VerticesCoversion(подходит для выпуклых многоугольников из-за вызова сортировки в конце)

Принимает строку вершин многоугольника (string rawVertices) в формате “x1 y1 x2 y2 … xn yn” и количество вершин (int K). Возвращает массив точек PointF[K], в который помещается преобразованная строка rawVertices. Полученный массив точек сортируется по часовой стрелке методом GetSortedPolygon().

CheckVerticesPairs

Принимает строку вершин многоугольника (string rawVertices) в формате “x1 y1 x2 y2 … xn yn” и количество вершин (int numberOfVertices). Возвращает ложь, если введенное количество точек не кратно 2 и не равно numberOfVertices, иначе истина.

CheckLinearObjectParameters

Принимает строку параметров линейных объектов в формате “w A B C”, где w – ширина линейного объекта (string linObjStr). Возвращает истину, если количество параметров = 4 и w > 0, иначе ложь.

LinearObjectsConversionFromConsole(каждый линейный объект вводится отдельно)

Принимает строку параметров линейных объектов в формате “w A B C”, где w – ширина линейного объекта (string linObjStr), индекс линейного объекта в массиве (int index (по этому индексу будут размещене параметры для нового линейного объекта), массив параметров линейных объектов (float[,] paramsArray), который будет обновляться.

LinearObjectsConversion(линейные объекты вводятся вместе)

Принимает строку параметров линейных объектов в формате “w A B C”, где w – ширина линейного объекта (string linObjStr), количество линейных объектов (int N). Преобразовывает строку linObjStr в массив символов (string[] linObjParamsString) и обходит его в цикле с шагом 4 (количество параметров для одного линейного объекта). Если первый объект (w) в новой итерации > 0, то происходит добавление списка параметров в массив (float[N, 4] linObjParams). Для определения строки массива, куда происходит добавление, используется счетчик (int stringCounterForLinObjArr). Если добавление произошло, счетчик инкрементируется на 1. После обхода массива linObjParamsString производится проверка на равенство счетчика stringCounterForLinObjArr и количества линейных объектов N. Если не равно (количество итоговых полученных объектов меньше изначально указанного N), то возвращается новый массив, в который копируются первые stringCounterForLinObjArr \* 4 параметры линейного объекта из массива linObjParams. Если равно, то возвращается массив linObjParams.

CheckIntersectionsOfPointAndSection

Принимает координаты точки (float xPoint, float yPoint) и координаты концов отрезка (float x1Sec, float x2Sec, float y1Sec, float y2Sec). Возвращает истину, если координаты точки равны координатам начала или конца отрезка. Иначе вычисляется float result(округляется до 3 знаков) и возвращает истину, если result = 0 и (x1Sec < xPoint < x2Sec или x2Sec < xPoint < x1Sec или y1Sec < yPoint < y2Sec или y2Sec < yPoint < y1Sec), иначе ложь.

FindIntersectionPoints

Принимает параметры прямой (float A, float B, float C) и массив точек многоугольника (PointF[] polygon). В каждой итерации цикла (обход вершин многоугольника) формируются переменные с координатами концов отрезка (float x1p, y1p, x2p, y2p) (отрезок – сторона многоугольника). Из этих координат получаются параметры уравнения прямой “A1 B1 C1” методом GetParamsFrom2PairsOfCoordinates(). После вычисляются точки x и y, которые являются точкой пересечения двух прямых, одной заданной и другой полученной. Если эта точка лежит на отрезке (проверка CheckIntersectionsOfPointAndSection()), то она добавляется в список точек пересечения (List<PointF> intersectionPointsList). Заданная прямая может иметь точки пересечения с двумя сторонами многоугольника, если проходит в месте их соединения. Это порождает появление одинаковых значений в списке точек пересечения intersectionPointsList. Чтобы этого не было используется Distinct(). Возвращает список точек пересечения intersectionPointsList.

GetParamsFrom2PairsOfCoordinates

Принимает координаты концов отрезка (float x1, float x2, float y1, float y2). Возвращает массив float[] paramArray с параметрами “A B C” прямой.

Используемое уравнение: .

CoverAllCheck

Принимает массив линейных объектов (float[,] linearObjectsArray), массив точек многоугольника (PointF[] convexPolygon), заданное количество точек многоугольника (int N). Исследует линейные объекты на полное перекрытие заданного многоугольника. Если существует хотя бы один линейный объект, который в себе содержит все точки многоугольника, то цикл прерывается и возвращается истина, иначе ложь.

CreateNewPolygon

Принимает список точек пересечения (List<PointF> intersectionPoints), массив точек многоугольника (PointF[] polygon), параметры прямой (float A, float B, float C), область исследования для прямой (bool plus (если истина, то берутся точки многоугольника с одной стороны прямой, если ложь, то с другой). В списке точек пересечения могут быть только 2 точки, которые в действительности являются точками пересечения с данным многоугольником. Эти точки добавляются в новый список точек многоугольника (List<PointF> newPolygon). Далее происходит обход введенного многоугольника по точкам. Точка добавляется в newPolygon, если она лежит с нужной стороны от прямой. Полученный список точек конвертируется в массив (PointF[] newPolygonArr). У newPolygonArr находится центральная точка GetCenterPoint(), относительно которой происходит сортировка точек по часовой стрелке GetSortedPolygon(). Возвращает массив точек newPolygonArr (новый многоугольник).

GetCenterPoint

Принимает массив точек многоугольника (PointF[] points). Возвращает центральную (среднюю) точку (PointF centerPoint).

FindAreaOfPolygon(для выпуклых многоугольников)

Принимает массив точек многоугольника (PointF[] polygon). Используется формула площади Гаусса. Возвращает площадь многоугольника (double area).

CoverPolygon

Принимает параметры линейного объекта (float A, float B, float CPlus, float CMinus), массив точек многоугольника (PointF[] convexPolygon). Если многоугольник находится внутри линейного объекта, то возвращает истину, иначе ложь.

CreateListOfNotUnderTheLinObjPolygonsForOneLinObj

Принимает список многоугольников (List<PointF[]> polygonsList), параметры линейного объекта (float A, float B, float CPlus, float CMinus). Так как дана средняя линия линейного объекта и его ширина, то границы линейного объекта, можно получить, если к параметру C прибавить и отнять половину ширины (w/2). В результате будем иметь 2 новых объекта (прямых) с такими же параметрами A и B и параметрами CPlus или CMinus, в зависимости от выполненного действия сложения или вычитания. В процессе обхода списка многоугольников, каждый многоугольник заносится в переменную (PointF[] currentPolygon). Заводится флаг polygonHasIntersections = ложь, который меняется на истину, если хотя бы у одной прямой (CPlus или CMinus) есть 2 точки пересечения с многоугольником currentPolygon. Сначала происходит исследование для прямой, которая получается путем прибавления параметра ширины (CPlus) (флаг isPlus = истина). Если есть 2 точки пересечения, то в список новых многоугольников (List<PointF[]> newPolygonsList) добавляется многоугольник, полученный с помощью метода CreateNewPolygon(). Флаг polygonHasIntersections меняется на истину. Потом флаг isPlus меняется на ложь. Происходит исследование для прямой, которая получается путем вычитания параметра ширины (CMinus). Процесс исследования такой же. Если ни одна прямая не имеет точек пересечения с данным многоугольником (флаг polygonHasIntersections = ложь) и данный многоугольник не находится внутри данного линейного объекта, то данный многоугольник добавляется в список многоугольников (newPolygonsList). Возвращает список новых многоугольников (newPolygonsList), которые могли быть получены вследствие отсечения прямыми от многоугольников в заданном списке, или которые не имеют точек пересечения (не отсекаются) и не находятся внутри линейного объекта.

FindAreaUnderLinearObjects

Принимает площадь изначально заданного многоугольника (double mainArea), список многоугольников (List<PointF[]> referencePolygonList). Находит суммарную площадь всех многоугольников в списке. Итоговая площадь округляется до целого числа (просто так, можно убрать по желанию). Возвращает разницу площади заданного многоугольника и суммы площадей многоугольников из списка, которая по заданию является площадью земельного участка под линейными объектами.

CreateReferencePolygonList

Принимает введенный многоугольник (PointF[] enteredPolygon), массив линейных объектов (float[,] linearObjectsArray), количество линейных объектов (int N). Заводится флаг polygonFlag, который обозначает какой список многоугольников будет обновлен. Изначально происходит исследование первого списка многоугольников (List<PointF[]> polygonsList1), в который помещается введенный многоугольник. Переменная referencePolList – вспомогательная, нужна чтобы в конце не делать проверку, какой список обновился последним. При обходе циклом всех линейных объектов происходит считывание их параметров в переменные (float w, A, B, C) и если установлен флаг polygonFlag = 1 и ширина линейного объекта > 0 (w>0), то обновляется второй список многоугольников (List<PointF[]> polygonsList2) путем выполнения CreateListOfNotUnderTheLinObjPolygonsForOneLinObj(). Передаваемым для исследования списком многоугольников является polygonsList1. Далее значение флага polygonFlag приравнивается к 2. Переменная referencePolList приравнивается к polygonsList2. Если установлен флаг polygonFlag = 2 и ширина линейного объекта > 0 (w>0), то обновляется первый список многоугольников (polygonsList1) путем выполнения CreateListOfNotUnderTheLinObjPolygonsForOneLinObj(). Передаваемым для исследования списком многоугольников является polygonsList2. Далее значение флага polygonFlag приравнивается к 1. Переменная referencePolList приравнивается к polygonsList1. Происходящий процесс можно назвать жонглированием списками. Возвращает список отсеченных многоугольников (referencePolList).

NonConsoleInputResult

Принимает количество вершин многоугольника (int K), количество линейных объектов (int N), строку координат вершин многоугольника (string rawVertices), строку параметров линейных объектов (string rawLinObj). Возвращает площадь многоугольника, занятую линейными объектами.

GetSortedPolygon

Принимает массив вершин многоугольника (PointF[] convexPolygon). Находит центральную (среднюю) точку для вершин многоугольника, относительно которой происходит сортировка. В конец отсортированного многоугольника добавляется его первая точка для облегчения обхода его прямых. Возвращает отсортированный по часовой стрелке многоугольник.