|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Лабораторная работа № \_\_8\_\_**

|  |  |
| --- | --- |
| **Тема РЕАЛИЗАЦИЯ АЛГОРИТМА ОТСЕЧЕНИЯ ОТРЕЗКА ПРОИЗВОЛЬНЫМ ВЫПУКЛЫМ ОТСЕКАТЕЛЕМ (АЛГОРИТМ КИРУСА-БЕКА)**  **Студент \_Брянская Е.В.\_\_\_\_\_\_\_\_**  **Группа \_ИУ7-42Б\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель \_Куров А.В.** |  |

Москва.

2020 г.

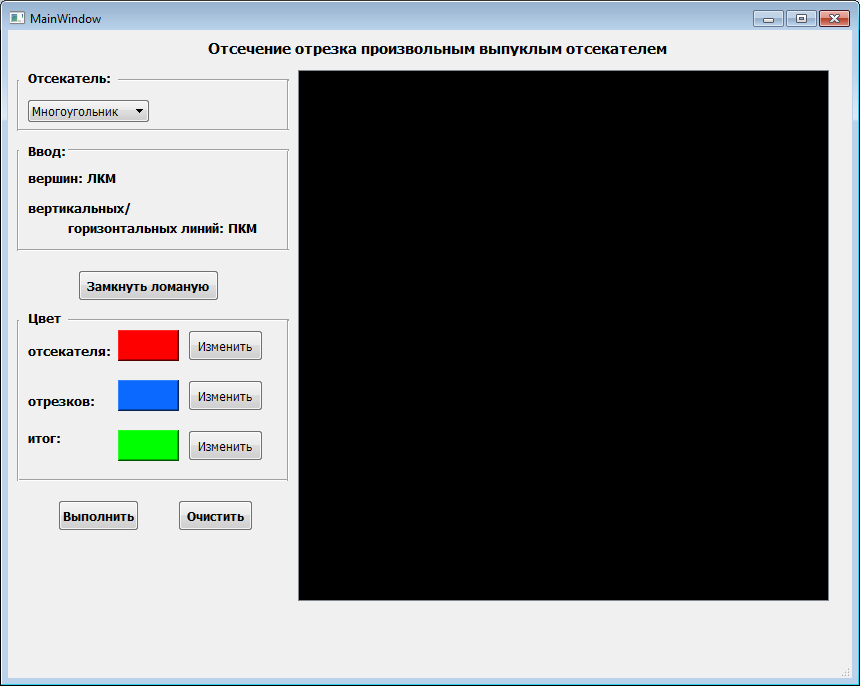
**Цель работы:** изучение и программная реализация алгоритма отсечения отрезка произвольным выпуклым отсекателем.

Необходимо предусмотреть:

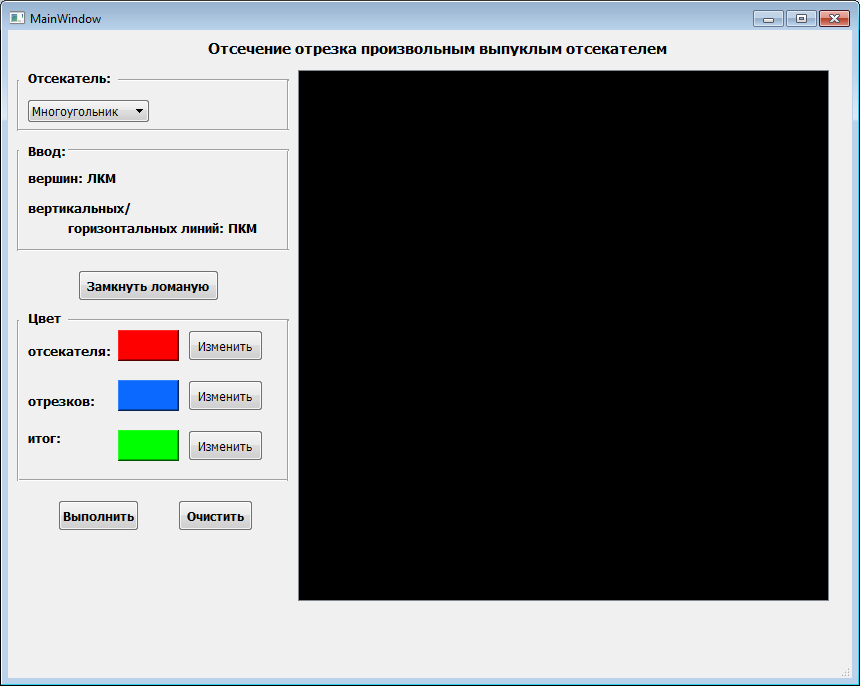
1. Необходимо обеспечить ввод отсекателя – произвольного многоугольника. Высветить его первым цветом.
2. Обеспечить ввод нескольких (до десяти) различных отрезков (высветить их вторым цветом). Отрезки могут иметь произвольное расположение: горизонтальные, вертикальные, имеющие произвольный наклон.
3. Ввод осуществлять с помощью мыши и нажатия других клавиш.
4. Выполнив отсечение отрезков, показать результат третьим цветом. Исходные отрезки не удалять.

**Интерфейс**

Пользователю предоставляется следующий интерфейс для работы:

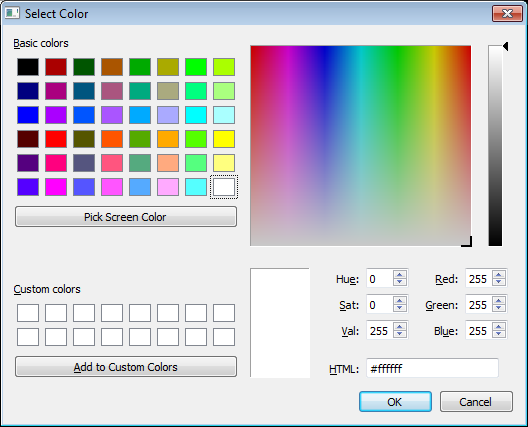


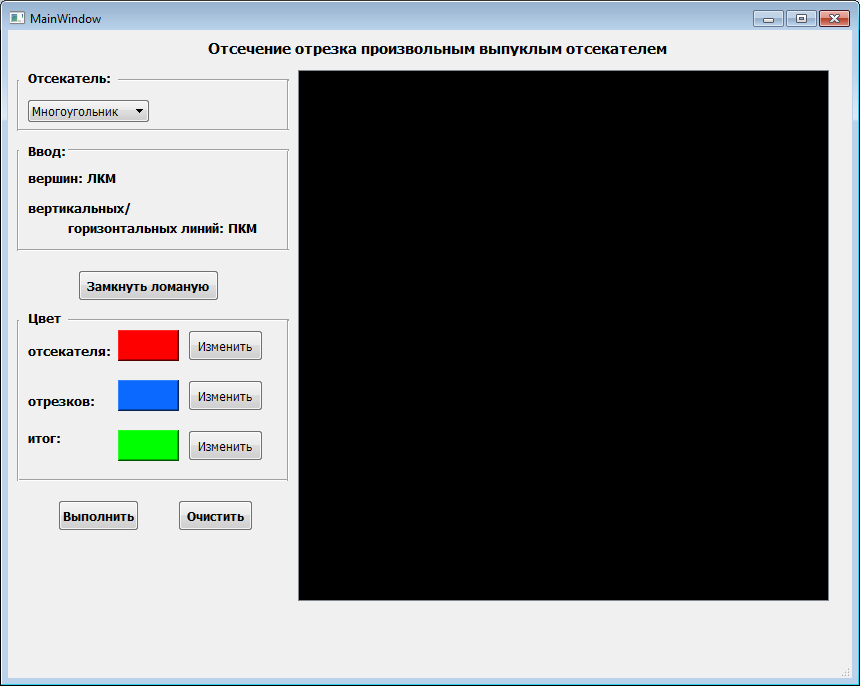
Пользователь сначала должен ввести отсекатель. Ввод вершин осуществляется с помощью мыши, путём нажатия на черный холст. Если нажимается левая кнопка мыши, то рисуется соответствующее ребро, если пользователю нужно изобразить вертикальную или горизонтальную прямую, то он должен сместить курсор на необходимое расстояние и нажать правую кнопку мыши, программа сама скорректирует координаты и выполнит задачу.

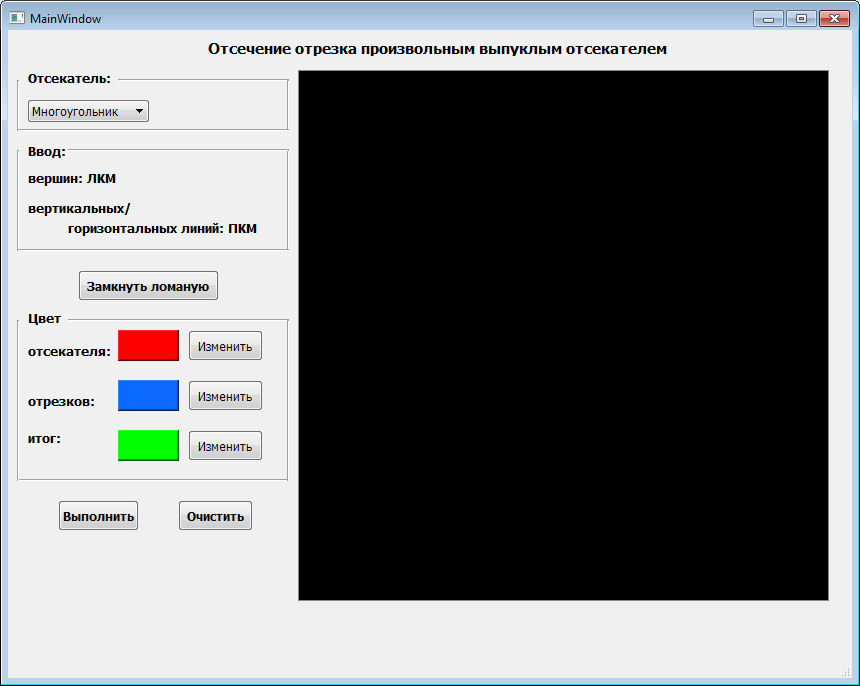
Команда «Замкнуть ломаную» автоматически соединяет начальную точку с конечной.

Пользователь может сделать это и самостоятельно, достаточно относительно близко к начальной вершине нажать мышкой (предусмотрена погрешность, попав в которую координаты программно корректируются) и рисуется соответствующее ребро.

Чтобы ввести отрезки (до 10 штук), пользователь должен указать их расположение на холсте, используя левую кнопку мыши. Если необходимо ввести горизонтальный или вертикальный отрезок, то тогда нужно сместить курсор на необходимое расстояние и нажать правую кнопку мыши (аналогично вводу рёбер многоугольника).

Также пользователь может выбирать цвет отсекателя, отрезков и итогового результата:



Ещё предусмотрены такие команды, как «Выполнить» и «Очистить» для соответствующих задач.

**Алгоритм Кируса - Бека**

*Об алгоритме*

Отсекающей областью в данном алгоритме является выпуклый многоугольник. Это одно из ключевых условий, которое проверяется на начальном этапе работы.

В этом алгоритме важна ориентация отрезков для осуществления анализа их видимости (на основе скалярных произведений). В скалярном произведении используется косинус, знак которого зависит от угла между векторами, таким образом, распознаётся положение отрезка по отношению к отсекателю, и делается вывод о том, продолжать анализ или нет.

Сами отрезки представляются в виде параметрического уравнения:

, где 0 <= t <= 1

Анализ расположения текущего отрезка производится также и по величине t (определяет точку пересечения отрезка с границей отсекателя). В этом алгоритме используются внутренние нормали к каждому ребру многоугольника. Поэтому их также необходимо найти, и важно следить за тем, чтобы нормаль являлась внутренней.

В самом начале работы предполагается, что отрезок видимый. Используются такие величины, как D – директриса, показывает направление текущего отрезка, wi – вектор, соединяющий конец отрезка Р1 и какую-либо точку на текущей стороне многоугольника.

Далее вычисляется скалярное произведение нормали и D, нормали и wi, анализируя их знак, проводится заключение о расположении отрезка, подсчитываются значения параметра t. Причём эти значения разделяются на те, что определяют начало видимой части (нормаль \* D > 0 и t < 1) и на те, что определяют конец (нормаль \* D < 0 и t > 0). На основе этого выбирается максимальное значение t (для первой группы) и минимальное (для второй). Таким образом, определяется видимая часть.

Если скалярное произведение нормали и D не равно нулю, то можно посчитать параметр t, допустимыми значениями которого являются: для нижней границы t <= 1, а для верхней t >= 0. Если оказалось, что t имеет недопустимые значения, то отрезок полностью невидимый, и алгоритм завершается.

Если же оно равно нулю, значит, отрезок параллелен текущему ребру отсекателя. Точку пересечения в этом случае определить нельзя (возникает деление на ноль), поэтому анализируется скалярное произведение нормали и wi. Если оно отрицательно, то отрезок невидим, работа с этим отрезком прекращается. Если же нет, то осуществляется анализ уже относительно следующего ребра.

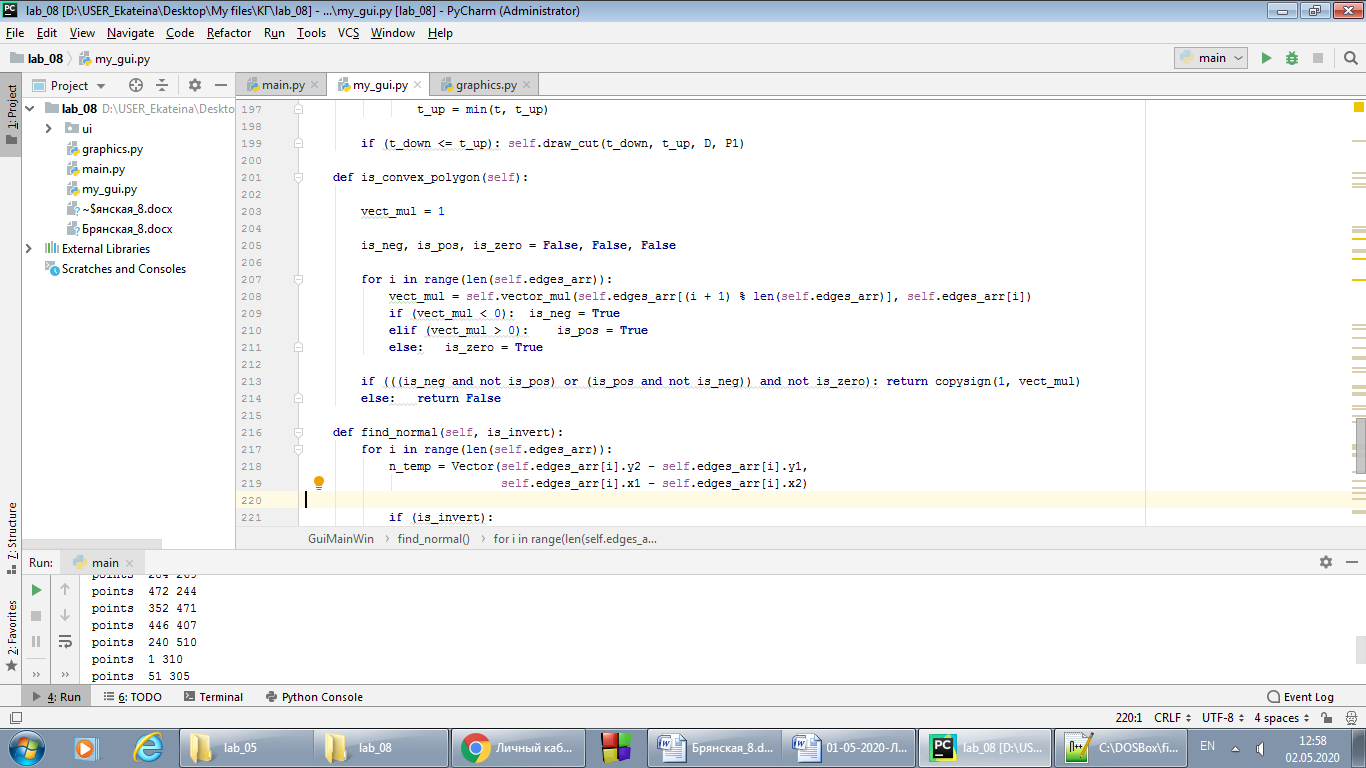
Перед отрисовкой итогового результата проверяется правильность границ видимой области (левая не должна превышать правую), это делается для того, чтобы избежать ситуации изображения невидимого отрезка.

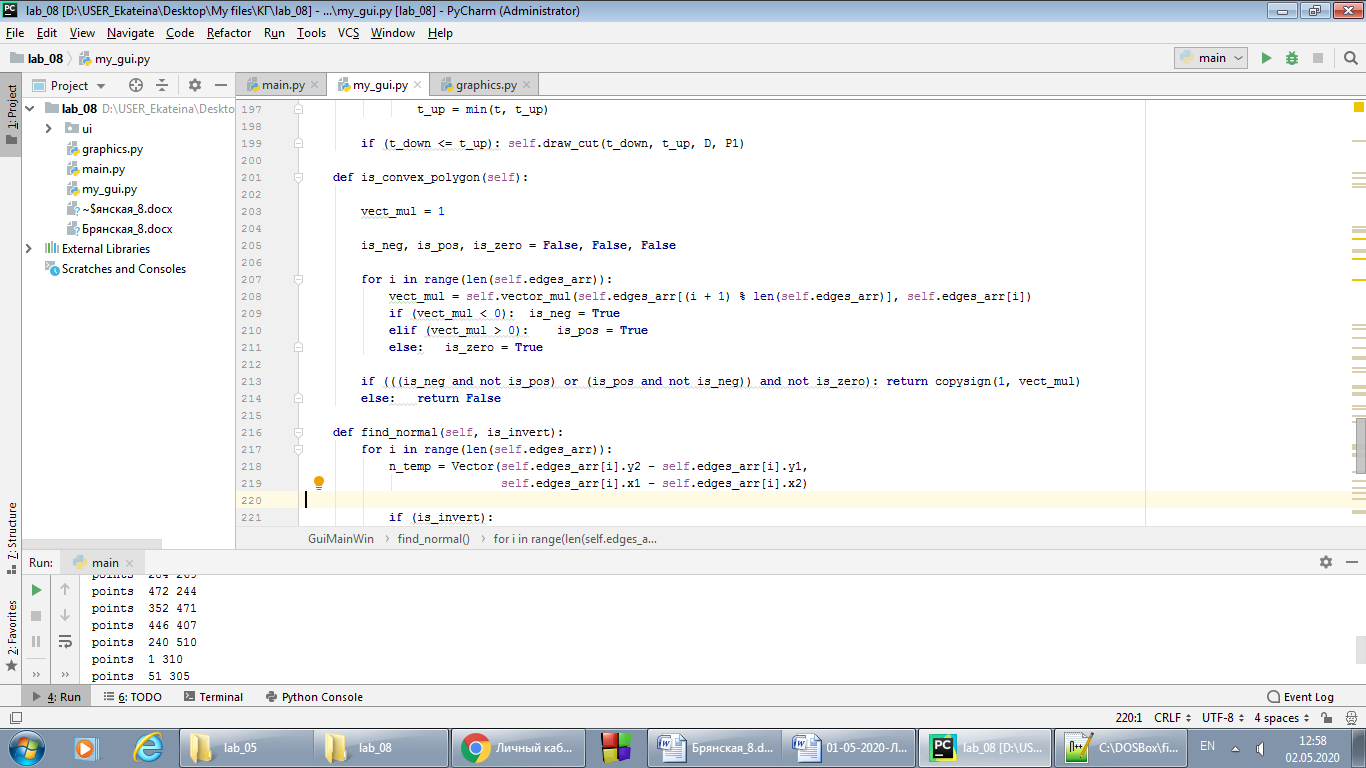
*Код программы:*

Перед тем, как начать работу алгоритма, нужно убедиться в том, что отсекатель – выпуклый многоугольник. Проверка осуществляется путём вычисления векторных произведений смежных сторон (важно учитывать направление, так как важен знак).

Если все значения равны нулю, то многоугольник вырождается в отрезок; если есть как положительные, так и отрицательные значения – многоугольник невыпуклый (дальнейшая работа производиться не будет); и многоугольник выпуклый, если все знаки неотрицательные или неположительные.

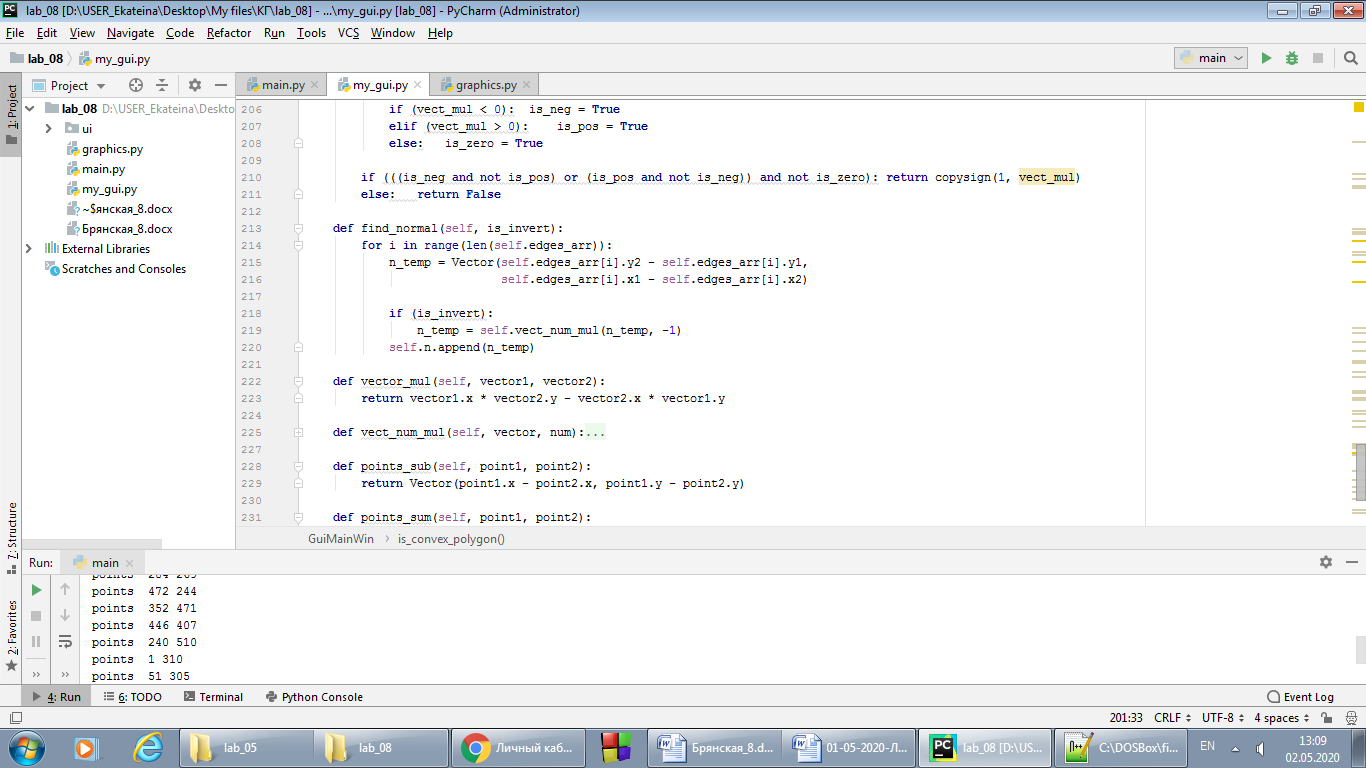
Определение выпуклости:



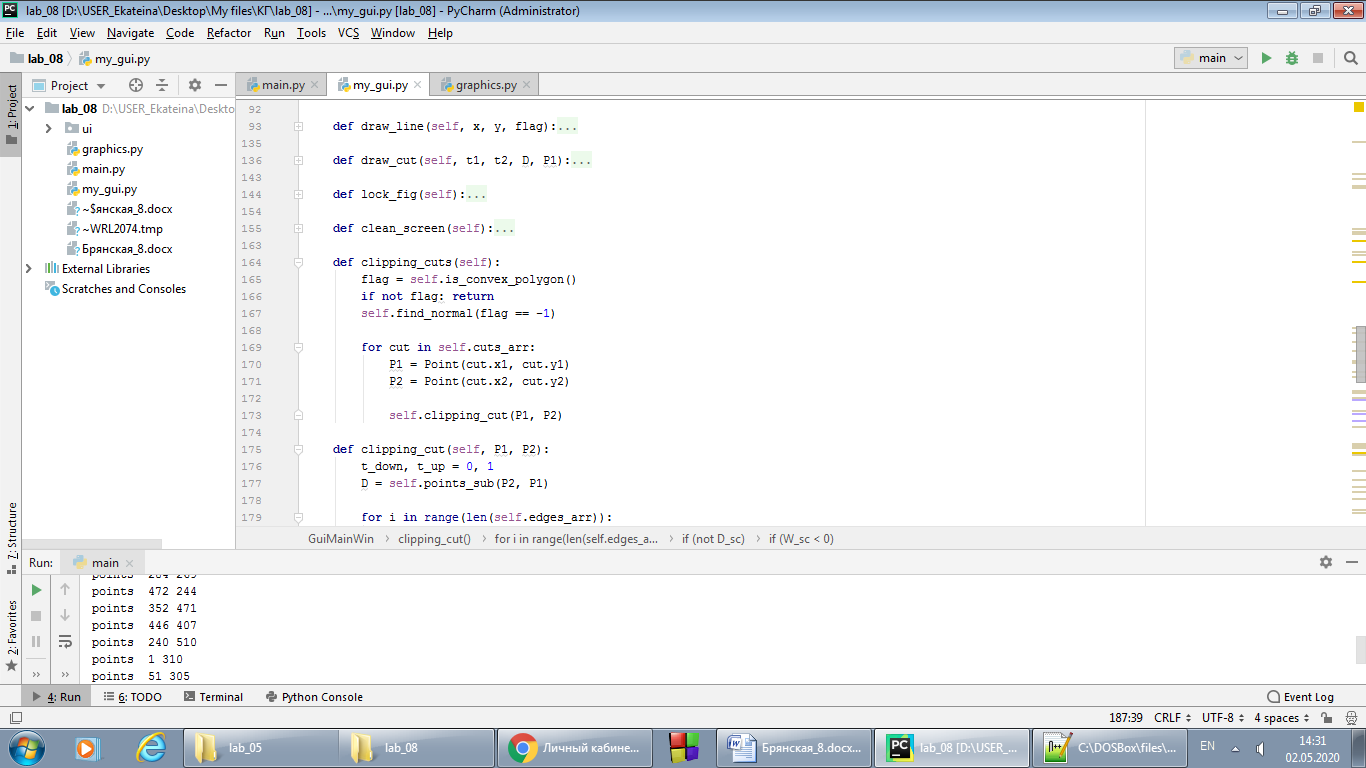


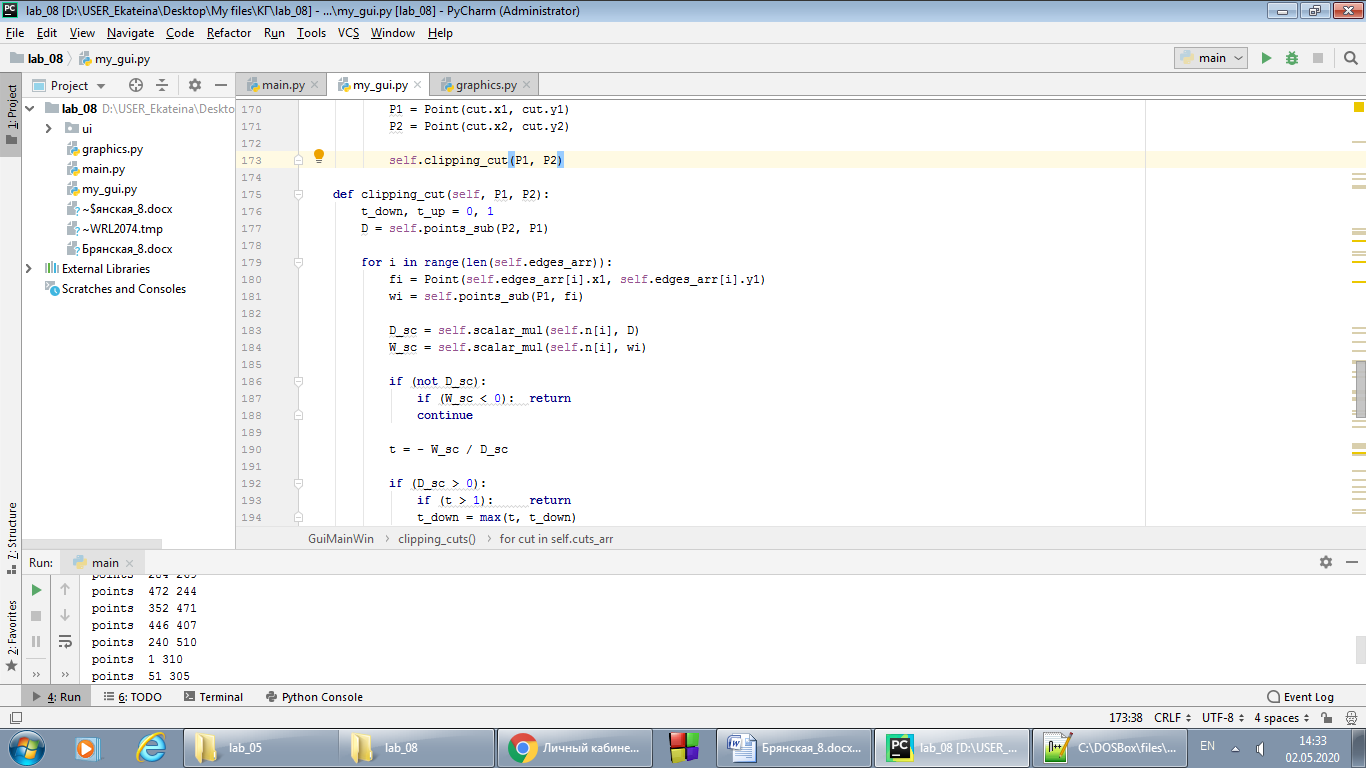
Нахождение нормали:

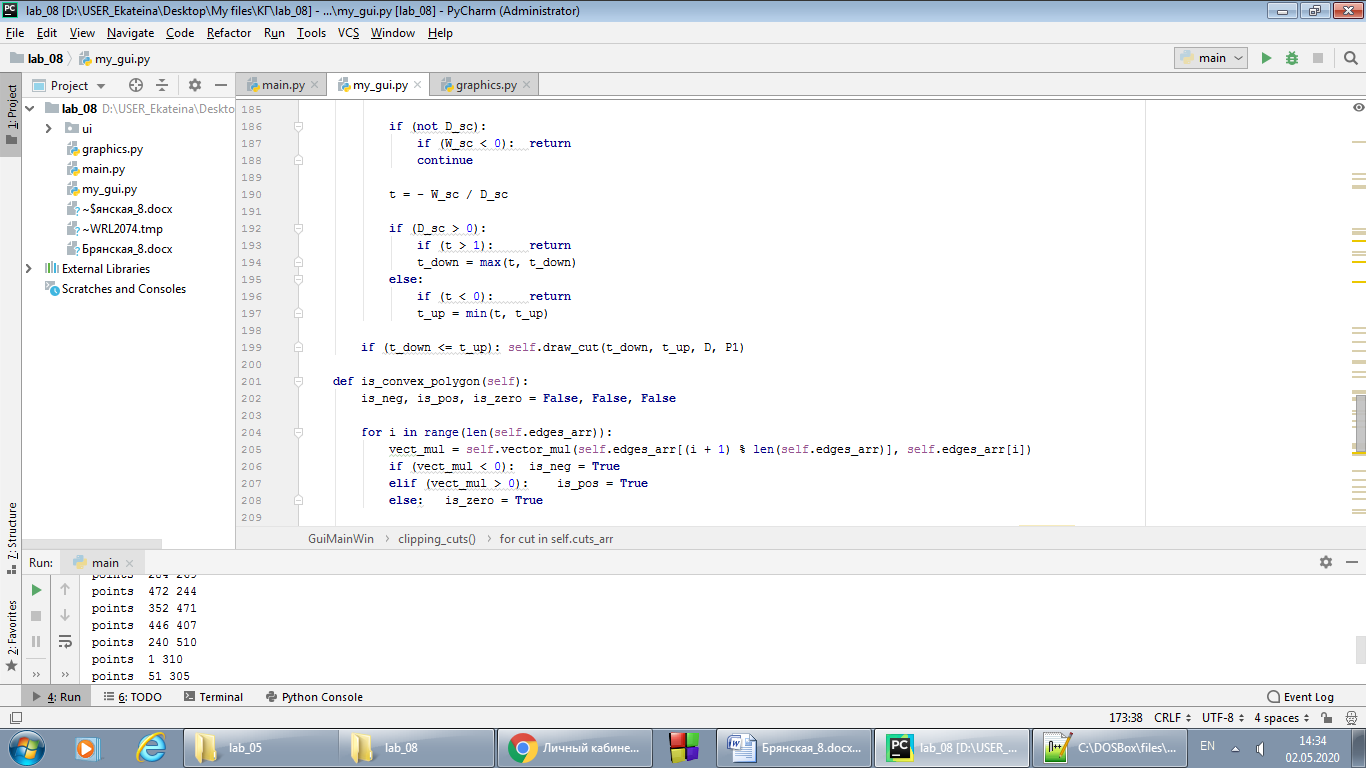
Нормаль находится через уравнение прямой, проходящей через текущее ребро многоугольника. Если векторные произведения сторон отсекателя имели отрицательный знак (на этапе проверки на выпуклость), то получаются внешние нормали, и их надо умножить на -1, чтобы получить внутренние. В другом случае ничего делать не надо, так как нормали получаются внутренними сразу.



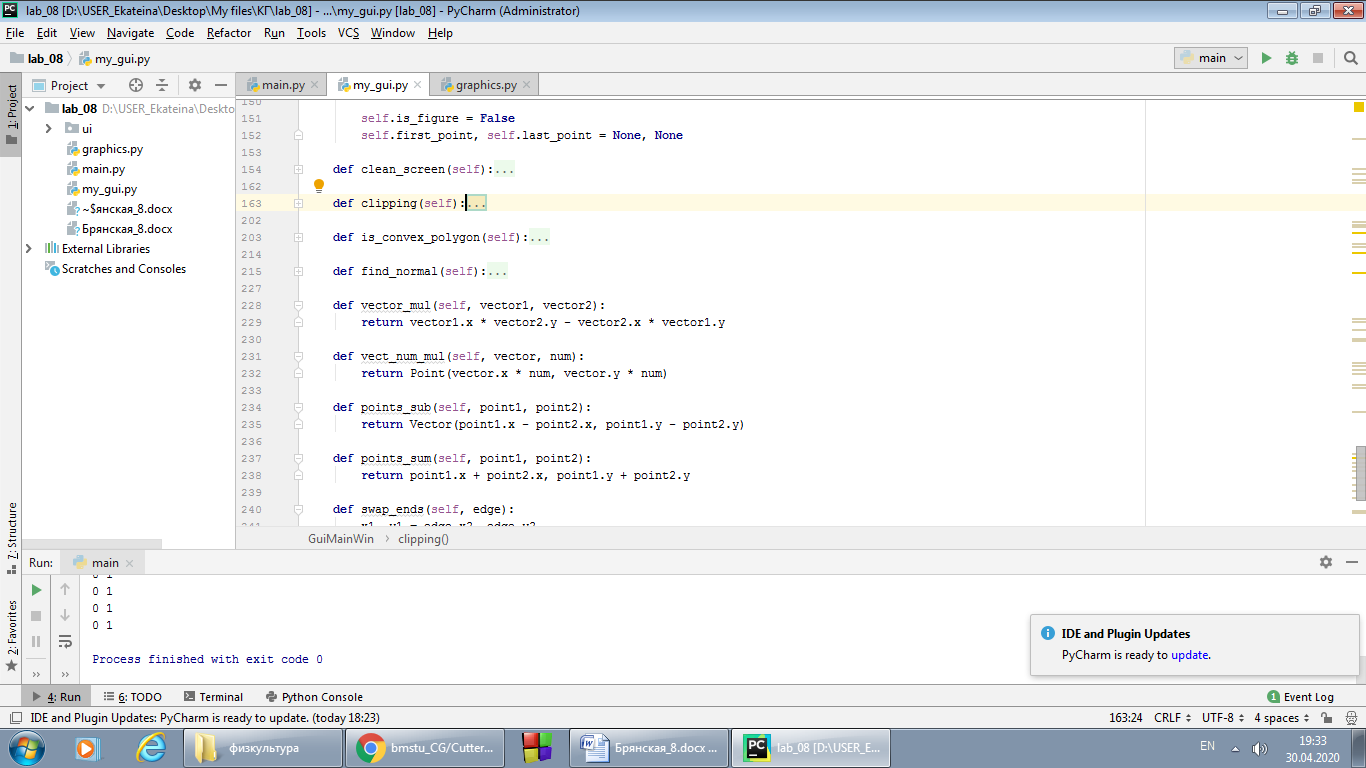
Алгоритм:

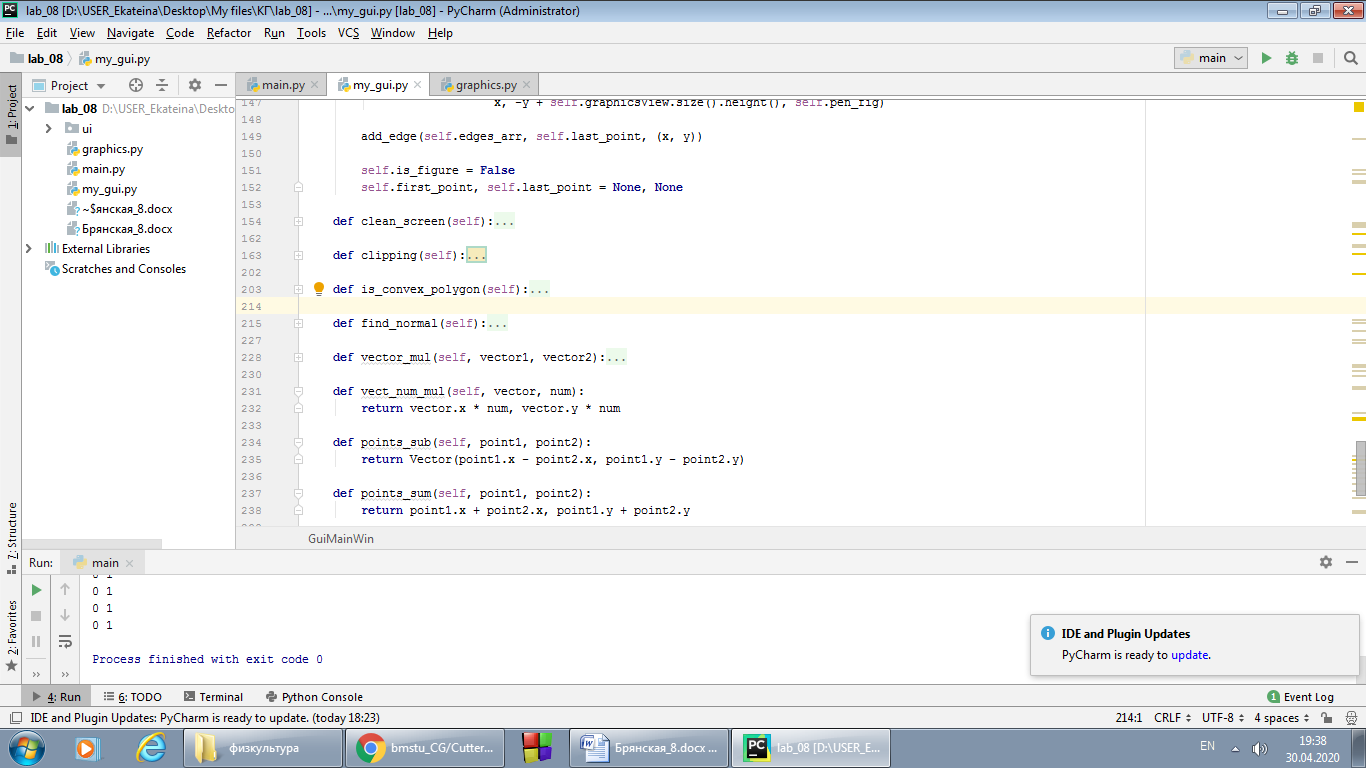


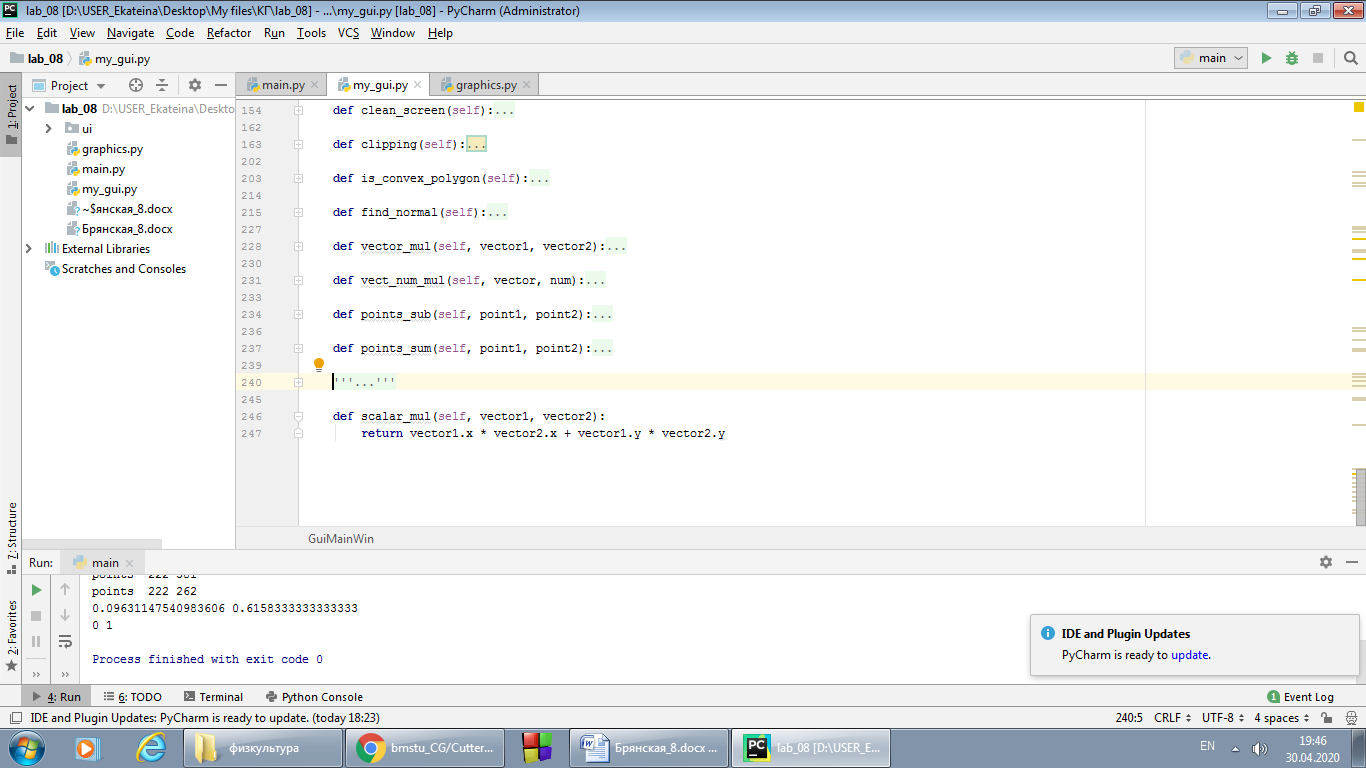




Вспомогательные функции: векторное произведение, умножение вектора на число, скалярное произведение:







*Примеры работы:*

|  |  |
| --- | --- |
|  |  |
|  |  |
|  | *2.png* |
|  |  |
|  |  |
|  |  |

**Вывод**

В ходе лабораторной работы был изучен и реализован алгоритм отсечения отрезка произвольным выпуклым многоугольником (алгоритм Кируса-Бека), работа которого основывается на двух скалярных произведениях.