МИНИСТЕРСТВО НАУКИ и высшего образования

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

**Институт среднего профессионального образования**

**Отчёт по лабораторной работе № 2**

**по учебной дисциплине «Основы алгоритмизации и программирования»**

**Тема: "Программирование алгоритмов разветвленной структуры"**

Выполнил студент

специальности 09.02.07

Информационные системы и

программирование

II курса группы 22919/9

Сараев Никита

Александрович

Преподаватель

Молькова Лолита Юрьевна

Санкт-Петербург,

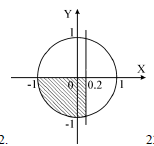
2022

**Цель работы:**

-Изучение условного оператора и приобретение навыков программирования разветвляющихся алгоритмов.

**Задание:** по заданным координатам точки определить, где находится точка и выбрать правильный вариант ответа.

Заданная область:

****

Варианты ответа для кода:

1. Внутри заштрихованной области
2. Вне заштрихованной области
3. На границе области

**Ход работы:**

Блок-схема приложения с использованием условного оператора if else со всеми возможными исходами (рис. 1)

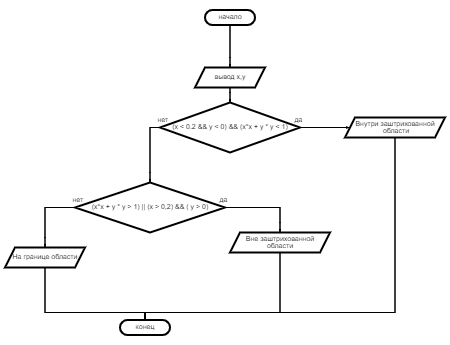
****

Рисунок 1

Код, написанный с добавлением условного оператора if и определяющий какой из трех заданных областей принадлежит точка через входные данные, задаваемые пользователем (рис. 2).

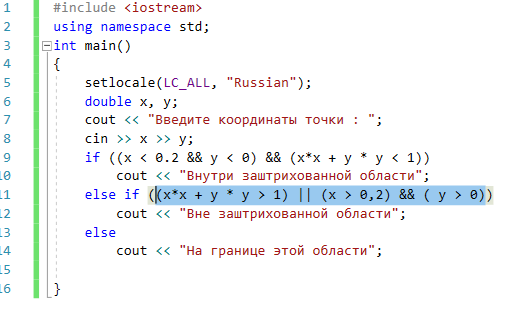


Рисунок 2

После запуска программы, консоль просит ввести две переменные, после их обрабатывает и выводит один из трех ответов, которому принадлежат заданные числа (рис. 3).

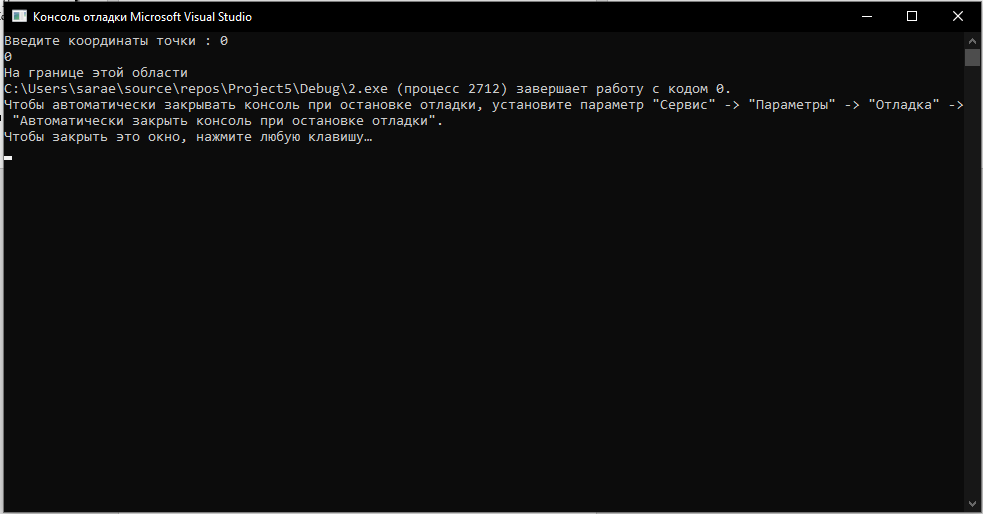


Рисунок 2

**Контрольные** **вопросы**

1. Условный оператор реализует выполнение определённых команд при условии, что некоторое логическое выражение (условие) принимает значение «истина» true . В большинстве языков программирования условный оператор начинается с ключевого слова if (в переводе с англ. — «если»).
2. Условная конструкция в С++ всегда записывается в круглых скобках после оператора if . Внутри фигурных скобок указывается тело условия. Если условие выполнится, то начнется выполнение всех команд, которые находятся между фигурными скобками. Каждому оператору if соответствует только один оператор else. Совокупность этих операторов — else if означает, что если не выполнилось предыдущее условие, то проверить данное. Если ни одно из условий не верно, то выполняется тело оператора else. Если после оператора if, else или их связки else if должна выполняться только одна команда, то фигурные скобки можно не ставить.
3. Общее правило работы с Null-значениями (то, что результат операций над Null-значениями есть Null-значение) применяется к следующим операциям:
4. к арифметическим;
5. к побитным операциям отрицания, конъюнкции и дизъюнкции (кроме законов поглощения);
6. к операциям со строками (например, конкотинации – сцепления строк);
7. к операциям сравнения (<, ≤, ≠, ≥, >).
8. Операторы сравнения и логические операторы: Операторы сравнения.==(равно), != (не равно), < (меньше), <= (меньше или равно), > (больше), >= (больше или равно).

A && B — эквивалент «И». Соответственно возвращает true, если A и B являются истиной.

A || B — эквивалент логического «ИЛИ». Вернет true ели хотя бы одно из выражений является истинным.

A xor B — этот оператор можно сравнить с «ТОЛЬКО ОДИН», соответственно вернет true если A == true и B == false, или наоборот.

!A — данный оператор инвертирует значение A. То есть, если A == true, то он вернет false и наоборот.

1. Отличие составного оператора от блока заключается в том, что составной оператор включает только исполняемые операторы, а блок - также описания объектов. Пример составного оператора: {n++; summa+= (float)n;} Пример блока: {int n=0; n++; summa+= (float)n;} Все операторы программы выполняются последовательно в порядке записи их в программе (сверху вниз, слева направо), если нет операторов передачи управления, изменяющих этот порядок.
2. Оператор условного перехода (любая его форма) может быть вложен в любой другой оператор, управляющий ходом выполнения вычислительного процесса. Это могут быть другие конструкции языка, такие как операторы цикла, оператор when, оператор условного перехода и тому подобное. Допускается любое количество уровней вложений операторов.