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 №11

по дисциплине ОСНОВЫ ПРОГРАММИРОВАНИЯ

на тему: «Передача имён функций в качестве параметров»

Работу выполнил

студент гр. 4941 Н.С. Горбунов

Санкт-Петербург

2020

**Цель лабораторной работы:** изучение методов передачи имён функций в качестве параметров других функций; изучение методов нахождения приближённого значения определённого интеграла произвольной функции на заданном интервале и с заданной точностью; совершенствование навыков процедурного программирования на языке C/C++ при решении задач с использованием рекурсивных функций.

**Задание на программирование:** используя технологию процедурного программирования реализовать заданный метод нахождения значения определённого интеграла произвольной функции на заданном интервале и с заданной точностью и применить его для определения площадей областей на плоскости в соответствии с индивидуальным заданием.

**Порядок выполнения работы:**

1. Получить у преподавателя индивидуальное задание:

- метод нахождения значения определённого интеграла произвольной функции на заданном интервале интегрирования и с заданной точностью;

- задачу определения места нахождения точки с произвольно заданными координатами на координатной плоскости (вариант и формулировка этой задачи берётся из лабораторной работы №2).

2. Разработать математическую модель:

- привести уравнения линий, ограничивающих выделенные штриховкой области;

- описать условия попадания точки в каждую выделенную область;

- составить аналитические формулы определения площади каждой выделенной области;

- определить аналитический вид функций, графики которых соответствуют линиям, ограничивающим выделенные штриховкой области;

- определить площадь каждой выделенной области с использованием значений определённых интегралов соответствующих функций на заданном интервале интегрирования и с заданной точностью.

3. Составить программу на языке C/C++.

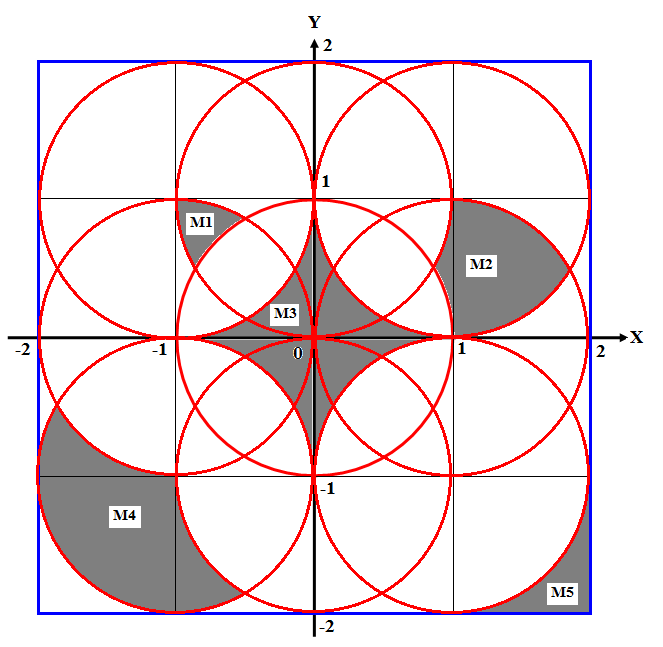
4. Входные данные должны вводиться с клавиатуры по запросу.

5. Выходные данные (сообщения) должны выводиться на экран в развернутой форме. Операторы вывода результатов работы должны находиться либо в функции main(), либо в специальной функции вывода, вызов которой осуществляется из функции main().

6. Проверить и продемонстрировать преподавателю работу программы на полном наборе тестов.

7. Оформить отчет о лабораторной работе в составе: постановка задачи, математическая модель, текст программы, контрольные примеры (скриншоты).

**Задание 13, метод 4**



**Математическая модель**

* условие 1 (принадлежность области *М*1):

{входит в окружность с центром в точке (0;1)}

{входит в окружность с центром в точке (-1;0)}

{не входит в окружность с центром в точке (0;0)}

Площадь области *М*1:

Площадь квадрата – 2\* (площадь квадрата -2\* площадь сектора в 30° - площадь равностороннего треугольника) – четверть площади круга = 0,1278

Аналитический вид функций, подлежащих интегрированию:

{верхняя полуокружность левой центральной окружности}; интервал интегрирования [-1,-0.5]; результат *s*1.

{верхняя полуокружность центральной окружности}; интервал интегрирования [-1,-0.5]; результат *s*2.

{нижняя полуокружность нижней центральной окружности}; интервал интегрирования [-1, - / 2]; результат *s*3.

Площадь области *M*1=*s*1-*s*2-*s*3.

* условие 2 (принадлежность области *М*2):

{входит в окружность с центром в точке (1;0)}

{входит в окружность с центром в точке (1;1)}

{не входит в окружность с центром в точке (0;0)}

{не входит в окружность с центром в точке (0;1)}

Площадь области *М*2:

2\*(площадь сектора в 60° - площадь равностороннего треугольника) + площадь квадрата -2\* площадь сектора в 30° = 0,6576

Аналитический вид функций, подлежащих интегрированию:

{верхняя полуокружность центральной окружности}; интервал интегрирования [ / 2, 1]; результат *s*1.

{нижняя полуокружность верхней центральной окружности}; интервал интегрирования [ / 2, 1];результат *s*2.

{верхняя полуокружность правой центральной окружности}; интервал интегрирования [1,  / 2]; результат *s*3.

{нижняя полуокружность правой верхней окружности}; интервал интегрирования [1,  / 2]; результат *s*4.

Площадь области *M*2=*s*3-*s*4+(*s*2 –*s*1).

* условие 3 (принадлежность области *М*3):

{входит в окружность с центром в точке (0;0)}

{не входит в окружность с центром в точке (1;1)}

{не входит в окружность с центром в точке (-1;1)}

{не входит в окружность с центром в точке (1; -1)}

{не входит в окружность с центром в точке (-1; -1)}

Площадь области *М*3:

(площадь квадрата – четверть площади круга) \* 4 = 0,8585

Аналитический вид функций, подлежащих интегрированию:

{нижняя полуокружность правой верхней окружности}; интервал интегрирования [0,1]; результат *s*1.

{нижняя полуокружность левой верхней окружности}; интервал интегрирования [-1,0]; результат *s*2.

{верхняя полуокружность правой нижней окружности}; интервал интегрирования [0,1]; результат *s*3.

{верхняя полуокружность левой нижней окружности}; интервал интегрирования [-1,0]; результат *s*4.

Площадь области *M*3= *s*1 + *s*2 + |*s*3| + |*s*4|.

* условие 4 (принадлежность области М4):

{входит в окружность с центром в точке (-1;-1)}

{не входит в окружность с центром в точке (-1;0)}

{не входит в окружность с центром в точке (0;-1)}

Площадь области *М4*:

3\*Площадь четверти круга – 2\*(площадь сектора 120 - площадь треугольник) = 1,128

Аналитический вид функций, подлежащих интегрированию:

{верхняя полуокружность левой центральной окружности}; интервал интегрирования [-2,-1]; результат *s*1.

{нижняя полуокружность левой нижней окружности}; интервал интегрирования [-2,-1]; результат *s*2.

{нижняя полуокружность верхней центральной окружности}; интервал интегрирования [-1, - / 2]; результат *s*3.

{верхняя полуокружность центральной окружности}; интервал интегрирования [-1, - / 2]; результат *s*4.

{горизонтальная прямая}; интервал интегрирования -2,-1]; результат *s*5.

Площадь области *M*4=|s2| – |s5| + 2(|s5| - |s1| - |s3-s4|.

* условие 5 (принадлежность области *М*5):

{правее линии х=1}

{левее линии х=2}

{выше линии у=-2}

{ниже линии у=-1}

Площадь области *М*5:

Площадь квадрата – четверть площади круга = 0,2146

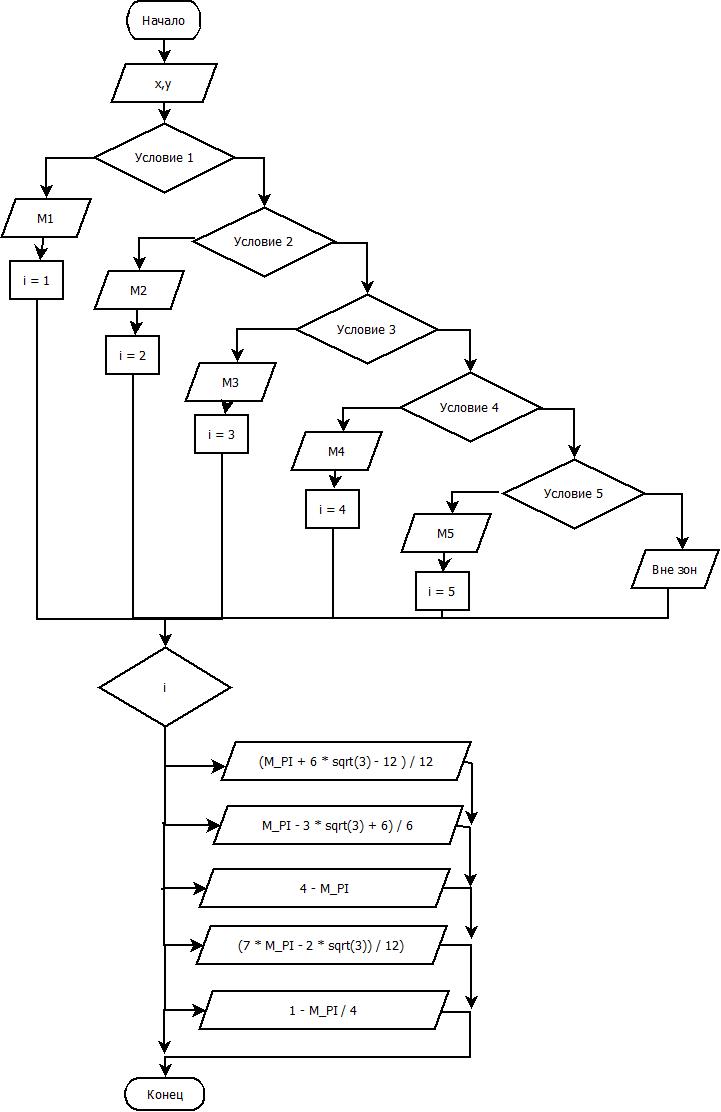
Аналитический вид функций, подлежащих интегрированию:

{нижняя полуокружность левой нижней окружности }; интервал интегрирования [1,2]; результат *s*1.

{горизонтальная прямая }; интервал интегрирования [1,2]; результат *s*2.

Площадь области *M*5=s2 –s1.

**Блок-схема алгоритма**



**Текст программы**

//Определение номера области, в которую попадает точка с произвольно заданными

//координатами на плоскости, и площади этой области

#include<iostream>

#include<math.h>

#include<locale.h>

using namespace std;

typedef double(\*Tfun)(double);

double integral(double a, double b, int n, double eps, Tfun fun, double s1)

{

double x,

h,

s = 0;

int i;

h = (b - a) / n;

for (i = 0; i < n; i++)

{

x = a + i \* h + h / 2;

s += fun(x);

}

s \*= h;

if (fabs(s - s1) > eps)

s = integral(a, b, 2 \* n, eps, fun, s);

return s;

}

double f1(double x)

{

return 1 - sqrt(1 - x \* x);

}

double f2(double x)

{

return sqrt(1 - x \* x);

}

double f3(double x)

{

return sqrt(1 - (x+1) \*( x+1));

}

double f4(double x)

{

return sqrt(1 - (x - 1) \* (x - 1));

}

double f5(double x)

{

return 1 - sqrt(1 - (x - 1) \* (x - 1));

}

double f6(double x)

{

return - sqrt(1 - (x + 1) \* (x + 1)) -1;

}

double f9(double x)

{

return -1;

}

double f10(double x)

{

return -sqrt(1 - (x - 1) \* (x - 1)) - 1;

}

double f11(double x)

{

return -2;

}

double f12(double x)

{

return 1 - sqrt(1 - x \* x);

}

double f13(double x)

{

return -1 + sqrt(1 - (x + 1) \* (x + 1));

}

double f14(double x)

{

return -1 + sqrt(1 - (x - 1) \* (x - 1));

}

double f15(double x)

{

return 1 - sqrt(1 - (x + 1) \* (x + 1));

}

int main()

{

int i = 0; //номер области на рисунке

float x, y, //координаты точки

s1, s2, s3, s4, s5; //площади областей

float M\_PI = 3.1415;

double eps = .001, //точность вычисления

s = 0; //вычисленная площадь

setlocale(LC\_ALL, "Russian");

cout << "Установленная точность: "<< eps << "\nВведите координаты точки: x, y ";

cin >> x >> y;

if (x \* x + (y - 1) \* (y - 1) < 1 && (x + 1) \* (x + 1) + y \* y < 1 && x \* x + y \* y > 1)

{

cout << "Точка в области M1. "; i = 1;

}

else if ((x - 1) \* (x - 1) + y \* y < 1 && (x - 1) \* (x - 1) + (y - 1) \* (y - 1) < 1 && x \* x + y \* y > 1 && x \* x + (y - 1) \* (y - 1) > 1)

{

cout << "Точка в области М2. "; i = 2;

}

else if (x \* x + y \* y < 1 && (x - 1) \* (x - 1) + (y - 1) \* (y - 1) > 1 && (x + 1) \* (x + 1) + (y - 1) \* (y - 1) > 1 && (x + 1) \* (x + 1) + (y + 1) \* (y + 1) > 1 && (x - 1) \* (x - 1) + (y + 1) \* (y + 1) > 1)

{

cout << "Точка в области М3. "; i = 3;

}

else if ((x + 1) \* (x + 1) + (y + 1) \* (y + 1) < 1 && (x + 1) \* (x + 1) + y \* y > 1 && x \* x + (y + 1) \* (y + 1) > 1)

{

cout << "Точка в области М4. "; i = 4;

}

else if (x > 1 && x < 2 && y < -1 && y > -2 && (x - 1) \* (x - 1) + (y + 1) \* (y + 1) > 1)

{

cout << "Точка в области М5. "; i = 5;

}

else cout << "Точка вне выделенных областей";

cout.precision(4); //число знаков после дес. точки

switch (i)

{

case 1: cout << "S1 = " << (s1 = (M\_PI + 6 \* sqrt(3) - 12) / 12) << endl << "Integral:S1 = " << integral(-1, -0.5, 10 , eps, f3, s) - integral(-sqrt(3) / 2, -0.5, 10, eps, f2, s) - integral(-1, -sqrt(3)/2, 10, eps, f12, s);

break;

case 2: cout << "S2 = " << (s2 = (M\_PI - 3 \* sqrt(3) + 6) / 6) << endl << "Integral:S2 = " << integral(1, 1+ sqrt(3) / 2, 10, eps, f4, s) - integral(1, 1 + sqrt(3) / 2, 10, eps, f5, s) + (integral(sqrt(3) / 2, 1, 10, eps, f1, s) - integral(sqrt(3) / 2, 1, 10, eps, f2, s));

break;

case 3: cout << "S3 = " << (s3 = 4 - M\_PI) << endl << "Integral:S3 = " << integral(-1, 0, 10, eps, f15, s) + integral(0, 1, 10, eps, f5, s) + fabs(integral(-1, 0, 10, eps, f13, s) + integral(0, 1, 10, eps, f14, s));

break;

case 4: cout << "S4 = " << (s4 = 3 \* M\_PI / 4 - 2\*(M\_PI / 3 - sqrt(3) / 4)) << endl << "Integral:S4 = " << fabs(integral(-2, -1, 10, eps, f6, s)) - fabs(integral(-2, -1, 10, eps, f9, s)) + 2 \* (fabs(integral(-2, -1, 10, eps, f9, s)) - fabs(integral(-2, -1, 10, eps, f3, s)) - (integral(-1, -sqrt(3) / 2, 10, eps, f1, s) - integral(-1, -sqrt(3) / 2, 10, eps, f2, s)));

break;

case 5: cout << "S5 = " << (s5 = 1 - M\_PI / 4) << endl << "Integral:S5 = " <<fabs(integral(1, 2, 10, eps, f11, s)) - fabs(integral(1, 2, 10, eps, f10, s));

}

cout << "\n Повторить-1, Выход-2: ";

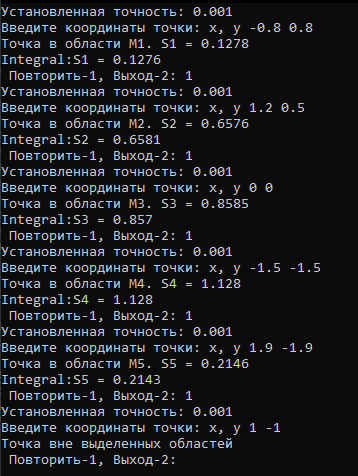
cin >> i;

if (i == 1) main();

return 0;

}

**Скриншот примера работы программы**

****

**Вывод:** используя технологию процедурного программирования реализовал заданный метод нахождения значения определённого интеграла произвольной функции на заданном интервале и с заданной точностью и применил его для определения площадей областей на плоскости в соответствии с индивидуальным заданием.