

# **ОТЧЕТ**

## **по лабораторной работе №4 «Алгоритмы отсечения отрезков и многоугольников»**

**Вариант: 15**

Выполнил: Студент 4 гр. Минковский Виталий

### **1. Цель работы**

Закрепление теоретического материала и практическое освоение основных методов и алгоритмов отсечения отрезков и многоугольников. Разработка веб-приложения для визуализации работы алгоритмов.

### **2. Постановка задачи**

Написать приложение, иллюстрирующее работу следующих алгоритмов (согласно варианту 15):

- 1. Часть 1:** Алгоритм отсечения отрезков прямоугольным окном — **Алгоритм средней точки (Midpoint Subdivision).**
- 2. Часть 2:** Алгоритм отсечения выпуклого многоугольника — **Алгоритм Сазерленда-Ходжмана** (для нечетных вариантов).

**Входные данные:** Текстовый файл, содержащий количество отрезков, координаты отрезков и координаты отсекающего окна

### **Требования:**

- Отображение системы координат.
- Визуализация окна и исходных фигур разными цветами.
- Визуализация результата отсечения.

### **3. Теоретические сведения**

#### **3.1. Алгоритм средней точки (Midpoint Subdivision)**

Алгоритм является модификацией метода Коэна-Сазерленда. Он используется для отсечения отрезков прямоугольным окном.

#### **Принцип работы:**

1. Каждой конечной точке отрезка присваивается 4-битный код области (Outcode), определяющий ее положение относительно окна (слева, справа, снизу, сверху).

2. Выполняется проверка на тривиальную видимость (код OR = 0) или невидимость (код AND != 0).
3. Если отрезок нельзя тривиально принять или отбросить, он делится пополам.
4. Процесс рекурсивно повторяется для полученных половин до тех пор, пока отрезок не станет меньше заданной точности или не будет принят/отброшен.

Это позволяет избежать сложных вычислений точек пересечения через уравнения прямых, заменяя их быстрым вычислением средних координат.

### **3.2. Алгоритм Сазерленда-Ходжмана**

Используется для отсечения многоугольников прямоугольным окном. Алгоритм позволяет корректно отсечь фигуру, сохранив её замкнутость (добавляя новые ребра вдоль границ окна).

**Принцип работы:**

Многоугольник последовательно отсекается каждой из четырех сторон окна (левой, правой, нижней, верхней). Выходной набор вершин после обработки одной стороной становится входным для следующей. При обходе ребер ( $S \rightarrow E$ ) возможны 4 случая:

1. **Внутри → Внутри:** Сохраняется  $EE$ .
2. **Внутри → Снаружи:** Сохраняется точка пересечения  $II$ .
3. **Снаружи → Снаружи:** Ничего не сохраняется.
4. **Снаружи → Внутри:** Сохраняется точка пересечения  $II$  и точка  $EE$ .

### **4. Реализация**

Приложение разработано на языке **Python** с использованием фреймворка **FastAPI**.

- **Backend:** Реализует математическую логику алгоритмов.
- **Frontend:** HTML5 + JavaScript (Canvas API) для отрисовки и ввода данных.

### **5. Результаты работы программы**

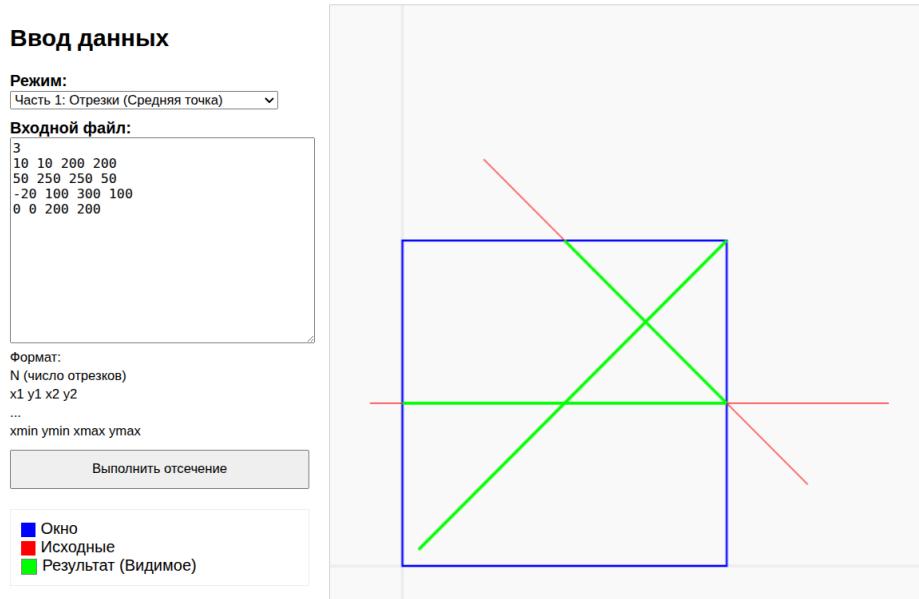
#### **Тест 1. Отсечение отрезков (Алгоритм средней точки)**

**Входные данные:**

3

10 10 200 200  
50 250 250 50  
-20 100 300 100  
0 0 200 200

### Результат:



- Синий прямоугольник — отсекающее окно.
- Красные линии — исходные отрезки.
- Зеленые отрезки — видимые части.
- Отрезок (10,10)-(200,200) полностью видим.
- Отрезок (-20,100)-(300,100) обрезан слева и справа.

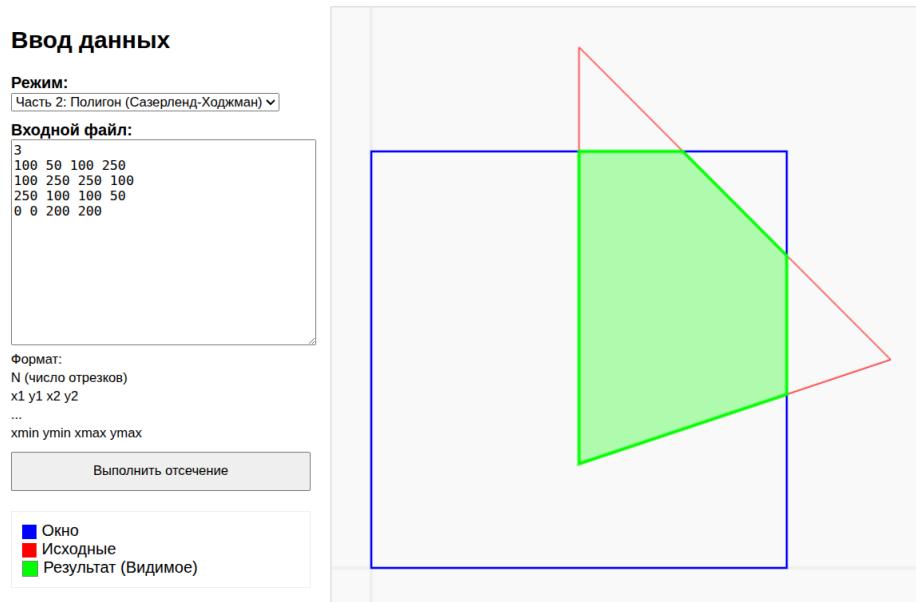
### Тест 2. Отсечение многоугольника (Алгоритм Сазерленда-Ходжмана)

Для тестирования использован треугольник, частично перекрывающий окно.

#### Входные данные:

3  
100 50 100 250  
100 250 250 100  
250 100 100 50  
0 0 200 200

## Результат:



Программа корректно замкнула контур и отсекла части треугольника, выходящие за границы окна (верхний угол и правый угол). Зеленым цветом закрашена результирующая область.

## 6. Вывод

В ходе выполнения лабораторной работы были изучены и реализованы алгоритмы отсечения: алгоритм средней точки для отрезков и алгоритм Сазерленда-Ходжмана для выпуклых многоугольников. Создано веб-приложение на FastAPI, которое парсит входной файл, выполняет вычисления на стороне сервера и визуализирует результат с помощью Canvas API, корректно обрабатывая масштаб координат. Алгоритм средней точки показал свою эффективность за счет отсутствия сложных вычислений пересечений, а алгоритм Сазерленда-Ходжмана продемонстрировал способность сохранять замкнутость полигона при отсечении.