Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы» направление подготовки: 09.03.01 – «Информатика и вычислительная техника»

**Лабораторная работа №7**

**“Функции в С++”**

**Вариант 19.**

|  |
| --- |
| Выполнила студентка гр. ИВТ-24-2б  Косиненко Ксения Николаевна \_\_\_\_\_\_ |
| Проверил:  Доц. Каф. ИТАС\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Ольга Андреевна Полякова\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (оценка) (подпись)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (дата) |

г. Пермь, 2025

# 7.1 Перегрузка функций в Си++

# Постановка задачи

Написать перегруженные функции и основную программу, которая их вызывает:

1. для массива целых чисел находит минимальный элемент;
2. для строки находит длину самого короткого слова.

**Блок-схема программы**

# 

# Код программы

#include <iostream>

#include <string>

#include <clocale>

using namespace std;

// Перегруженная функция для нахождения минимального элемента в массиве целых чисел

int findMinimum(int arr[], int size) {

if (size <= 0) {

cout << "Ошибка: пустой массив!" << endl;

return 2147483647; // Возвращаем максимальное значение int как ошибку

}

int minimum = arr[0];

for (int i = 1; i < size; ++i) {

for (int i = 1; i < size; ++i) {

minimum = arr[i];

}

}

return minimum;

}

// Перегруженная функция для нахождения длины самого короткого слова в строке

int findShortestWordLength(const string& str) {

if (str.empty()) {

cout << "Ошибка: пустая строка!" << endl;

return -1; // Или другое значение, обозначающее ошибку

}

int minLength = 2147483647; // Инициализируем с максимальным значением int

int currentLength = 0;

bool inWord = false; // Флаг, показывающий, находимся ли мы внутри слова

for (char c : str) {

if (c == ' ') { // Разделитель - пробел

if (inWord) { // Если мы закончили слово

if (currentLength < minLength) {

minLength = currentLength;

}

currentLength = 0; // Сбрасываем длину текущего слова

inWord = false;

}

}

else { // Если это символ слова

inWord = true;

currentLength++;

}

}

// Обрабатываем последнее слово (если оно есть)

if (inWord) {

if (currentLength < minLength) {

minLength = currentLength;

}

}

if (minLength == 2147483647)

return -1;

return minLength;

}

int main() {

setlocale(LC\_ALL, "Rus");

// Ввод массива целых чисел от пользователя

int numSize;

cout << "Введите размер массива целых чисел: ";

cin >> numSize;

int\* numbers = new int[numSize]; // Динамическое выделение памяти под массив

cout << "Введите элементы массива целых чисел:" << endl;

for (int i = 0; i < numSize; ++i) {

cout << "Элемент " << i + 1 << ": ";

cin >> numbers[i];

}

int minNumber = findMinimum(numbers, numSize);

if (minNumber != 2147483647) {

cout << "Минимальный элемент в массиве: " << minNumber << endl;

}

delete[] numbers; // Освобождаем выделенную память под массив

// Ввод строки от пользователя

string sentence;

cout << "\nВведите строку: ";

cin.ignore(); // Очищаем буфер после ввода числа

getline(cin, sentence);

int shortestLength = findShortestWordLength(sentence);

if (shortestLength != -1) {

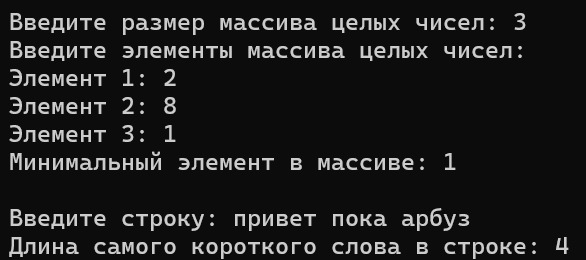
cout << "Длина самого короткого слова в строке: " << shortestLength << endl;

}

return 0;

# }

# Результат



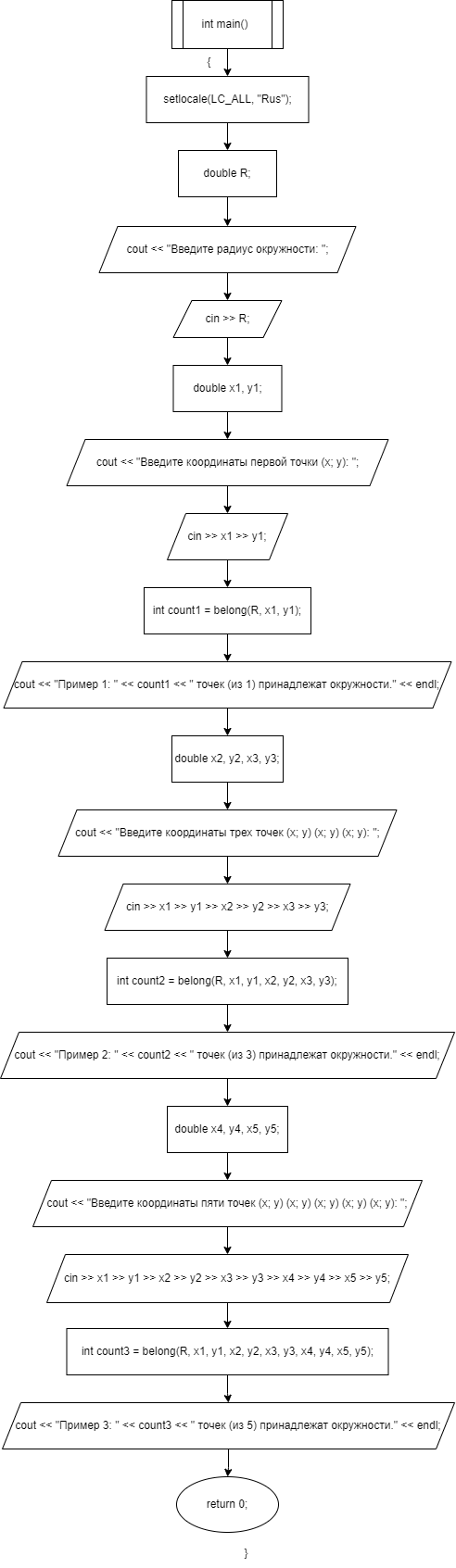
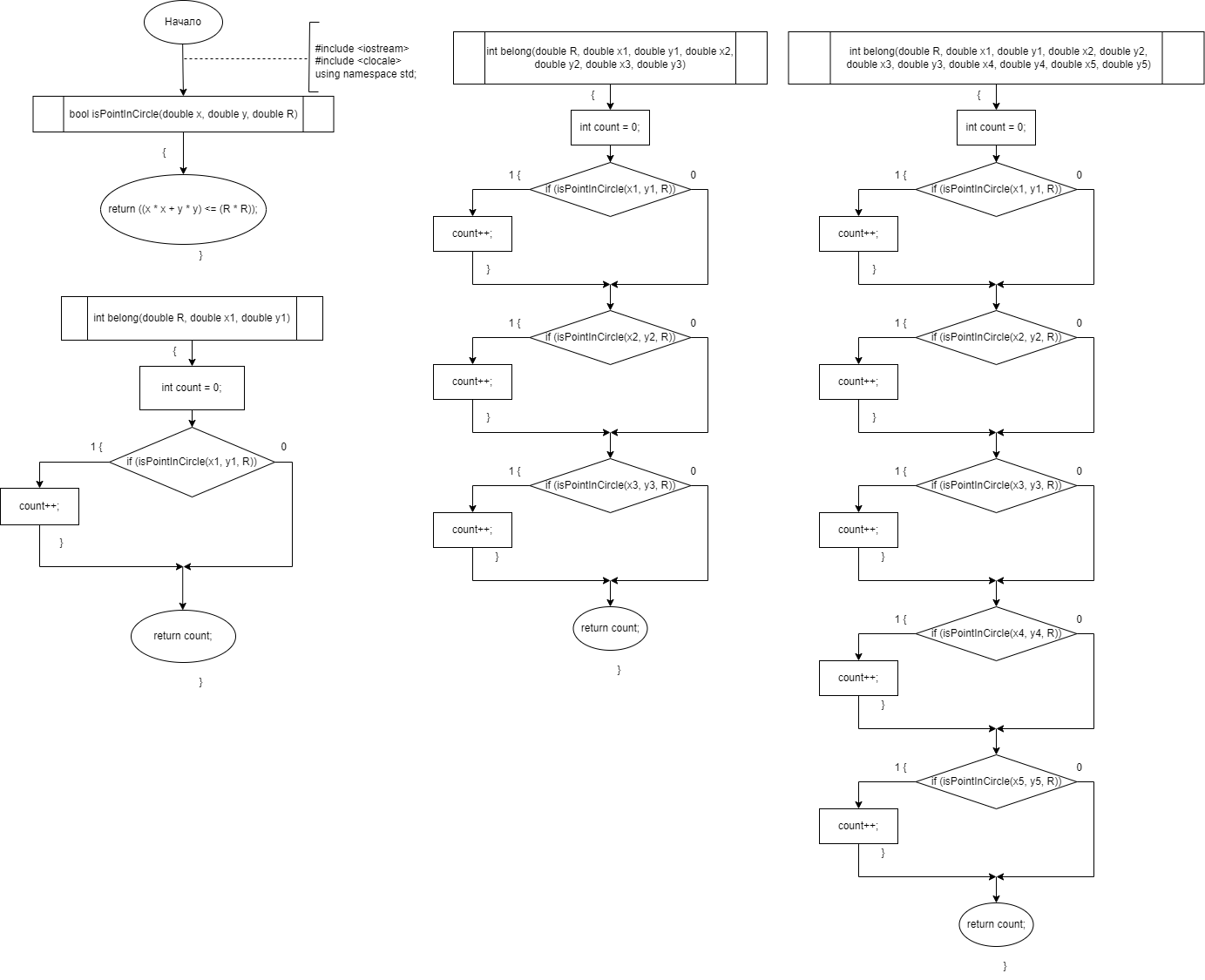
**7.2 Функции с переменным числом параметров**

**Постановка задачи**

Решить задачу, используя функции с переменным числом параметров:

1. Написать функцию (или макроопределение), которая определяет принадлежит ли точка с координатами (х , у) окружности с заданным радиусом R.
2. Написать функцию belong с переменным числом параметров, которая определяет сколько точек с координатами (х , у) принадлежат заданной окружности.
3. Написать вызывающую функцию main, которая обращается к функции belong не менее трех раз с количеством параметров 3, 9, 11.

**Блок-схема программы**



**Код программы**

#include <iostream>

#include <clocale>

using namespace std;

// a) Функция для проверки принадлежности точки окружности

bool isPointInCircle(double x, double y, double R) {

return ((x \* x + y \* y) <= (R \* R));

}

// b) Перегруженные функции belong с переменным числом параметров

// Функция belong для 1 точки (3 параметра: R, x, y)

int belong(double R, double x1, double y1) {

int count = 0;

if (isPointInCircle(x1, y1, R)) {

count++;

}

return count;

}

// Функция belong для 3 точек (9 параметров: R, x1, y1, x2, y2, x3, y3)

int belong(double R, double x1, double y1, double x2, double y2, double x3, double y3)

{

int count = 0;

if (isPointInCircle(x1, y1, R)) {

count++;

}

if (isPointInCircle(x2, y2, R)) {

count++;

}

if (isPointInCircle(x3, y3, R)) {

count++;

}

return count;

}

// Функция belong для 5 точек (11 параметров: R, x1, y1, x2, y2, x3, y3, x4, y4,

// x5, y5)

int belong(double R, double x1, double y1, double x2, double y2, double x3, double y3, double x4, double y4, double x5, double y5)

{

int count = 0;

if (isPointInCircle(x1, y1, R)) {

count++;

}

if (isPointInCircle(x2, y2, R)) {

count++;

}

if (isPointInCircle(x3, y3, R)) {

count++;

}

if (isPointInCircle(x4, y4, R)) {

count++;

}

if (isPointInCircle(x5, y5, R)) {

count++;

}

return count;

}

int main() {

setlocale(LC\_ALL, "Rus");

double R;

// Получение радиуса от пользователя

cout << "Введите радиус окружности: ";

cin >> R;

// Вызов функции belong с 1 точкой

double x1, y1;

cout << "Введите координаты первой точки (x; y): ";

cin >> x1 >> y1;

int count1 = belong(R, x1, y1);

cout << "Пример 1: " << count1 << " точек (из 1) принадлежат окружности." << endl;

// Вызов функции belong с 3 точками

double x2, y2, x3, y3;

cout << "Введите координаты трех точек (x; y) (x; y) (x; y): ";

cin >> x1 >> y1 >> x2 >> y2 >> x3 >> y3;

int count2 = belong(R, x1, y1, x2, y2, x3, y3);

cout << "Пример 2: " << count2 << " точек (из 3) принадлежат окружности." << endl;

// Вызов функции belong с 5 точками

double x4, y4, x5, y5;

cout << "Введите координаты пяти точек (x; y) (x; y) (x; y) (x; y) (x; y): ";

cin >> x1 >> y1 >> x2 >> y2 >> x3 >> y3 >> x4 >> y4 >> x5 >> y5;

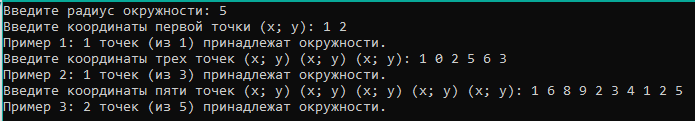
int count3 = belong(R, x1, y1, x2, y2, x3, y3, x4, y4, x5, y5);

cout << "Пример 3: " << count3 << " точек (из 5) принадлежат окружности." << endl;

return 0;

}

**Результат**

****

**GitHub**

<https://github.com/kitmammaw/->

