МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

### федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

(ГУАП)

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

Преподаватель

канд. техн. наук, доцент Л.Н. Бариков

Отчёт

по лабораторной работе №6

по дисциплине ОСНОВЫ ПРОГРАММИРОВАНИЯ

на тему: «Перегрузка функций»

Работу выполнил

студент гр. 4141 В.С. Сыворотнев

Санкт-Петербург

2022

***Цель лабораторной работы:*** *изучение концепции подпрограмм и способов передачи аргументов в функции, освоение методов процедурного программирования на языке C++ при решении типовых вычислительных задач.*

***Задание на программирование:*** *используя технологию процедурного программирования, разработать программу, решающую поставленную задачу в соответствии с индивидуальным заданием.*

***Вариант № 31:***

*Разработать функцию, определяющую координаты центра и радиус описанной окружности для треугольника, заданного координатами вершин этого треугольника на плоскости.*

*Ввести координаты вершин трёх треугольников. Используя разработанную функцию найти наибольший радиус среди окружностей, описывающих эти треугольники.*

***Математическая модель решения***

По условию задачи необходимо разработать функцию, которая должна быть реализована 3 способами:

- как функция, возвращающая искомое значение;

- как функция, возвращающая искомое значение через дополнительный аргумент - указатель;

- как функция, возвращающая искомое значение через дополнительный аргумент - ссылку.

Эти три функции будут иметь одно и то же имя, т.е. будут перегруженными.

В рассматриваемом примере нахождение наибольшего радиуса описанной окружности в функциях будет осуществляться по формуле

R = √ ( (x - circleX)^2 + (y – circleY)^2 ).

Конкретное значение наибольшего радиуса для всех перечисленных реализаций будет передаваться в каждую функцию через аргумент.

При вызове первой из реализуемых функций искомый результат будет возвращаться оператором *return*.

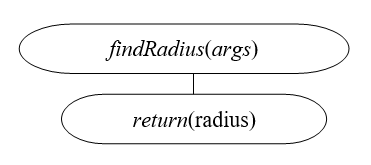
Во втором случае искомое значение будет возвращаться через дополнительный аргумент – указатель.

В третьем случае искомое значение будет возвращаться через дополнительный аргумент – ссылку.

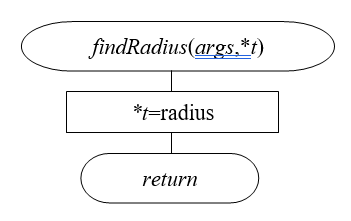
Решение задачи начинается с ввода исходных данных. В качестве исходных данных вводятся координаты вершин для трёх треугольников. После этого следуют последовательные вызовы каждой из реализованных функций. После чего выводятся результаты вычислений.

Задача решена.

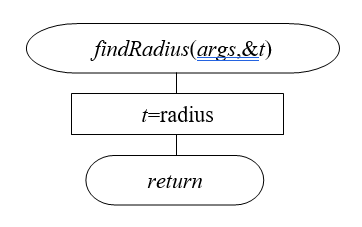
***Схема алгоритма функции, возвращающей значение наибольшего радиуса описанной окружности (функция findRadius)***

**

***Схема алгоритма функции, возвращающей значение наибольшего радиуса описанной окружности через аргумент-указатель (функция findRadius)***

******

***Схема алгоритма функции, возвращающей значение наибольшего радиуса описанной окружности через аргумент-ссылку (функция findRadius)***

**

***Текст программы***

*/\* Разработать функцию, определяющую координаты центра и радиус описанной окружности для треугольника,*

*заданного координатами вершин этого треугольника на плоскости.*

*Ввести координаты вершин трёх треугольников. Используя разработанную функцию найти наибольший радиус*

*среди окружностей, описывающих эти треугольники.*

*\*/*

//Реализованы три функции с одним именем:

//-функция, возвращающая значение;

//-функция с дополнительным аргументом-указателем;

//-функция с дополнительным аргументом-ссылкой.

//Вызов всех функций - из функции main()

#include<iostream>

#include<cmath>

#include<locale.h>

using namespace std;

double findRadius(double, double, double, double, double, double);

void findRadius(double, double, double, double, double, double, double \*);

void findRadius(double, double, double, double, double, double, double &);

int main() {

    double ft\_x1, ft\_y1, ft\_x2, ft\_y2, ft\_x3, ft\_y3,

            st\_x1, st\_y1, st\_x2, st\_y2, st\_x3, st\_y3,

            tt\_x1, tt\_y1, tt\_x2, tt\_y2, tt\_x3, tt\_y3, ft\_radius, st\_radius, tt\_radius, max\_radius;

    setlocale(LC\_ALL,"Russian");

    cout << "Введите координаты вершин первого треугольника:" << endl;

    cin >> ft\_x1 >> ft\_y1 >> ft\_x2 >> ft\_y2 >> ft\_x3 >> ft\_y3;

    cout << "Введите координаты вершин второго треугольника:" << endl;

    cin >> st\_x1 >> st\_y1 >> st\_x2 >> st\_y2 >> st\_x3 >> st\_y3;

    cout << "Введите координаты вершин третьего треугольника:" << endl;

    cin >> tt\_x1 >> tt\_y1 >> tt\_x2 >> tt\_y2 >> tt\_x3 >> tt\_y3;

    cout.precision(4);

    ft\_radius = findRadius(ft\_x1, ft\_y1, ft\_x2, ft\_y2, ft\_x3, ft\_y3);

    st\_radius = findRadius(st\_x1, st\_y1, st\_x2, st\_y2, st\_x3, st\_y3);

    tt\_radius = findRadius(tt\_x1, tt\_y1, tt\_x2, tt\_y2, tt\_x3, tt\_y3);

    if (ft\_radius > st\_radius && ft\_radius > tt\_radius && isinf(ft\_radius) == 0) {

        max\_radius = ft\_radius;

    } else if (st\_radius > ft\_radius && st\_radius > tt\_radius && isinf(ft\_radius) == 0) {

        max\_radius = st\_radius;

    } else if (tt\_radius > ft\_radius && tt\_radius > st\_radius && isinf(ft\_radius) == 0) {

        max\_radius = tt\_radius;

    } else if (st\_radius == ft\_radius && ft\_radius == tt\_radius && isinf(ft\_radius) == 0) {

        max\_radius = st\_radius;

    }

    cout << "**\n**Функция, возвращающая значение" << endl;

    cout << "Наибольший радиус описанной окружности: " << max\_radius << endl;

    findRadius(ft\_x1, ft\_y1, ft\_x2, ft\_y2, ft\_x3, ft\_y3, &ft\_radius);

    findRadius(st\_x1, st\_y1, st\_x2, st\_y2, st\_x3, st\_y3, &st\_radius);

    findRadius(tt\_x1, tt\_y1, tt\_x2, tt\_y2, tt\_x3, tt\_y3, &tt\_radius);

    if (ft\_radius > st\_radius && ft\_radius > tt\_radius && isinf(ft\_radius) == 0) {

        max\_radius = ft\_radius;

    } else if (st\_radius > ft\_radius && st\_radius > tt\_radius && isinf(ft\_radius) == 0) {

        max\_radius = st\_radius;

    } else if (tt\_radius > ft\_radius && tt\_radius > st\_radius && isinf(ft\_radius) == 0) {

        max\_radius = tt\_radius;

    } else if (st\_radius == ft\_radius && ft\_radius == tt\_radius && isinf(ft\_radius) == 0) {

        max\_radius = st\_radius;

    }

    cout << "**\n**Функция c дополнительным аргументом-ссылкой" << endl;

    cout << "Наибольший радиус описанной окружности: " << max\_radius << endl;

    findRadius(ft\_x1, ft\_y1, ft\_x2, ft\_y2, ft\_x3, ft\_y3, ft\_radius);

    findRadius(st\_x1, st\_y1, st\_x2, st\_y2, st\_x3, st\_y3, st\_radius);

    findRadius(tt\_x1, tt\_y1, tt\_x2, tt\_y2, tt\_x3, tt\_y3, tt\_radius);

    if (ft\_radius > st\_radius && ft\_radius > tt\_radius && isinf(ft\_radius) == 0) {

        max\_radius = ft\_radius;

    } else if (st\_radius > ft\_radius && st\_radius > tt\_radius && isinf(ft\_radius) == 0) {

        max\_radius = st\_radius;

    } else if (tt\_radius > ft\_radius && tt\_radius > st\_radius && isinf(ft\_radius) == 0) {

        max\_radius = tt\_radius;

    } else if (st\_radius == ft\_radius && ft\_radius == tt\_radius && isinf(ft\_radius) == 0) {

        max\_radius = st\_radius;

    }

    cout << "**\n**Функция c дополнительным аргументом-указателем" << endl;

    cout << "Наибольший радиус описанной окружности: " << max\_radius << endl;

    return 0;

}

double findRadius(double x1, double y1, double x2, double y2, double x3, double y3) {

    double z, z1, z2, z3, circleX, circleY, radius;

    z1 = x1 \* x1 + y1 \* y1;

    z2 = x2 \* x2 + y2 \* y2;

    z3 = x3 \* x3 + y3 \* y3;

    z = (x1 - x2) \* (y3 - y1) - (y1 - y2) \* (x3 - x1);

    circleX = -((y1 - y2) \* z3 + (y2 - y3) \* z1 + (y3 - y1) \* z2) / (2 \* z);

    circleY = ((x1 - x2) \* z3 + (x2 - x3) \* z1 + (x3 - x1) \* z2) / (2 \* z);

    radius = sqrt(pow(x1 - circleX, 2) + pow(y1 - circleY, 2));

    return radius;

}

void findRadius(double x1, double y1, double x2, double y2, double x3, double y3, double \*t) {

    double z, z1, z2, z3, circleX, circleY, radius;

    z1 = x1 \* x1 + y1 \* y1;

    z2 = x2 \* x2 + y2 \* y2;

    z3 = x3 \* x3 + y3 \* y3;

    z = (x1 - x2) \* (y3 - y1) - (y1 - y2) \* (x3 - x1);

    circleX = -((y1 - y2) \* z3 + (y2 - y3) \* z1 + (y3 - y1) \* z2) / (2 \* z);

    circleY = ((x1 - x2) \* z3 + (x2 - x3) \* z1 + (x3 - x1) \* z2) / (2 \* z);

    radius = sqrt(pow(x1 - circleX, 2) + pow(y1 - circleY, 2));

    \*t = radius;

}

void findRadius(double x1, double y1, double x2, double y2, double x3, double y3, double &t) {

    double z, z1, z2, z3, circleX, circleY, radius;

    z1 = x1 \* x1 + y1 \* y1;

    z2 = x2 \* x2 + y2 \* y2;

    z3 = x3 \* x3 + y3 \* y3;

    z = (x1 - x2) \* (y3 - y1) - (y1 - y2) \* (x3 - x1);

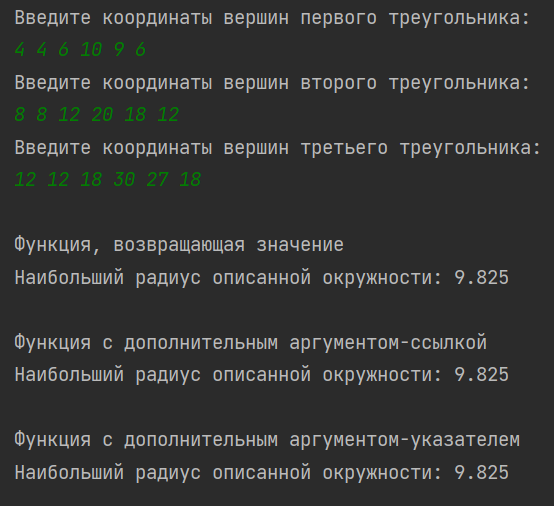
    circleX = -((y1 - y2) \* z3 + (y2 - y3) \* z1 + (y3 - y1) \* z2) / (2 \* z);

    circleY = ((x1 - x2) \* z3 + (x2 - x3) \* z1 + (x3 - x1) \* z2) / (2 \* z);

    radius = sqrt(pow(x1 - circleX, 2) + pow(y1 - circleY, 2));

    t = radius;

}

***Скриншот результатов выполнения программы***

***Вывод:***

*Я изучил концепции подпрограмм и способы передачи аргументов в функции, освоил методы процедурного программирования на языке C++ при решении типовых вычислительных задач.*