границе или вне области. Заданная область показана на рисунке 1. Предусмотреть возможность повторения программы по желанию пользователя и изменения параметров.

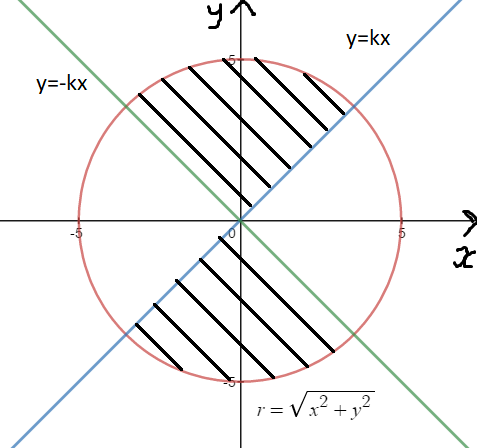


Рис. 1. Заданная область

**2. Уточнение задания**

Заданы функции y=kx, y=-kx, r^2=x^2+y^2. На вход программа получает числа: k,r,x,y. Программа должна определить, принадлежит ли точка с координатами (x,y) к области, заданной r^2 > x^2+y^2, y > kx и y > -kx при y > 0, или y < kx и y < -kx при y < 0, на ее границах или не принадлежит ей.

**3. Описание переменных**

Описание переменных приведено в таблице 1.

Таблица 1. Описание переменных

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Назначение** |
| x | float | Абсцисса точки |
| y | float | Ордината точки |
| k | int | Коэффициент для уравнений y=kx и y=-kx |
| r | int | Радиус окружности |

**4. Контрольные примеры**

Контрольные примеры приведены в таблице 2.

Таблица 2. Контрольные примеры

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № примера | Исходные данные | | | | Результат |
| X | Y | K | R |
| 1 | - 4,3 | 2,5 | 3 | 8 | Точка вне области |
| 2 | 0 | 10,0 | 5 | 10 | Точка на границе |
| 3 | - 0,6 | 0,28 | 1 | 2 | Точка вне области |
| 4 | - 1,7 | 3,4 | 2 | 6 | Точка на границе |
| 5 | - 4,0 | 0 | 102 | 4 | Точка вне области |
| 6 | 0,05 | 3,9 | 50 | 4 | Точка в области |
| 7 | 0,7 | 7,0 | 21 | 7 | Точка вне области |
| 8 | 0,5 | 5,978 | 7 | 6 | Точка в области |
| 9 | 0,7 | 3,7 | 2 | 1 | Точка вне области |
| 0 | 0 | 0,02 | 10 | 1 | Точка в области |

**5. Краткое описание алгоритма**

Шаг 1.

Ввод коэффициента K и радиуса окружности R, координат точки.

Шаг 2.

Проверяется, находится ли точка вне окружности. Если точка находится за границами окружности, то выводится сообщение «Точка вне области», и происходит переход к Шагу 6.

Шаг 3.

Проверяется, находится ли точка на границе обозначенной области. Если точка - на границе, то выводится сообщение «Точка на границе области», и происходит переход к Шагу 6.

Шаг 4.

Проверяется, находится ли точка в обозначенной области. Если да, то выводится сообщение «Точка в области», и происходит переход к Шагу 6.

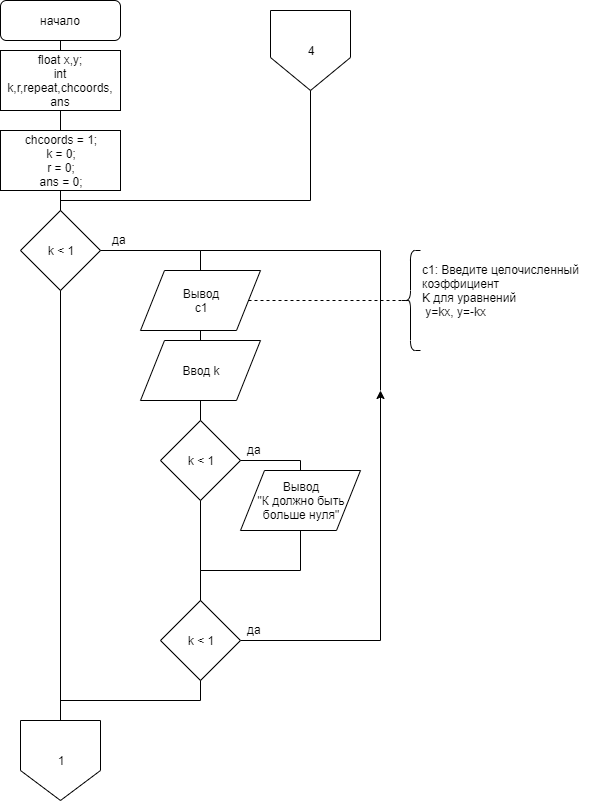
Шаг 5.

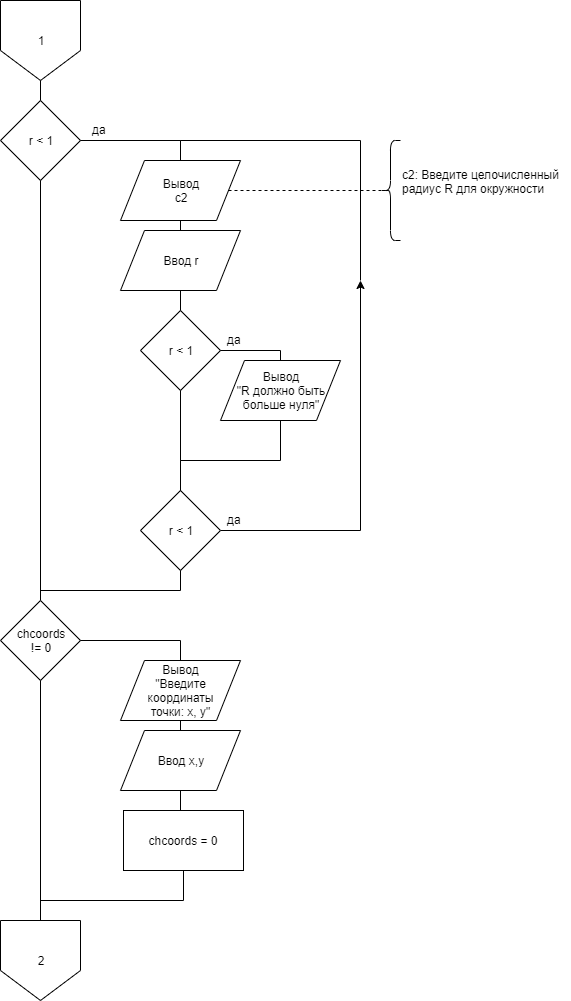
Если не была пройдена ни одна из проверок, то выводится сообщение «Точка вне области».

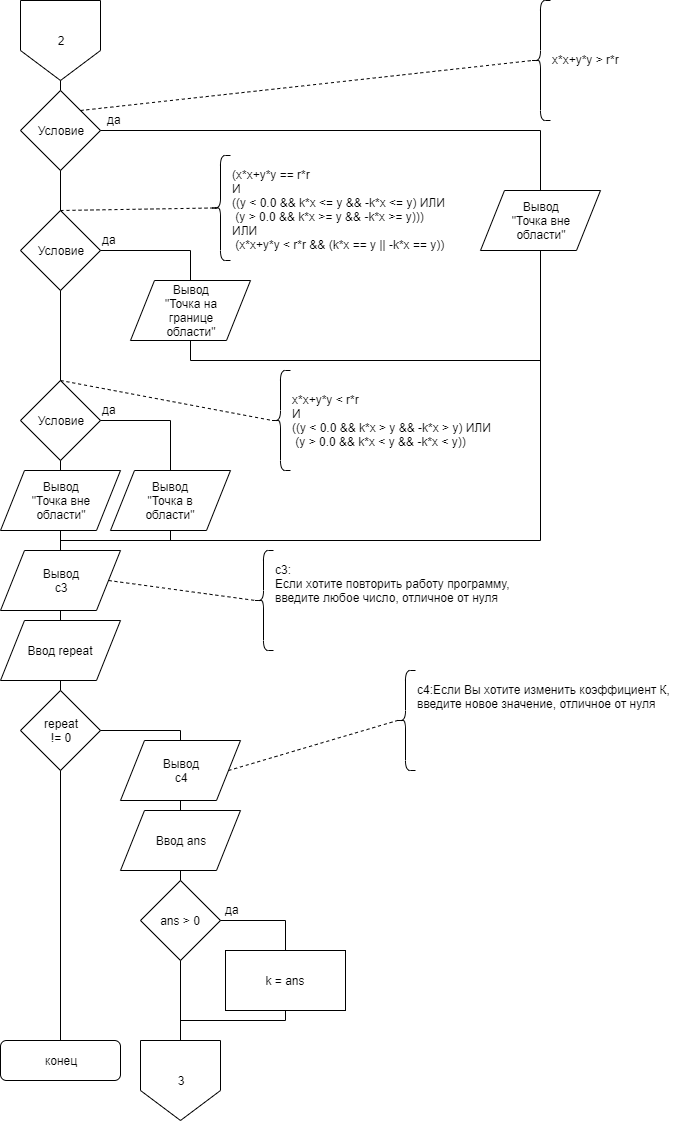
Шаг 6.

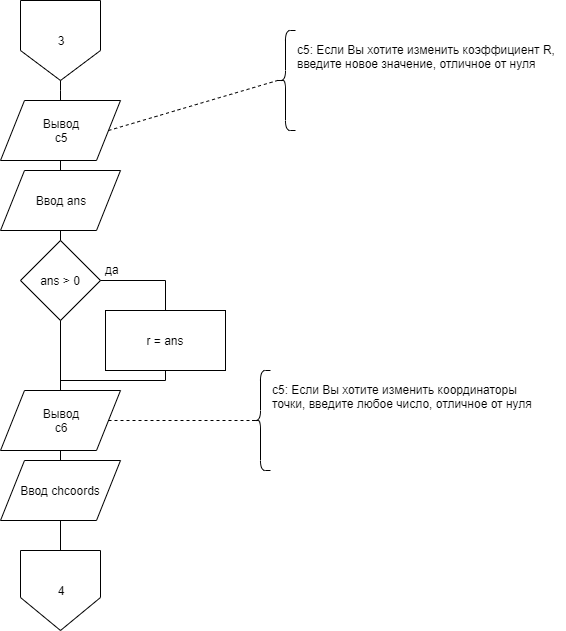
Пользователю предлагается повторить работу программы с изменением параметров по его желанию

**6. Схема алгоритма**









*Рисунок 2. Схема алгоритма*

**7. Текст программы**

#include <stdio.h>

#include <locale.h>

int main()

{

setlocale(LC\_ALL, "russian");

float x, //координаты точки

y;

int k, //Коэффициент К для уравнений y=kx y=-kx

r, //Радиус окружности

repeat, //Флаг повтора программы

chcoords, //Флаг для изменения координат

ans; //Флаг для изменения коэффициентов К и радиуса R

repeat = 0, chcoords = 1, k = 0, r = 0, ans = 0;

do

{

while (k < 1)

{

puts("Введите целочисленный коэффициент K для уравнений y=kx, y=-kx");

scanf("%d", &k);

if (k < 1) puts("К должно быть больше нуля!");

}

while (r < 1)

{

puts("Введите целочисленный радиус R для окружности");

scanf("%d", &r);

if (r < 1) puts("R должно быть больше нуля!");

}

if (chcoords)

{

puts("Введите координаты точки: x, y");

scanf("%f %f", &x, &y);

chcoords = 0;

}

if (x \* x + y \* y > r\* r)

{

puts("Точка вне области");

}

else if ((x \* x + y \* y == r \* r && ((y < 0.0 && k \* x <= y && -k \* x <= y) || (y > 0.0 && k \* x >= y && -k \* x >= y))) || (x \* x + y \* y < r \* r && (k \* x == y || -k \* x == y)))

{

puts("Точка на границе области");

}

else if (x \* x + y \* y < r \* r && ((y < 0.0 && k \* x > y && -k \* x > y) || (y > 0.0 && k \* x < y && -k \* x < y))) puts("Точка в области");

else puts("Точка вне области");

puts("Если хотите повторить работу программу, введите любое число, отличное от нуля");

scanf("%d", &repeat);

if (repeat)

{

puts("Если Вы хотите изменить коэффициент К, введите новое значение, отличное от нуля");

scanf("%d", &ans);

if (ans > 0) k = ans;

puts("Если Вы хотите изменить коэффициент R, введите новое значение, отличное от нуля");

scanf("%d", &ans);

if (ans > 0) r = ans;

puts("Если Вы хотите изменить координаторы точки, введите любое число, отличное от нуля");

scanf("%d", &chcoords);

}

puts("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

} while (repeat);

return 0;

}

**8. Результаты работы программы**

В процессе выполнения программы получены результаты, совпадающие с контрольными примерами, указанными в таблице 2. Ошибок в процессе работы программы не обнаружено

**Заключение**

При выполнении лабораторной работы получены практические навыки разработки разветвляющихся алгоритмов и использование условных операторов для программирования