Министерство образования и науки России

Федеральное государственное автономное образовательное

учреждение высшего образования

“Санкт-Петербургский государственный электротехнический

университет им. В. И. Ульянова (Ленина)”

(СПБГЭТУ “ЛЭТИ”)

Факультет компьютерных технологий и информатики

Кафедра вычислительной техники

**ОТЧЕТ**

**по лабораторной работе № 1 на тему**

**“Условные операторы и циклы языка С / C++” по**

**дисциплине “Программирование”**

**Выполнил:** *Студент группы 5307 Грицай Н. Д.*

**Проверила:** *к.т.н., доцент Сискович Т.И.*

**Подпись преподавателя:\_\_\_\_\_\_\_\_**

Санкт-Петербург

2015

**Цель работы**

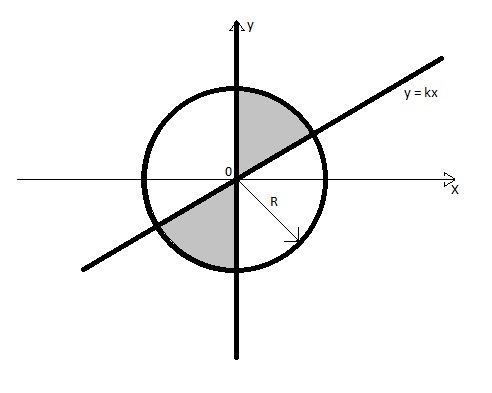
Получить практические навыки использования условных операторов и циклов языка С / C++.

**Задание**

Разработать алгоритм и написать программу, выполняющую многократно по желанию пользователя ввод координат точки на плоскости и определение: лежит точка в области, за пределами области или на ее границе. Область указана на рисунке 1. В программе предусмотреть изменение параметров, задающих область.

**Уточнение задания**

При выполнении программы пользователь должен ввести коэффициент прямой y=kx, радиус окружности R и координаты точки x, y. После выполнения пользователь может изменить координаты точки или коэффициенты уравнений. Область указана на рисунке 1 и ограничена окружностью, прямой y = kx и осью у. Радиус может быть равен нулю.

****

*Рисунок 1. Заданная область*

**Контрольные примеры**

Контрольные примеры представлены в таблице 1.

*Таблица 1. Контрольные примеры*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ примера** | **Исходные данные** | | | | **Результат** |
| **k** | **R** | **x** | **y** |
| 1 | 1.6 | 1 | 0 | 0 | “Точка лежит на границе” |
| 2 | 0 | 5 | 1 | 1 | “Точка лежит в области” |
| 3 | -42.8 | 1 | 0 | -2 | “Точка лежит вне области” |
| 4 | 0 | 10 | 1 | 1 | “Точка лежит в области” |
| 5 | -9.7 | 7.8 | -1 | -1 | “Точка лежит в области” |
| 6 | 0 | 5 | 3 | 4 | “Точка лежит на границе” |
| 7 | 0 | 0 | 0 | 0 | “Точка лежит на границе” |
| 8 | 14 | 0 | 5 | -7 | “Точка лежит вне области” |
| 9 | 6.6 | 100 | 2.8 | 4.9 | “Точка лежит в области” |
| 10 | 1.9 | 6.9 | 9 | 6 | “Точка лежит вне области” |

**Описание переменных**

Описание переменных представлено в таблице 2.

*Таблица 2. Описание переменных*

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Назначение** |
| x, y | float | Координаты точки |
| k | float | Коэффициент k прямой y = kx |
| R | float | Радиус окружности |
| result | short | Результат(0 – точка лежит вне области, 1- точка лежит в области, 2 – точка лежит на границе) |
| newcoefficients, newcoordinates | short | Переменные повтора цикла(1 – повторить цикл, 0 – выйти из него) |

**Краткое описание алгоритма**

1. Ввод k
2. Ввод R. Если R < 0, то: переход к пункту 2
3. Ввод x, y
4. Вывод значения переменных x, y, k, R
5. Если *(x\*x + y\*y < R\*R) && (((y > k\*x) && (x > 0)) || ((y < k\*x) && (x < 0)))*, то: result = 1 и переход в пункт 7, иначе: переход в пункт 6
6. Если *((x\*x + y\*y <= R\*R)) && ((y == k\*x) || (y == 0)) || ((x\*x + y\*y) == R\*R) && (((x >= 0) && (y >= k\*x)) || ((x < 0) && (y <= k\*x)))*, то: result = 2.
7. Вывод результата
8. Ввод newcoordinates
9. Если newcoordinates != 0, то: переход в пункт 2.
10. Ввод newcoefficients
11. Если newcoefficients != 0, то: переход в пункт 1, иначе: выход из программы.

**Схема алгоритма**

Схема алгоритма представлена в приложении 1

**Исходный код программы**

#include "stdafx.h"

#include "iostream"

#include "Windows.h"

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

float x, y, k, R;

short newcoefficient, newcoordinates, result;

//главный цикл

do

{

printf\_s("Введите k: ");

scanf\_s("%f", &k);

printf\_s("Введите R: ");

do

{

scanf\_s("%f", &R);

if (R < 0)

printf("R не может быть отрицательным.\nПовторите ввод R: ");

} while (R < 0);

do

{

printf\_s("Введите x: ");

scanf\_s("%f", &x);

printf\_s("Введите y: ");

scanf\_s("%f", &y);

system("cls");

result = 0;

//вывод парраметров

printf\_s("коэффициент k = %.2f , R = %.2f\nкоординаты точки = ( %.2f

; %.2f )\n", k, R, x, y);

//решение

if ((x\*x + y\*y < R\*R) && (((y > k\*x) && (x > 0)) || ((y < k\*x) && (x

< 0))))

result = 1;

else

if (((x\*x + y\*y <= R\*R)) && ((y == k\*x) || (y == 0)) || ((x\*x

+ y\*y) == R\*R) && (((x >= 0) && (y >= k\*x)) ||

((x < 0) && (y <= k\*x))))

result = 2;

//вывод

if (result == 1)

printf\_s("\nТочка лежит в области.\n");

else

if (result == 2)

printf\_s("\nТочка лежит на границе.\n");

else

printf\_s("\nТочка лежит вне области.\n");

//

printf\_s("\nИзменить координаты точки?(Да-1/Нет-0): ");

scanf\_s("%hi", &newcoordinates);

} while (newcoordinates);

//

printf\_s("\nИзменить коэффициенты уравнения?(Да-1/Нет-0): ");

scanf\_s("%hi", &newcoefficient);

} while (newcoefficient);

return 0;

}

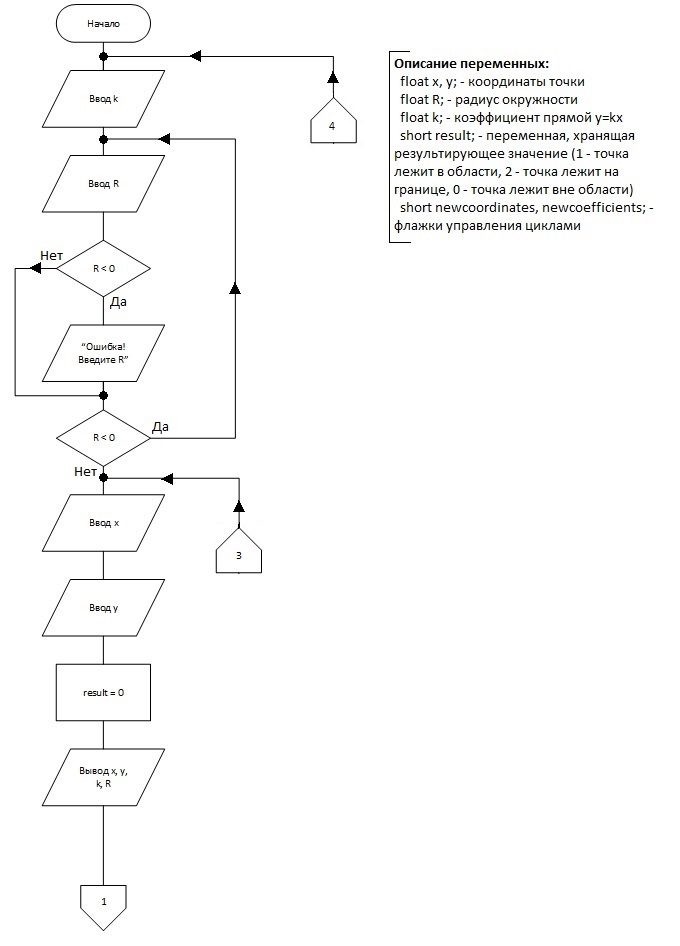
**Результат**

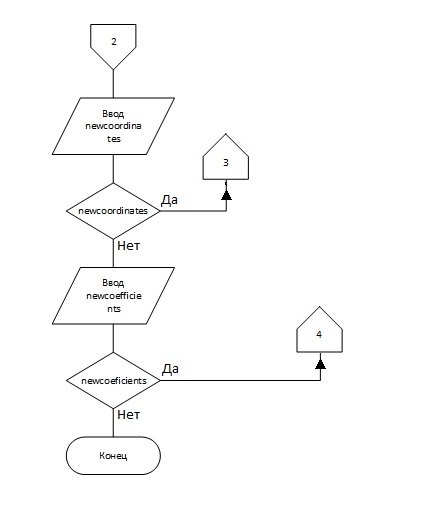
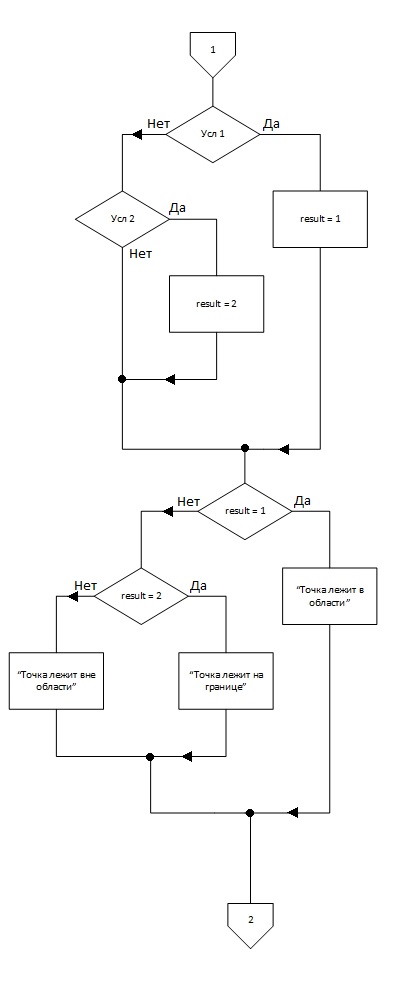
При выполнении программы получены результаты, совпадающие с контрольными примерами, представленными в таблице 1. Ошибки не обнаружены.

**Вывод**

В ходе выполнения работы были получены практические навыки использования условных операторов и циклов языка С / C++.

*Приложение. Схема программы*

****

****

*Рисунок 2. Схема программы*

Усл1: *((x\*x + y\*y < R\*R) && (((y > k\*x) && (x > 0)) || ((y < k\*x) && (x < 0))))*

Усл2: *(((x\*x + y\*y <= R\*R)) && ((y == k\*x) || (y == 0)) || ((x\*x + y\*y) == R\*R) && (((x*

*>= 0) && (y >= k\*x)) || ((x < 0) && (y <= k\*x))))*