Физико-математический лицей № 239

**Поиск наибольшей площади пересечения среди множеств острых углов и прямоугольников.**

Отчет о годовом проекте по информатике

Работу выполнил

Ученик 10-7 класса Сорокин Владимир

Санкт-Петербург

2021

# Постановка задачи

На плоскости заданы множества прямоугольников и острых углов. Найти такую пару прямоугольник - острый угол, что площадь фигуры, находящейся внутри найденного прямоугольника и острого угла будет максимальной.

# Уточнение исходных и выходных данных и ограничений на них

**Исходные данные:**

## Для угла: координаты вершины угла + координаты точек, лежащих на лучах угла. Для прямоугольника: координаты двух вершин прямоугольника и точки, лежащей на прямой, проходящей через 2 другие вершины.

Все точки задаются координатами x и y.



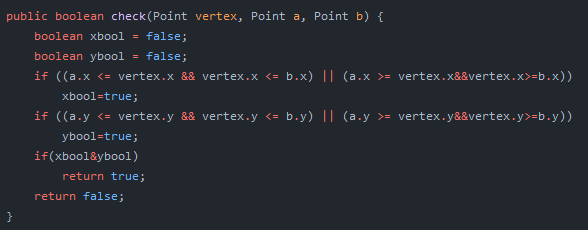
**Выходные данные:**

На экран выводится многоугольник, являющийся ответом на поставленную задачу. Также  
выводится пара прямоугольник – острый угол, задающие многоугольник с наибольшей площадью.

# Математическая модель

Для того чтобы искать точки пересечения среди лучей и отрезков, нужно было понять для начала задать прямую через отрезок и лук, а потом проверить: лежит ли точка на луче и отрезке.

Если координаты рассматриваемой точки лежат между координатами вершин отрезка, то точка принадлежит отрезку.



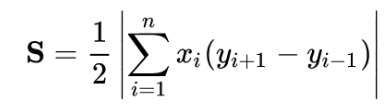
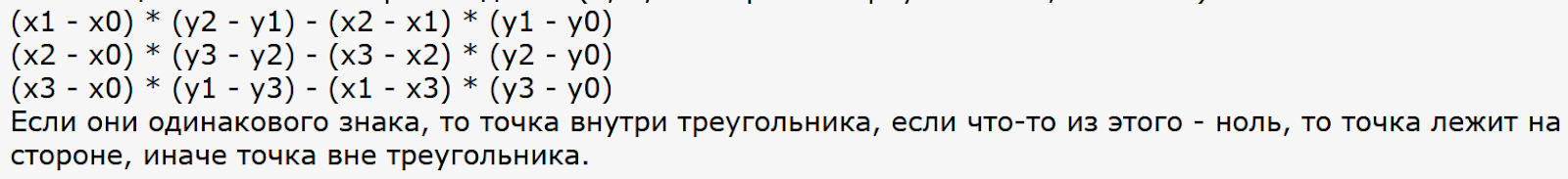
# Далее необходимо было построение перпендикулярных прямых для нахождения вершин прямоугольника. Пусть y = k1 \* x+b1 – исходная прямая. А t = k2 \* x + b2 – перпендикулярная ей. Тогда k1 \* k2 = -1.

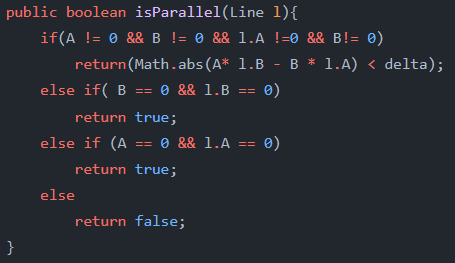
# https://lh3.googleusercontent.com/JVYmQZm3yBAT75AK203eTFAys6jrn8U_WpYLlN9sZhDNCFahpV266nngoKo4qw_X0-uncqhomlmh0-9BTIJN2MvqtlcClOc9I216uDieFEWyfGePJ-LKPTcCmbrxoTjhZa7CyJTmTZ4

# Также для определения точек пересечения между прямыми, нужен был метод, что мог быть их найти. Данный метод решает несложную систему и выводит точку пересечения прямых.

# https://lh3.googleusercontent.com/JVYmQZm3yBAT75AK203eTFAys6jrn8U_WpYLlN9sZhDNCFahpV266nngoKo4qw_X0-uncqhomlmh0-9BTIJN2MvqtlcClOc9I216uDieFEWyfGePJ-LKPTcCmbrxoTjhZa7CyJTmTZ4

Для нахождения площади многоугольника использовалась формула Гаусса.

  
  
  
 Проверка того, лежит ли точка внутри треугольника.  
  
  
  
  
  
  
  
  
 Для избежания случаев, когда прямоугольник и угол могли бы быть заданы одной прямой, нужен был метод, проверяющий параллельность прямых. У параллельных прямых коэффициенты **b** равны.



# Далее самой важной частью было сортировкой точек многоугольник таким образом, чтобы получался выпуклый многоугольник. Перебор осуществлялся по просмотру углов, образовавшихся между центром многоугольника и вершинами.

# 

**Анализ используемой структуры данных**

Для хранения углов, прямоугольников и многоугольника использовались динамические списки.  
 Список angles хранит все в себе все углы программы. Список rectangles хранит все введенные прямоугольники. Список rectangle\_end, angle\_end, pointsos хранят в себе: итоговую пару прямоугольник – острый угол и многоугольник с наибольшей площадью.  
 Остальные списки использовались, как временные хранилища.

Для задачи точек (Point) используется два числа типа double (x, y).  
Острый угол задается через 3 точки, через вершину угла и 2 точки, что лежат на двух разных лучах угла (vertex, a, b).  
Прямоугольник задается через 3 точки: 2 вершины и точку, лежащей на прямой, проходящей через 2 другие вершины.

**Выбор метода решения**

Для начала необходимо было перебрать все прямоугольники и острые углы. Рассматривая пару прямоугольник – острый угол:  
1)Необходимо найти точки пересечения отрезков и лучей (intersectionS).  
2) Проверка нахождения вершины угла в прямоугольнике и проверка нахождения вершин прямоугольника в угле (RectangleInAngle).  
3) Сортировка списка, содержащего вершины многоугольника, и нахождение его площади (Convexity).  
4) Сравнение площади данного многоугольника с максимальной на данный момент.

Здесь я оставлю цитаты моего одноклассника про бесполезность векторов.

«**ВНН. Вектора не нужны!** В этом небольшом блоке я хочу объяснить, почему использование векторов в этой задачи бессмысленно и от него нужно отказаться. Я не понимаю, зачем нужно реализовывать класс Vector, если это натурально тоже самое, что и класс точки Point. Нас просят сделать следующее:

«реализовать класс двумерного вектора problem.Vector2 с двумя публичными вещественными полями x и y, а также конструктором с двумя вещественными аргументами, задающий значения этих полей».

Зачем делать всё сложнее и работать с точкой, как с вектором? Зачем работать с вектором, интуитивное понимание которого сложнее, если можно работать с точкой? В реализации, по крайней мере, моей задачи, нигде вектора не используются в математическом смысле. Я рассматриваю, например, расположение точки относительно прямой. Можете себе представить такие же рассуждения с использованием вектора? Не думаю»

Николай Борисенко

**Анализ правильности решения**

Я считаю, что мой проект работает правильно. Все, что используется для решения вытекает из логического размышления над проектом. Используются довольно простые методы, так что вероятность каких – либо неурядиц мала.

**Перспективы проекта**

В этом проекте многое можно изменить для удобства пользователей. Можно будет добавить изменения разрешения, изменение точности. Также можно было бы создавать разные цветовые гаммы проекта (Например для дальтоников). В проекте многое можно изменить, добавить, убрать, оптимизировать. Поэтому я считаю, что некоторые перспективы у будущем у него все же есть.