МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

## «Московский технический университет связи и информатики»

Кафедра «Информатика»

Лабораторная работа №7 - Проект 1

**«Разработка проекта, решающего задачу**

**принадлежности или непринадлежности точки с**

**заданными координатами геометрической фигуре»**

по теме

**«Логические данные и выражения.**

**Операторы разветвления VС++. Программная реализация**

**базовых разветвляющиеся структур и**

**типовых алгоритмов»**

**по дисциплине**

**«Информатика»**

Выполнил: студент гр. БИК2107 Осетрова К.И.

Вариант №19

Проверил: доц. Кафедры

«Информатика» Гуриков С.Р.

Москва, 2021

Оглавление

[Индивидуальное задание на разработку проекта 3](#_Toc4145)

[Формализация и уточнение задачи 3](#_Toc6766)

[Разработка схем алгоритмов 3](#_Toc20235)

[Разработка программного кода проекта 5](#_Toc24612)

[Выполнение разработанного проекта 7](#_Toc25915)

[Доказательство правильности результата 8](#_Toc7488)

# Индивидуальное задание на разработку проекта

Заданы две простых фигуры: левая полуокружность радиуса 1 с центром в начале координат и дуга, соединяющая две крайние точки полуокружности ((0, 1), (0, -1)) как часть окружности с центром в точке (1,0) и радиусом √2. Попадет ли точка с координатами X, Y в «старый» полумесяц.

# Формализация и уточнение задачи

Построены «вручную» описанная в постановке задаче фигура, которая определяет область попадания точки в заданную геометрическую фигуру, рисунок 1.

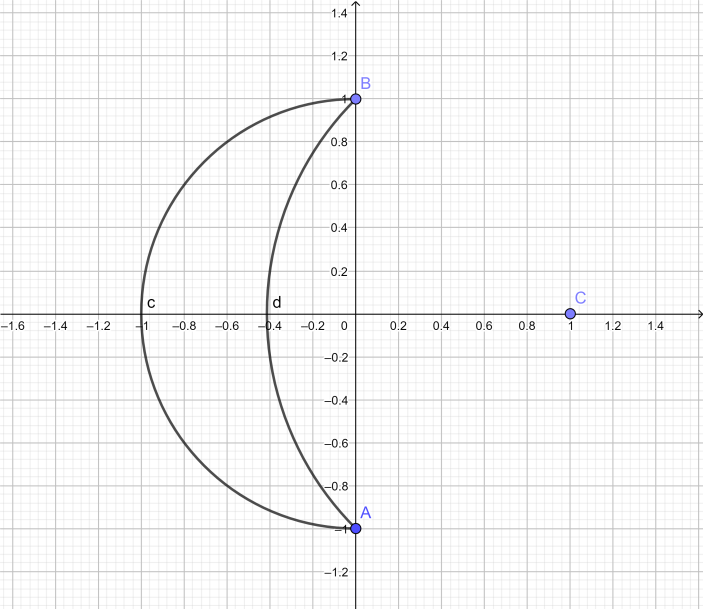


Рисунок 1 - Необходимая область

Определим уравнение полуокружности и кривой.

**Полуокружность:**

**Дуга:**

Требуется вместо уравнений записать неравенства, выполнение которых

необходимо для попадания точки в заданную область. В нашем примере эти неравенства должны выполняться одновременно, и поэтому записаны в виде системы:

При этом логической выражение можно записать следующим образом:

# Разработка схем алгоритмов

**Разработаны схемы 3-х алгоритмов:**

• схема алгоритма **Resh1** решения задачи, использующая стандартное

разветвление и сложное логическое выражение (рисунок 2);

• схема алгоритма **Resh2** решения задачи, использующая вложенные разветвления только с помощью операций отношения, без использования логических операций и сложных логических выражений (рисунок 3);

• схема алгоритма **Resh3** решения задачи, использующая только сложное

логические выражения (рисунок 4).

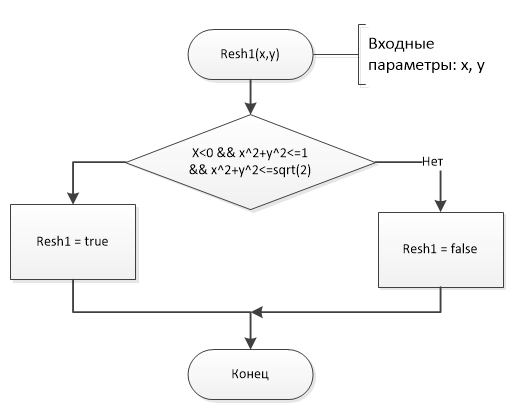


Рисунок 2 - Схема алгоритма функции, использующей стандартное

разветвление и сложное логическое выражение

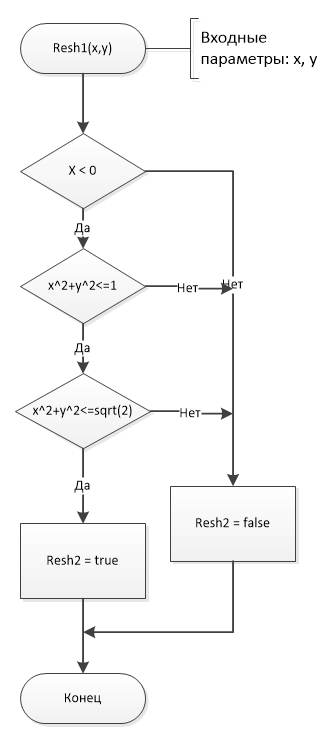


Рисунок 3 - Схема алгоритма функции, реализующей вложенные разветвления только с помощью операций отношения, без использования логических операций и сложных логических выражений

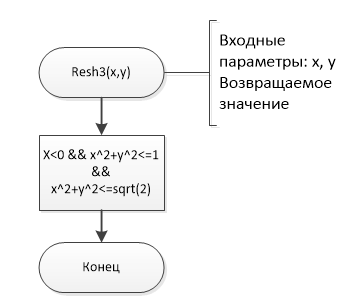


Рисунок 4 - Схема функционального алгоритма Resh3, использующая

только сложное логическое выражение

# Разработка программного кода проекта

Разработан программный код проекта (рисунок 5).

|  |
| --- |
| // Файл с функциями решения задачи, реализующие функциональные алгоритмы  #include <cmath>  // 1-ый способ – стандартное разветвление с логическими операциями  bool Resh1(float x, float y)  {  if (x < 0 && pow(x, 2) + pow(y, 2) <= 1 &&  pow(x, 2) + pow(y, 2) <= sqrt(2))  return (true);  else  return (false);  }  // 2-ой способ – вложенные разветвления только с помощью операций отношения  bool Resh2(float x, float y)  {  if (x < 0)  {  if (pow(x, 2) + pow(y, 2) <= 1)  {  if (pow(x, 2) + pow(y, 2) <= sqrt(2))  return (true);  }  }  return (false);  }  // 3-ой способ – только сложное логические выражения  bool Resh3(float x, float y)  {  return (x < 0 && pow(x, 2) + pow(y, 2) <= 1 && pow(x, 2) + pow(y, 2) <= sqrt(2));  }  // Файл main.cpp  #include <iostream>  using namespace std;  void GetXY(float& x, float& y);  void Put(bool, float, float);  bool Resh1(float x, float y);  bool Resh2(float x, float y);  bool Resh3(float x, float y);  int main()  {  float x, y;  bool b, c, d; // Признак попадания в заданную область  GetXY(x, y); // Вызов функции ввода исходных данных  cout << " Решение 1-й функции:" << endl;  b = Resh1(x, y); // Вызов 1-й функции решения  Put(b, x, y); // Вызов функции вывода результатов  cout << " \nРешение 2-й функции:" << endl;  c = Resh2(x, y); // Вызов 2-й ф-ции решения  Put(c, x, y);  cout << "\n Решение 3-й функции:" << endl;  d = Resh3(x, y); // Вызов 3-й ф-ции решения  Put(d, x, y);  system("PAUSE");  return 0;  }  // Файл GetPut.cpp с функциями ввода и вывода  #include <iostream>  using namespace std;  // Определение функции ввода  void GetXY(float& x, float& y)  {  setlocale(LC\_ALL, "rus");  cout << " Введите координаты точки x, y\n ";  cin >> x >> y;  }  // Определение функции вывода  void Put(bool b, float x, float y)  {  setlocale(LC\_ALL, "rus");  cout << " Точка с координатами ( " << x << " ," << y << " )" << endl;  if (b) cout << "попала в заданную область" << endl;  else cout << "не попала в заданную область" << endl;  } |

Рисунок 5 - Программный код проекта

# Выполнение разработанного проекта

Получены результаты выполнения проекта, приведенные на рисунках 6-7, для тестовых данных точек, попадающих и не попадающих в заданную область.

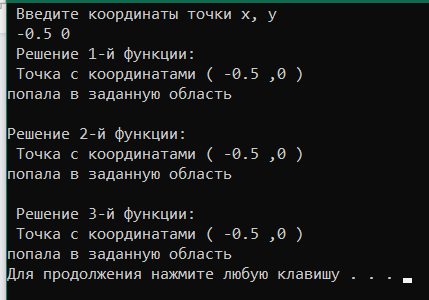


Рисунок 6 – Результаты выполнения проекта для тестовых данных

точки, попадающей в заданную область.

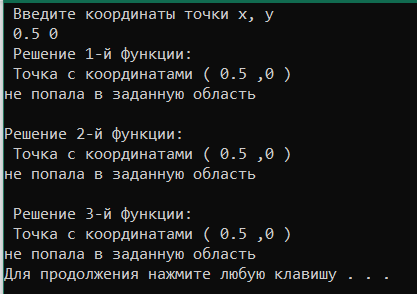


Рисунок 7 – Результаты выполнения проекта для тестовых данных точки,

не попадающей в заданную область

# Доказательство правильности результата

Разработаны с тестовые исходные данные для контрольного решения, проверяющие попадание и непопадание точки с различными координатами в заданную область, и получены результаты:

|  |  |  |
| --- | --- | --- |
| Исходные данные | | Результат |
| x = -0,5 | y = 0 | Точка попала в область |
| x = 0,5 | y = 0 | Точка не попала в область |

При тестовых данных результаты ручного расчета и вычисления на компьютере совпадают.