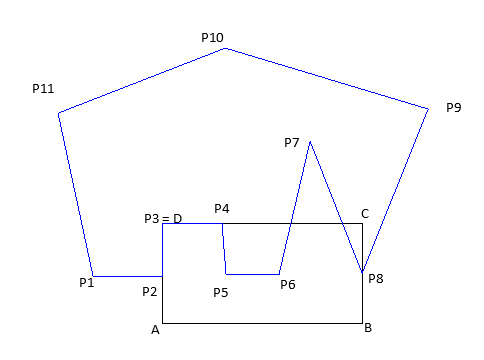
Граница отсекателя обходится по часовой стрелке, то его внутренняя область лежит по правую сторону от границы

**1. Отсечение многоугольника по AB:**



**определения видимости вершин многоугольника:**

P1: Z = (P1.x – A.x)(B.y – A.y) – (B.x – A.x)(P1.y – A.y) = ) – (B.x – A.x)(P1.y – A.y) > 0 (B.x – A.x > 0, P1.y – A.y < 0)

Получим:

Pi: Z > 0, i = 1..11, значит все вершины находятся на правой стороне по сравнению с АB, так что все вершины видимы для АB

newPolygon = []

P1 -Первая вершина

* F = P1
* S = P1
* S видимa для АB, занесение ее в массив newPolygon = [P1, ]

P2: SP2 = P1P2

* P1P2 полностью видимa для АB -> пересечения нет
* S = P2
* видимa для АB, занесение ее в массив newPolygon = [P1, P2]

…

P11- SP11 = P10P11

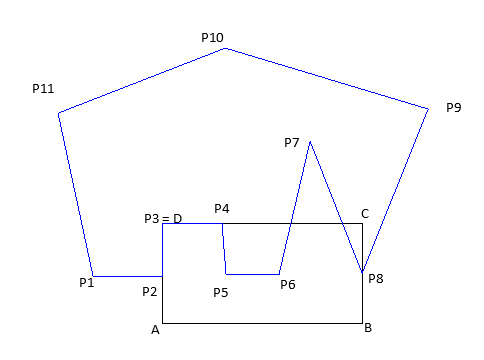
* P10P11 полностью видимa для АB -> пересечения нет
* S = P11
* S видимa, занесение ее в массив newPolygon = [P1, P2, …, P10, P11]

пересечения SF (P11P1) с AB нет

получаем:

P = newPolygon = [P1, P2, …, P11]

**2. Отсечение многоугольника по AB:**



T2

T1

определения видимости вершин многоугольника:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| Z | >0 | >0 | >0 | >0 | >0 | >0 | >0 | =0 | <0 | >0 | >0 |
|  |  |  |  |  |  |  |  |  | невидимa |  |  |

newPolygon = []

P1 -Первая вершина

* F = P1
* S = P1
* S видимa для АB, занесение ее в массив newPolygon = [P1, ]

…

newPolygon = [P1, P2, P3, P4, P5, P6,P7 ]

S = P7

P8: SP8 = P7P8

* видимо частично и выходит из отсекателя
* T = P8
* занесение T в массив newPolygon = [P1, P2, P3, P4, P5, P6,P7, T = P8 ]
* S = P8
* S видимa
* занесение T в массив newPolygon = [P1, P2, P3, P4, P5, P6,P7, T = P8, P8 ]

P9: SP9 = P8P9

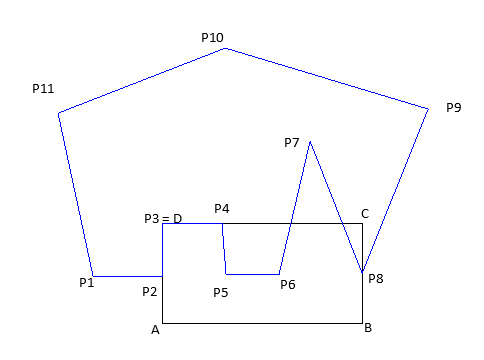
* P8P9 полностью невидимо, -> пересечения нет
* S = P9
* S

P10: SP10 = P9P10

* и входит внутрь отсекателя
* точкa пересечения T2
* занесение T в массив newPolygon = [P1, P2, P3, P4, P5, P6,P7, T = P8, T2]
* S = P10, видимa, занесение в массив = [P1, P2, P3, P4, P5, P6,P7, T = P8, T2, P10]

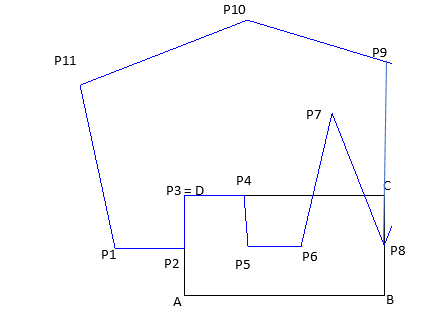
….

P = newPolygon = = [P1, P2, P3, P4, P5, P6,P7, T = P8, T2, P10, P11]



P9

**3. Отсечение многоугольника по CD:**



T4

T3

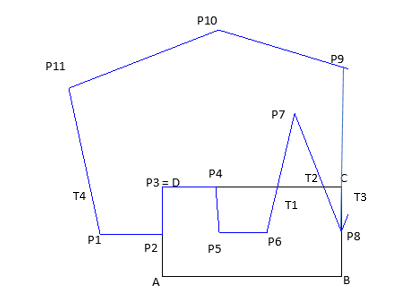
T2

T1

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| Z | >0 | >0 | = 0 | =0 | >0 | >0 | <0 | >0 | <0 | <0 | <0 |
|  |  |  |  |  |  |  | невидимa |  | невидимa | невидимa | невидимa |

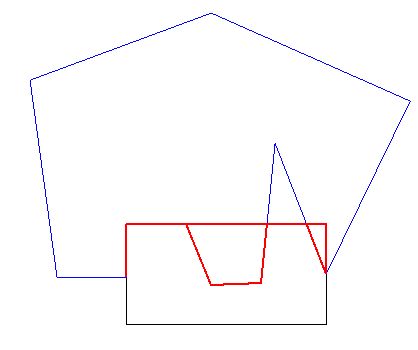
newPolygon = [P1, P2, P3, P4, P5, P6, T1, T2, P8, P8, T3, T4 ]

**4. Отсечение многоугольника по DA:**



|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P | P1 | P2 | P3 | P4 | P5 | P6 | T1 | T2 | P8 | T3 | T4 |
| Z | <0 | =0 | =0 | >0 | >0 | >0 | >0 | >0 | >0 | >0 | <0 |
|  | невидимa |  |  |  |  |  |  |  |  |  | невидимa |

newPolygon = [P2, P3, P4, P5, P6, T1, T2, P8, P8, T3, T5 = P3]



1. **Как определяли нужное направление нормали?**

* Если вектор стороны многоугольника образован как разность векторов пары смежных его вершин Pi-1 и Pi и если скалярное произведение нормали n и вектора от Pi-1 до Pi+1 положительно, то n - внутренняя нормаль. В противном случае n - внешняя нормаль. Для получения внутренней нормали полученное значение n надо умножить на -1.

1. **Какой недостаток имеет алгоритм, когда он появляется?**

* требует вычисления и запоминания большого количества вершин промежуточных многоугольников
* рисование ложных ребер: в случае, когда в результате отсечения получается несколько не связанных друг с другом многоугольников.
* использовать в качестве отсекателя только выпуклые многоугольники: когда необходимо выполнить отсечение невыпуклыми многоугольниками.

1. **Пересечения каких геометрических объектов находили и как?**

**В этом алгоритме, находил пересечения ребер многоугольника и отсекателя.**

* + - * Если концевые точки отрезка имеют разную видимость (одна вершина видима, а другая невидима), то пересечение есть, в противном случае пересечения не будет
      * Пусть P1P2 - концевые точки ребра отсекаемого многоугольника, а C1, C2 - концевые точки ребра отсекателя.
        + P(t) = P1 + (P2 – P1)t, 0 ≤ t ≤1
        + C(s) = C1 + (C2 – C2)s, 0 ≤ s ≤1
        + P(t) = C(s)
        + Если система не имеет решений, то отрезки параллельны
        + Если решение есть, но оно достигается при значениях параметров t или s, выходящих за требуемый диапазон, то пересекаются не сами отрезки, а прямые, проходящие через них (продолжения отрезков).