|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Определение принадлежности точки невыпуклому многоугольнику | | | |  |
| Внешняя спецификация | | | | | |
| Студент | |  | Нечта Б.О. |  | |
| Преподаватель | | Кулюкин К.С. | | | |
| Сдано | |  | | | |

Глоссарий понятий

Плоскость – двумерное пространство, в котором расположены все точки и фигуры в данной задаче. Плоскость задаётся декартовой системой координат

(x, y).

Точка – геометрический объект, имеющий координаты (x, y) на плоскости. В задаче рассматриваются точки, входящие в состав многоугольника, а также проверяемая точка.

Полуплоскость – часть плоскости, разделённая прямой. Используется в алгоритмах определения принадлежности точки многоугольнику.

Многоугольник – замкнутая геометрическая фигура, образованная последовательностью отрезков, соединяющих вершины.

Выпуклый многоугольник – многоугольник, у которого все углы меньше 180 градусов, и для любой проведённой прямой многоугольник лежит только по одну сторону от неё.

Невыпуклый многоугольник – многоугольник, содержащий хотя бы один внутренний угол, больший 180 градусов, что делает его форму "вогнутой".

Координаты точки в двумерной плоскости – пара целых чисел (x; y), записанная в формате x;y.

1. Назначение

Программа предназначена для определения принадлежности точки невыпуклому многоугольнику.

1. Функциональные требования

* 1. Действия над объектами

Программа должна определять, принадлежит или не принадлежит заданная точка невыпуклому многоугольнику.

* 1. Ограничения

-Программа должна обрабатывать координаты точек в диапазоне [-999, 999]. -Программа должна учитывать, что входные данные представляют вершины многоугольника в порядке обхода.

* Количество вершин многоугольника должно быть не менее 3 и не более 1000.
* Координаты точек должны быть целыми числами.
* Соответствие задаваемых вершин и координат
* Входные данные не должны содержать дублирующихся точек.
* Последовательность точек должна формировать корректный многоугольник (без пересечений рёбер, кроме соседних).
* Проверяемая точка должна находиться в пределах допустимого диапазона координат.
* Входные данные не должны содержать символов, кроме чисел, разделителей (точка с запятой) и перевода строки.
* Пустые строки или некорректное количество элементов в строках считаются ошибками ввода.
* Обработка в плоскости только (x;y), не учитывая многомерные плоскости.

3. Входные и выходные данные

Программа должна получать два параметра командной строки: имя входного файла с данными о многоугольнике и точке, а также имя файла для записи результата.

Пример команды запуска программы: polygon.exe C:\\Documents\\input.txt .\out.txt.

Входные данные представляются в виде текстового файла с расширением .txt. Данные разделяются с помощью переноса на другую строку и записываются в следующем порядке:

* Первое число — количество вершин многоугольника.
* Далее следуют координаты каждой вершины (x;y), записанные построчно в порядке обхода многоугольника.
* Последняя строка содержит координаты точки, принадлежность которой необходимо проверить.

Выходной файл должен быть текстовым файлом с расширением .txt. Он должен содержать одну строку: "принадлежит" или "не принадлежит" в зависимости от результата проверки.

1. Требования к надежности

В процессе работы программы не должно происходить ее аварийного завершения или зависания. В случае ошибки во входных данных, пользователь должен получать сообщения, перечисленные в таблице 1, после чего программа должна корректно завершаться. Сообщения об ошибке выводятся в консоль, выходной файл при этом не создается.

Таблица 1 – Список сообщений об ошибках

|  |  |  |
| --- | --- | --- |
| Ситуация | Пример | Сообщение об ошибке |
| Указанный входной файл не существует или нет доступа. |  | Неверно указан файл с входными данными. Возможно, файл не существует или нет прав на чтение. |

|  |  |  |
| --- | --- | --- |
| Невозможно создать указанный выходной файл. |  | Неверно указан файл для выходных данных. Возможно, указанного расположения не существует или нет прав на запись. |
| Количество вершин многоугольника меньше 3 или больше  1000. | 2  0;0  1;1  5;5 | Некорректное количество вершин.  Допустимый диапазон:  [3, 1000]. |
| Координаты вершины многоугольника выходят за пределы диапазона [-999, 999]. | 3  -1000;0  500;500  0;100  10;10 | Координаты вершины (-  1000;0) выходят за допустимый диапазон [999, 999]. |
| Координаты проверяемой точки выходят за пределы диапазона [-999, 999]. | 3  0;0  1;1  2;2  1000;1000 | Проверяемая точка  (1000;1000) выходит за допустимый диапазон [999, 999]. |
| Одна или несколько координат не являются целыми числами. | 3  0.5;1  2;2  3;3  2;3 | Координата 0.5 не  является целым числом.  Допустимы только целые числа. |
| Входные данные содержат дублирующиеся вершины | 4  0;0  1;1  2;2 | Входные данные содержат дублирующиеся вершины: (0;0). |

|  |  |  |  |
| --- | --- | --- | --- |
|  | 0;0  5;5 |  |  |
| Последовательность  точек формирует  некорректный многоугольник  (пересечение рёбер, кроме соседних). | 4  0;0  2;2  2;0  0;2  1;1 |  | Входные данные формируют некорректный многоугольник: пересечение рёбер. |
| Входные данные содержат некорректные символы (буквы, знаки препинания, лишние пробелы и т.д.). | 3  0;0  1;a  2;2  1;1 |  | Входные данные содержат некорректные символы. Разрешены только числа, точки с запятой и переводы строк. |
| Строка содержит некорректное количество элементов  (не 2 числа, разделённых ;). | 3  0;0  1;1;2  2;2  1;1 |  | Некорректное количество элементов в строке: 1;1;2. Каждая точка должна содержать два целых числа, разделённых ;. |
| Входной файл содержит пустые строки. | 3  0;0  (пустая  1;1  2;2  1;1 | строка) | Обнаружена пустая строка во входных данных. Удалите лишние строки. |
| Не соответствует количество координат, количеству вершин | 4  0;0  1;1; |  | Обнаружено не соответствие  количество вершин |
|  | 2;2  1;1 | | количеству координат. Добавьте или уберите координаты |
| Вершины заданы не в порядке обхода | 4  0;0  4;4  0;4  4;0  2;2 | | Вершины многоугольника заданы не в порядке обхода.  Упорядочьте их корректно. |

1. Требования программной совместимости

Программа будет разработана на языке С++ с использованием Microsoft Visual Studio 2019. Дополнительного программного обеспечения не требуется.

Входные файлы могут быть подготовлены в редакторе Блокнот.

Выходные файлы могут быть прочитаны с его помощью.

Приложение А

Примеры входных и выходных данных

|  |  |
| --- | --- |
| input.txt | output.txt |
| 4  0;0  4;0  4;4  0;4  2;2 | принадлежит |
| 4  0;0  4;0  4;4  0;4  5;5 | не принадлежит |
| 4  0;0  4;0  4;4  0;4  4;2 | принадлежит |

|  |  |
| --- | --- |
| (Файл отсутствует или указан неверный путь) | Неверно указан файл с входными данными. Возможно, файл не существует или нет прав на чтение. |
| (Неверный путь к файлу вывода) | Неверно указан файл для выходных данных. Возможно, указанного расположения не существует или нет прав на запись. |