

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМ. Р.Е. АЛЕКСЕЕВА

ИНСТИТУТ РАДИОЭЛЕКТРОНИКИ И ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ

Кафедра «Информационные радиосистемы»

КУРСОВАЯ РАБОТА

***«Из набора точек на декартовой плоскости сформировать новый
набор, в который включаются только те точки, которые находятся вне
окружности с центром (x_C , y_C) и радиусом R .»***

Выполнил:

студент группы 24-Рз И. И. Иванов

Проверил:

доцент кафедры ИРС С. Б. Сидоров

Нижний Новгород

2025

Содержание

Введение.....	3
1. Постановка задачи.....	4
2. Руководство пользователя.....	5
3 Руководство программиста.....	7
3.1 Структура программы.....	7
3.2 Структура данных.....	7
3.3 Алгоритм вхождений точек в окружность.....	8
Заключение.....	10
Список литературы.....	11

Введение

В работе рассматривается решение задачи разработки прикладной программы поиска всех изданий указанного жанра, вышедшие в указанный период времени.

Одним из способов преодоления сложности данной задачи является использование отдельной функции для выполнения обработки. Этот подход позволяет свести задачу к последовательному решению более простых задач. Его использование позволяет уменьшить затраты на отладку и повысить надёжность программы. Также существенным моментом является использование структурных типов данных для адекватного отображения сущностей проблемной области в область программной реализации.

В начале основной части отчёта приводится точная формализованная постановка задачи с указанием полного набора операций, выполнение которых должна обеспечивать прикладная программа.

В руководстве пользователя раскрывается назначение программы, её возможности и выполняемые операции. Подробно объясняются правила пользования программой и приводятся конкретные примеры диалога с пользователем.

В руководстве программиста рассматриваются вопросы внутренней организации программы, в том числе перечень функций и их взаимодействие. Кроме этого описываются используемые структуры данных и наиболее важные, и интересные с точки зрения их реализации алгоритмы.

В заключении делаются выводы о полноте решения поставленной задачи, а также приводится ряд соображений по возможным направлениям доработки полученной прикладной программы. Также приведена техническая информация, включающая листинги программы.

1. Постановка задачи

Рассматривается модель информации о точках на декартовой плоскости и окружностях. Описание точки представлено совокупностью свойств: координата_x, координата_y. В то же время, описание окружности представлено координатами центра x/y и радиусом. В области программной реализации модель точки и окружности имеет вид структурного типа данных.

Рассматривается набор точек с конечным количеством элементов. Требуется получить программную реализацию заданной обработки такого набора структурированных данных.

Все данные, необходимые для обработки, запрашиваются у пользователя: значения элементов, входящих в набор, и дополнительная информация, необходимая для выполнения конкретной обработки данных.

Обработка должна быть реализована отдельной функцией. При этом вся необходимая для выполнения обработки информация должна передаваться в функцию через список аргументов. Результат обработки набора структурированных данных должен передаваться из функции также через список аргументов.

Полученные результаты обработки должны быть выданы на монитор, то есть на стандартное устройство вывода.

Задана определенная окружность и набор точек. Требуется найти все точки за пределами окружности, и вывести их координаты на стандартное устройство.

2. Руководство пользователя

Программа предназначена для определения точек, выходящих за предел окружности, в режиме диалога с пользователем.

Программа позволяет пользователю задать координаты точек, и параметры окружности – координаты центра и радиус. Результаты обработки выводятся на экран монитора. Через консоль (терминал).

Запуск программы осуществляется либо через командную строку с вводом имени исполняемого файла, полученного в результате компиляции, с последующим нажатием клавиши *Enter*, либо иным способом в зависимости от операционной системы.

Программа является интерактивным консольным приложением. Весь диалог с пользователем осуществляется в текстовом режиме.

Сначала пользователю предлагается указать параметры окружности – координаты x/y , ввод в виде: “ $x\ y$ ”, радиус:

- Enter the circle's center (x_c, y_c):
- Enter the circle's radius (r):

После этого у пользователя запрашивается количество точек в наборе:

- Enter the number of points:

Далее для каждой точки из набора будут запрошены координаты x/y , ввод в виде: “ $x\ y$ ”.

После ввода набора точек выполняется поиск вхождений в окружность. По завершении обработки выводится результат следующего вида:

Points outside the circle:

- (x, y)
- (x, y)

Если подходящих точек не найдено, выводится соответствующее сообщение:

- No points are outside the circle.

По завершении выдачи результатов программа завершает свою работу.

3 Руководство программиста

3.1 Структура программы

Прикладная программа разработана с использованием принципов императивного программирования. Она является совокупностью взаимодействующих функций. Структура программы представлена на рис. 1.

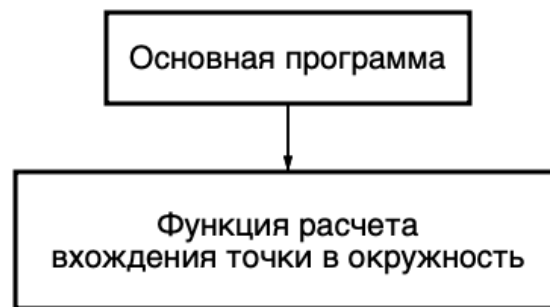


Рисунок 1. Структура программы

Программа состоит из двух функций, назначение каждой из которых приведено ниже:

1. `main` – основная функция приложения
2. `is_outside_circle` – функция расчета вхождения точки в окружность

3.2 Структура данных

Для описания окружности определяем структурный тип данных `circle_t`. Чтобы задать окружность, достаточно трех параметров – координаты `x/y` центра и радиус. Таким образом получаем следующее определение структурного типа данных, для описания понятия “Окружность”:

```
typedef struct {  
    double x;  
    double y;
```

```
double radius;  
  
} circle_t;
```

Для описания точки определяем структурный тип данных `point_t`. Из постановки задачи явно следует, что каждая точка задается двумя координатами: x , y . Каждой координате соответствует свое поле в структурном типе данных.

Таким образом получаем следующее определение структурного типа данных, для описания пункта “Точка”:

```
typedef struct {  
  
    double x;  
    double y;  
} point_t;
```

Для представления набора точек используем массив из элементов `point_t` `points[num_points]`. Максимально допустимое количество элементов в массиве ограничим переменной `num_points`, ее значение запрашивается у пользователя.

Так же, для хранения входящих в окружность точек, вводим второй массив из элементов типа `point_t` `outside_points[num_points]`. Так как, количество точек вне окружности не может быть больше общего количества точек, то ограничить массив для хранения, так же, можно используя переменную `num_points`.

3.3 Алгоритм вхождения точек в окружность

Определение вхождения точки в окружность сводится к расчету расстояния к точке из центра окружности и сравнению его с радиусом окружности, соответственно, проверке логического выражения для каждой точки из набора:

$$((\text{точка_}x - \text{окружность_}x)^2 + (\text{точка_}y - \text{окружность_}y)^2) > \text{окружность_радиус}^2$$

Если текущая точка удовлетворяет условию, значение логического выражения – ИСТИНА, в таком случае точка заносится в массив `outside_count`.

Заключение

В данной работе задача разработки прикладной программы поиска точек вне окружности, была решена с использованием императивного программирования. На основании проведённой отладки и испытаний с помощью контрольных примеров можно сделать вывод, что полученная прикладная программа решает поставленную задачу правильно и в полном объёме.

Список литературы

1. Керниган, Брайан У., Ритчи, Деннис М. Язык программирования С, 2-е издание. [пер. с англ.] / Б.У. Керниган, Д.М. Ритчи – М.: Вильямс, 2007.
2. Павловская, Т.А. С/С++. Программирование на языке высокого уровня: учебник для ВУЗов / Т.А. Павловская. – СПб.: Питер, 2009.
3. Орлов, С.А. Технологии разработки программного обеспечения. учеб. пособие. 2-е изд./ С.А. Орлов, – СПб.: Питер, 2003. – 480 с.: ил.
4. Борисенко, В.В. Основы программирования / В.В.Борисенко, – Интернет-университет информационных технологий – ИНТУИТ.ру, 328 стр. – 2005 г.
5. Шилдт, Г. Полный справочник по С: учеб. пособие / Г. Шилдт. – 4-е изд. – М.: Изд. дом "Вильямс", 2008.
6. Костюкова, Н.И. Язык Си и особенности работы с ним / Н.И. Костюкова, Н.А. Калинина – Интернет-университет информационных технологий – ИНТУИТ.ру, 208 стр. – 2006 г.

Приложение А

Заголовочный файл

```
#ifndef CIRCLE_H
#define CIRCLE_H

#include <stdio.h>
#include <stdbool.h>

typedef struct {
    double x;
    double y;
} point_t;

typedef struct {
    double x;
    double y;
    double radius;
} circle_t;

bool is_outside_circle(point_t point, circle_t circle);

#endif // CIRCLE_H
```

Основная программа

```
#include "circle.h"

bool is_outside_circle(point_t point, circle_t circle) {
    double distance_squared = (point.x - circle.x) *
        (point.x - circle.x) + (point.y - circle.y) * (point.y -
        circle.y);
    if (distance_squared > circle.radius * circle.radius)
    {
        return true;
    } else {
        return false;
    }
}

int main() {
    circle_t circle;
    printf("Enter the circle's center (xc, yc): \n");
    scanf("%lf %lf", &circle.x, &circle.y);
    printf("Enter the circle's radius (r): \n");
    scanf("%lf", &circle.radius);

    int num_points;
    printf("Enter the number of points: \n");
    scanf("%d", &num_points);
    point_t points[num_points];
    printf("Enter the points (x y):\n");
    for (int i = 0; i < num_points; i++) {
        scanf("%lf %lf", &points[i].x, &points[i].y);
    }
}
```

```

}

point_t outside_points[num_points];
int outside_count = 0;

for (int i = 0; i < num_points; i++) {
    if (is_outside_circle(points[i], circle)) {
        outside_points[outside_count++] = points[i];
    }
}

if (outside_count == 0) {
    printf("No points are outside the circle.\n");
} else {
    printf("Points outside the circle:\n");
    for (int i = 0; i < outside_count; i++) {
        printf("(%.2lf, %.2lf)\n",
            outside_points[i].x, outside_points[i].y);
    }
}

return 0;
}

```