МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Отчет № 11

по дисциплине «Информатика»

на тему: «Подпрограммы. Подпрограмма-функция и подпрограмма-процедура. Работа с массивами. Сложные сортировки. Сортировка массива сложным выбором (с помощью двоичного дерева)»

Выполнил:

студент группы 3530902/90001 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Непушкин Сергей Александрович

Проверил:

Доцент ВШКФСиУ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Теплова Наталья Витальевна

Санкт-Петербург

2019 г.

Оглавление

1.Задание………………………………………………………………………………………….3

2.Блок-схема алгоритма…………………………………………………………………………4

3.Текст кода………………………………………………………………………………………5

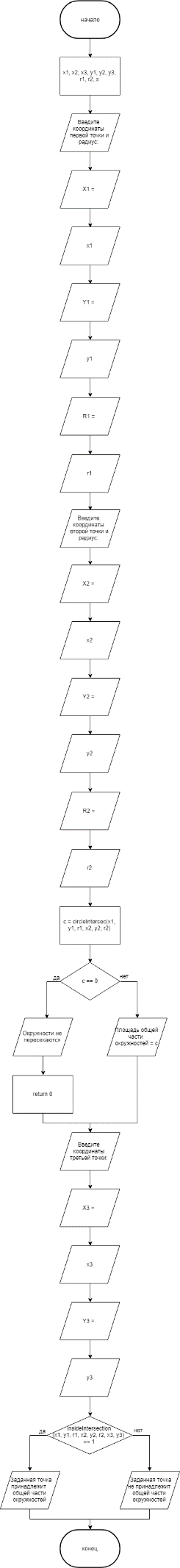
4.Пример работы программы……………………………………………………………………7

**1. Задание.**

Написать программу, рассчитывающую площадь пересечения двух кругов.  Определить, принадлежит ли точка с заданными координатами (х3, у3) площади пересечения.

Входные данные: х1, у1, R1, x2, y2, R2, х3, у3

**2.Блок-схема алгоритма**

****

**3.Текст кода.**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <math.h>

#include <locale.h>

#include <stdlib.h>

#include <time.h>

#define PI 3.1415926535

double dist(double x1, double y1, double x2, double y2)

{

double a = (x1 - x2) \* (x1 - x2);

double b = (y1 - y2) \* (y1 - y2);

double res = sqrt(a + b);

return res;

}

double triagS(double a, double b, double c)

{

double p = (a + b + c) / 2;

double S = sqrt(p \* (p - a) \* (p - b) \* (p - c));

return S;

}

double circleIntersec(double x1, double y1, double r1, double x2, double y2, double r2)

{

double d = dist(x1, y1, x2, y2);

if (d >= r1 + r2)

{

return 0;

}

double minR = (r1 > r2 ? r2 : r1);

double maxR = (r1 > r2 ? r1 : r2);

double r = maxR - minR;

if (d <= r)

{

return PI \* minR \* minR;

}

double S = 2 \* triagS(maxR, minR, d);

double h = 2 \* S / d;

double alphaMax = acos(1 - h \* h / 2 / maxR / maxR);

double alphaMin = acos(1 - h \* h / 2 / minR / minR);

double sectMax = alphaMax \* maxR \* maxR / 2;

double sectMin = alphaMin \* minR \* minR / 2;

double triagMax = maxR \* maxR \* sin(alphaMax) / 2;

double triagMin = minR \* minR \* sin(alphaMin) / 2;

if (d > maxR)

{

return sectMax - triagMax + sectMin - triagMin;

}

else

{

return sectMax - triagMax + PI \* minR \* minR - sectMin + triagMin;

}

}

int insideIntersection(double x1, double y1, double r1, double x2, double y2, double r2, double x3, double y3)

{

double dist1 = dist(x1, y1, x3, y3);

double dist2 = dist(x2, y2, x3, y3);

if (dist1 <= r1 && dist2 <= r2)

{

return 1;

}

else

{

return 0;

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

double x1, x2, x3, y1, y2, y3, r1, r2, s;

printf("Введите координаты первой точки и радиус: \n");

printf("X1 = ");

scanf("%lf", &x1);

printf("Y1 = ");

scanf("%lf", &y1);

printf("R1 = ");

scanf("%lf", &r1);

printf("Введите координаты второй точки и радиус: \n");

printf("X2 = ");

scanf("%lf", &x2);

printf("Y2 = ");

scanf("%lf", &y2);

printf("R2 = ");

scanf("%lf", &r2);

double c = circleIntersec(x1, y1, r1, x2, y2, r2);

if (c == 0)

{

printf("Окружности не пересекаются");

return 0;

}

else

{

printf("Площадь общей части окружностей = %lf \n", c);

}

printf("Введите координаты третьей точки: \n");

printf("X3 = ");

scanf("%lf", &x3);

printf("Y3 = ");

scanf("%lf", &y3);

if (insideIntersection(x1, y1, r1, x2, y2, r2, x3, y3) == 1)

{

printf("Заданная точка принадлежит общей части окружностей");

}

else

{

printf("Заданная точка не принадлежит общей части окружностей");

}

return 0;

}

**4.Пример работы программы**

