МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Воронежский государственный технический университет»

**Факультет** экономики, менеджмента и информационных технологий

**Кафедра** систем управления и информационных технологий в строительстве

**Отчет по лабораторной работе**

Тема: **«Реализация разветвляющихся алгоритмов»**

По дисциплине: Основы программирования и алгоритмизации

Выполнил студент: Гладнева Евгения Геннадьевна

Группа: бИСТ-222

Руководитель: доцент, к.т.н. Курипта О. В.

Работа защищена « » \_\_\_\_\_\_\_\_\_ 2022г.

С оценкой \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Воронеж 2022

**Постановка задачи**

**Условие задачи:** Написать программу, которая по введенной пользователем координате точки A(x,y) определяет, к какой четверти она относиться.

**Исходные данные:**

x, y.

**Алгоритм решения:**

1. Необходимо запросить у пользователя координаты x, y.
2. Определить, не лежит ли точка на какой-либо оси, т.к. в таком случае она не будет иметь четверти. Точка лежит на оси, если хотя бы одна координата равна 0.
3. Определить четверть, в которой находится точка, учитывая, что:
   1. I четверть: x ∈ (0, +∞), y ∈ (0, +∞)
   2. II четверть: x ∈ (-∞, 0), y ∈ (0, +∞)
   3. III четверть: x ∈ (-∞, 0), y ∈ (-∞, 0)
   4. IV четверть: x ∈ (0, +∞), y ∈ (-∞, 0)

**Контрольный пример №1**:

Входные данные: 0, 20

Результат: Точка А(0, 20) лежит на оси

**Контрольный пример №2**:

Входные данные: 234, -45

Результат: Точка (234, -45) находится в 4 четверти

**Словесный алгоритм**

**Алгоритм представлен пошаговой детализацией:**

**Шаг 1:** установка директивы препроцессора для функции scanf, подключение необходимых заголовочных файлов

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <locale.h>

**Шаг 2:** объявление функции main

void main(void) {};

**Шаг 3:** установка локализации

setlocale(LC\_ALL, "RUS");

**Шаг 4:** объявление переменных

double x, y;

**Шаг 5:** ввод значений пользователем

puts("Введите координаты точки A");

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

**Шаг 6:** проверка точки на нахождение на оси и вывод результата при необходимости

if (x == 0 || y == 0) printf("Точка A(%lg, %lg) лежит на оси", x, y);

**Шаг 7:** определение четверти в отдельной функции

int checking(double x, double y)

{

if (x > 0 && y > 0) return 1;

else {

if (x < 0 && y > 0) return 2;

else {

if (x < 0 && y < 0) return 3;

else return 4;

}

}

};

**Шаг 8:** вывод результата

else {

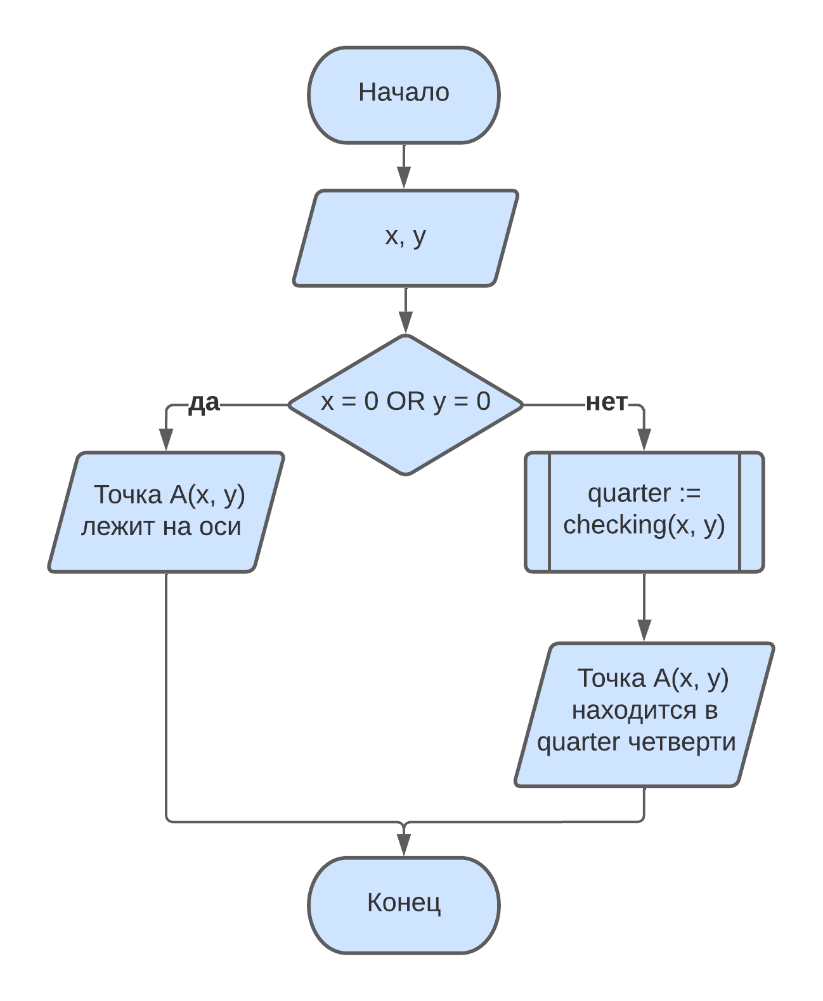
int quarter = checking(x, y);

printf("Точка A(%lg, %lg) находится в %d четверти", x, y, quarter);

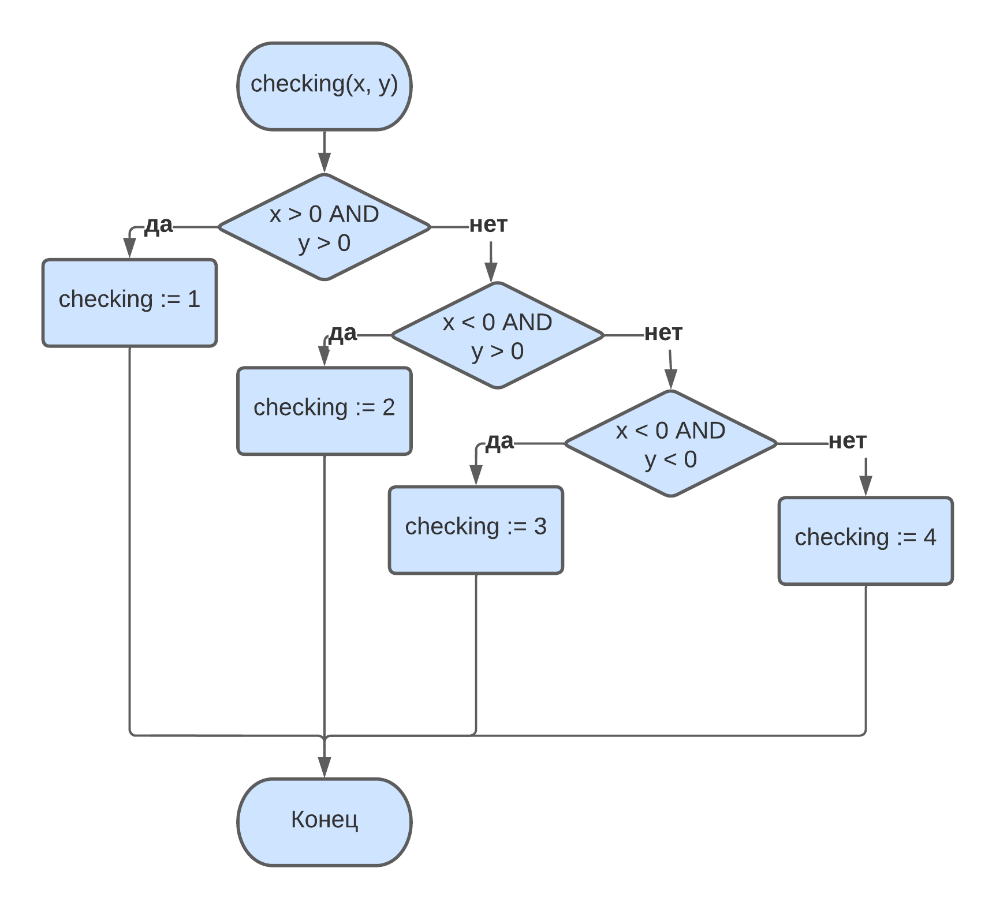
};

**Блок – схема программы**

Блок – схема программы, которая определяет четверть, в которой расположена точка, по её координатам, представлена на рисунке 1.



а)



б)

а) — блок-схема головного алгоритма

б) — блок-схема функции checking

Рисунок 1 — Блок-схема программы

**Результат работы программы**

На рисунке 2 представлено окно ввода данных

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2 — Окно ввода данных

На рисунке 3 представлена работа программы с данными из контрольного примера №1.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3 — Окно с выполненной программой

На рисунке 4 представлена работа программы с данными из контрольного примера №2.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4 — Окно с выполненной программой

**Код программы**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <locale.h>

int checking(double x, double y)

{

if (x > 0 && y > 0) return 1;

else {

if (x < 0 && y > 0) return 2;

else {

if (x < 0 && y < 0) return 3;

else return 4;

}

}

};

void main(void)

{

setlocale(LC\_ALL, "RUS");

double x, y;

puts("Введите координаты точки A");

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

if (x == 0 || y == 0) printf("Точка A(%lg, %lg) лежит на оси", x, y);

else {

int quarter = checking(x, y);

printf("Точка A(%lg, %lg) находится в %d четверти", x, y, quarter);

};

}