Уважаемые господа студенты, внимание!

В силу неких печальных обстоятельств завтра запланированные две пары по КГ не состоятся. Приезжать в корпус не надо.

Во вложении находится файл (этот файл – прим. старосты) для занятий по КГ, который надо изучить и выполнить задания.
В следующий раз запланирован Тест 4 по мотивам лекций 8-9 (Свойства пространства) от 16.11.21, занятия "Фракталы" и заданий из данного файла.

Тест 3 уже проверен, результаты находятся у меня на столе в ауд.216. Работы может взять староста во вторник или среду (в среду я буду в корпусе).

В файле:
Занятие 10 - прочитать, выполнить в тетради со своими данными.
Занятие 11 - прочитать теорию, усвоить, принять к сведению.
Занятие 12 - прочитать, выполнить в тетради со своими данными.

На следующем занятии или консультации показать мне выполненные задания в тетради.

Напоминаю, что семестр заканчивается. Надо отчитываться по работам, например, на консультациях. Пока народу на консультациях мало.

– О.И. Мухин

Задание 10

Проектирование

В трехмерном пространстве появляются новые объекты – плоскости, с которыми нам надо научиться работать. В силу важности этой геометрической конструкции в графических программах используют целых 4 способа вычисления параметров плоскости.

Теория

Уравнение плоскости имеет вид P: Ax+By+Cz+D=0.

Внимание, вопрос! Здесь остановитесь и подумайте, почему эта математическая конструкция описывает не прямую линию в трехмерном пространстве, а плоскость (бесконечное множество определенных прямых линий). И как описать прямую линию в 3D?

Все **способы построения плоскости** разберем на одном и том же примере, чтобы убедиться в одинаковости результатов.

Для этого примем точки с координатами:

A(0,2,9)

B(0,1,10)

C(3,3,6)

D(3,0,9)

Способ 1 «по трем точкам»

Если в пространстве 3D заданы три точки (не лежащие на одной прямой или не совпадающие в одну точку) с тремя координатами (*x,y,z*) каждая, то через них можно провести плоскость, причем только одну. Можете проверить этот факт экспериментально.

Значит, с точки зрения алгебры надо записать 3 уравнения плоскости и подставить в каждое из них числовые координаты (x,y,z) одной из заданных точек по очереди. Поскольку все точки существуют одновременно для одной плоскости, то все три уравнения с коэффициентами A,B,C,D плоскости составляют алгебраическую систему, должны быть верны (И первое, И второе, И третье уравнение выполняются совместно).

$$\begin{cases} Ax_A+By_A+Cz_A+D=0 \\ Ax_B+By_B+Cz_B+D=0 \\ Ax_C+By_C+Cz_C+D=0 \end{cases}$$

Теперь, чтобы определить плоскость, надо найти 4 ее неизвестных параметра (коэффициенты A,B,C,D) из 3 уравнений. В то время, как (x_A,y_A,z_A,x_B,y_B,z_B,x_C,y_C,z_C)- известные числа.

Как найти сразу 4 неизвестных всего из 3 уравнений?

В этой системе уравнений один коэффициент *свободный*.

Разделите левую и правую часть уравнения на D и убедитесь, что на самом деле коэффициентов в уравнении плоскости только 3, а не 4 – один коэффициент определился и равен 1, а неизвестных осталось только 3. 3 уравнения с 3 неизвестными решить можно.

$$\begin{cases} A/D*x_A+B/D*y_A+C/D*z_A+1=0 \\ A/D*x_B+B/D*y_B+C/D*z_B+1=0 \\ A/D*x_C+B/D*y_C+C/D*z_C+1=0 \end{cases}$$

Обозначим, A/D = a, B/D = b, C/D = c.
Значит, один из коэффициентов трехмерной плоскости может быть назначен произвольным числом (но не 0, так как делить на ноль нельзя!). После деления на *D* получим:

$$ax_A+by_A+cz_A=-1$$

$$ax_B+by_B+cz_B=-1$$

$$ax_C+by_C+cz_C=-1$$

Для нашего примера, подставляя координаты точек A,B,C в уравнения:

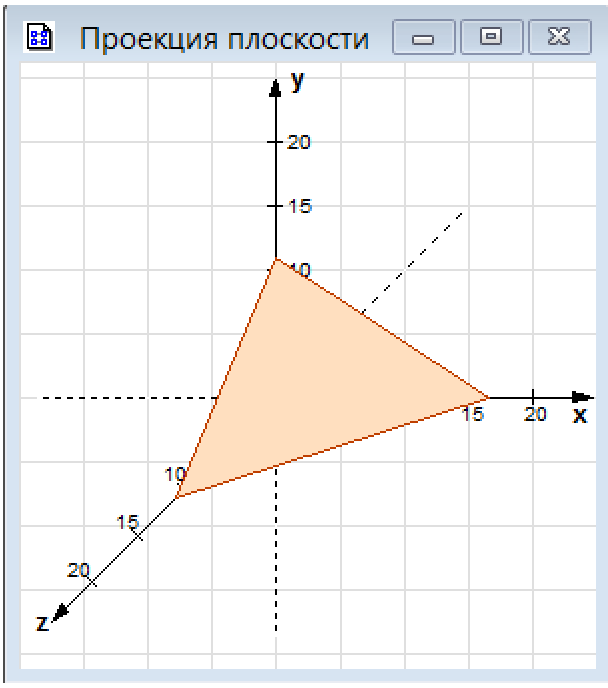
0	2	9		a		-1
0	1	10	*	b	=	-1
3	3	6		c		-1

Примечание: здесь и далее синим цветом выделены матрицы

получаем в числах: *a=-2/33*, *b=-3/33*, *c=-3/33*, *d=1*.
Или P: ***-2x-3y-3z+33=0*** (математическая запись).

Сразу замечу, что для программистов плоскость в теле программы задается как вектор чисел: *(**-2,-3,-3,33**)*, *а не как уравнение. Причина этого в том, что* современные программы не понимают уравнений, а присвоения из уравнений делать чревато. Естественно, эта же плоскость может описана как *(-2/33,-3/33,-3/33,1)* или как *(4,6,6,-66)* и так далее, так как уравнение всегда можно делить на любое число (кроме 0), его свойства от этого не меняются.

Обычно для удобства и единообразия выбирают одну из этих записей с единицей в четвертой компоненте: *(-2/33,-3/33,-3/33,1)*. (Однородные координаты!)



Плоскость P: $-2x-3y-3z+33=0$

Заодно, к слову, у математиков есть еще способ задания плоскости - нормалью к ней: $\underline{n}(-2,-3,-3)$. Нормаль – это вектор, перпендикулярный к плоскости. Если Вы построите данную плоскость и вектор $\underline{n}$, то можете в этом убедиться сами.

Надо сразу сказать, что такая запись говорит только о ***направлении*** плоскости, так как коэффициенты *a,b,c* при переменных *x,y,z* говорят только о наклонах плоскости к осям координат *OX,OY,OZ*. А таких плоскостей в 3D бесконечно много (они параллельны друг другу и образуют слоеную структуру). Вот если договориться, что всегда $D=1$, то нормаль и стандарт понимания коэффициента D вместе взятые, зададут плоскость полностью. В нашем случае: $n(-2/33,-3/33,-3/33)$.

<u>Способ 2 «по двум векторам» или «по нормали»</u>

!!! Поскольку в Word в непосредственной доступности нет верхнего подчеркивания, то <u>вектор</u> я буду обозначать НИЖНИМ подчеркиванием!!! Вот так: <u>с</u> или так <u>АВ</u>. (Б. Гейтс, ведь, - не математик, а математики – не Гейтсы.)

Нормаль к плоскости задается векторным произведением двух векторов.

Пример: $\underline{AB}(0,-1,1)$, $\underline{AC}(3,1,-3)$. Это следует из координат точек нашего примера:

$\underline{AB} = B(0,1,10)-A(0,2,9)= (0,-1,1)$, $\underline{AC} = C(3,3,6)-A(0,2,9)= (3,1,-3)$.

Составляем матрицу векторного произведения (первая строка – орты системы координат, вторая – первый вектор, третья строка – второй вектор, далее вычисляем значение детерминанта):

	$\underline{i}$	$\underline{j}$	$\underline{k}$	
$\underline{n} = \underline{AB}x\underline{AC} =$	0	-1	1	$= 2\underline{i}+3\underline{j}+3\underline{k} = (2,3,3)$
	3	1	-3	

Направления плоскости мы нашли, они заданы нормалью. Найдем теперь конкретную из семейства плоскостей. Для этого достаточно уточнить, в какой именно плоскости из

всех параллельных лежит хотя бы одна из наших заданных точек? Уточним это подстановкой координат любой точки в уравнение плоскости.
Подставим координаты точки *A(0,2,9)* в уравнение плоскости *P:* ***2x+3y+3z+D=0.***
*2*0+3*2+3*9+D=0*
Откуда мгновенно можно определить значение *D=-33.*
В итоге, *P:* ***2x+3y+3z-33=0*** в представлении математика или *(-2/33,-3/33,-3/33,1)* – представление плоскости в теле программы.
Примечание. Что я только что сделал? *A*P=0* – подстановку, а по сути, ***скалярное произведение точки на плоскость***. То есть я соответствующие компоненты точки и плоскости перемножил между собой и произведения сложил (определение скалярного произведения).
Обратите внимание! Ответ совпал с ответом способа 1.

Способ 3 «Метод Ньюэлла» или «по средней нормали»
Теперь приму в расчет все 4 заданные мне точки.
4 точки, если соединить их хотя бы мысленно друг с другом по порядку упоминания, образуют полигон. Каждая точка полигона принадлежит двум ребрам, смежным с ней. Проходя по очереди через все точки по контуру полигона, будем вычислять по очереди на каждой паре этих ребер нормали к плоскостям, образованным этими ребрами.
Нормалей получится много (в каждой точке), и все их мы можем сложить друг с другом. В результате получится средняя нормаль к плоскости. Чем больше нормалей я вычислю, тем точнее будут вычислены коэффициенты плоскости. А поскольку точки «пляшут», координаты

округляются (миллионы действий в секунду на современных компьютерах), то надежность вычисления плоскостей крайне важна. Иначе сцена развалится после нескольких поворотов и сдвигов.

Как известно, согласно великой центральной предельной теореме (ЦПТ) погрешности в суммах гасятся. (Опять я говорю о надежных программах). Поэтому попробуем привлечь свойство ЦПТ к исчислению параметров плоскости:

$$a=\textbf{Сумма}((y_i-y_{i+1})*(z_i+z_{i+1}),1,n-1)+(y_n-y_1)*(z_n+z_1)$$

$$b=\textbf{Сумма}((z_i-z_{i+1})*(x_i+x_{i+1}),1,n-1)+(z_n-z_1)*(x_n+x_1)$$

$$c=\textbf{Сумма}((x_i-x_{i+1})*(y_i+y_{i+1}),1,n-1)+(x_n-x_1)*(y_n+y_1),$$

где n – количество точек замкнутого полигона, ***Сумма(F_i,1,n)*** **–** оператор суммирования (операция численного интегрирования): $F_1+F_2+\ldots+F_i+\ldots+F_n$.

Внимательно присмотревшись к формуле под знаком суммы, можно увидеть, что это формула детерминанта, то есть векторное произведение ребер, смежных точке. (Если помните, векторное произведение – это детерминант, оно же соответствует синусу!). А вся формула – сумма векторных произведений ребер, смежных всем точкам заданного пользователем полигона.

И снова наш пример.

$a=(2-1)(9+10)+(1-3)(10+6)+(3-0)(6+9)+(0-2)(9+9)=-4$

$b=(9-10)(0+0)+(10-6)(0+3)+(6-9)(3+3)+(9-9)(3+0)=-6$

$c=(0-0)(2+1)+(0-3)(1+3)+(3-3)(3+0)+(3-0)(0+2)=-6$

То есть P: $-4x-6y-6z+D=0$.

Подстановкой координат точки в уравнение плоскости определяем значение свободного члена уравнения плоскости (вообще-то, это не что иное как скалярное произведение точки и плоскости – сумма произведений соответствующих компонент):

*A->P: P*A=(-4)*0+(-6)*2+(-6)*9+D=0*

D=66

Математическая запись уравнения искомой плоскости:

$-4x-6y-6z+66=0$

или

$-2x-3y-3z+33=0$.

А программист запишет так:

(-2/33,-3/33,-3/33, 1)

Способ 4 «по объему симплекса»

Как известно, объем плоскости равен нулю. А объем симплекса мы научились уже вычислять (смотри конспект по теме «Симплексы») по координатам 4 точек в 3D.

Но одну точку зададим переменными (*x*,*y*,*z*), чтобы найти все точки плоскости, которые лежат в плоскости:

	x	y	z	Ok	
V = 1/6*	0	2	9	1	***= 0***
	0	1	10	1	
	3	3	6	1	

Раскрывая определитель через миноры, запишем уравнение:

	2	9	1		0	9	1		0	2	1		0	2	9	
x*	1	10	1	-y*	0	10	1	+z*	0	1	1	-1*	0	1	10	=0
	3	6	1		3	6	1		3	3	1		3	3	6	

Итак, *P*: $2x+3y+3z-33=0$.

Как видим, во всех четырех случаях результат (параметры плоскости) получился один и тот же.

Выберите, как будущий инженер, самый быстродействующий из них, самый надежный и самый простой. Обоснуйте свой выбор.

Проектирование

Задайте самостоятельно координаты нескольких точек, нарисуйте их положение в осях координат (x,y,z) в тетради. Найдите всеми четырьмя способами параметры плоскости, в которой лежат эти точки.

Нарисуйте контур плоскости, проходящий через обозначенные точки, закрасьте его.

Найдите по векторам (ребрам полигона) и нарисуйте нормали в каждой из заданных точек плоскости.

Задание 11

Теория

Основой описания плоскости в 3D является работа с векторами.

!!! Поскольку в Word в непосредственной доступности нет верхнего подчеркивания, то вектор я буду обозначать НИЖНИМ подчеркиванием!!! Вот так: c или так AB. (Б. Гейтс, ведь, - не математик, а математики – не Гейтсы.)

Вектор соединяет прямой линией две точки пространства (например, А и В), каждая из которых имеет координаты (x_i,y_i) в пространстве 2D или (x_i,y_i,z_i) в пространстве 3D, где *i* – это А или В. Первой в названии вектора указывается точка его начала, второй – точка конца. Например, вектор из A в B обозначим как $\underline{AB}$.

Координаты вектора: (xB-xA,yB-yA), то есть длины проекций вектора на оси координат. Вектор можно переносить параллельно самому себе. Координаты его от этого не меняются.

Вектор имеет длину (модуль) $|\underline{AB}|$ и направление alfa. Зная координаты вектора, модуль и направление легко можно вычислить.

Модуль задается наикратчайшим расстоянием между точками по оговоренной заранее мере (чаще других используется Евклидова мера): $|\underline{AB}|=\sqrt{((xB-xA)^2+(yB-yA)^2)}$.

Направление задается углами наклона вектора к осям координат, например: $\cos(alfa)=(xB-xA)/|\underline{AB}|$, $\cos(beta)=(yB-yA)/|\underline{AB}|$.

Модуль и направление – характеристики вектора.

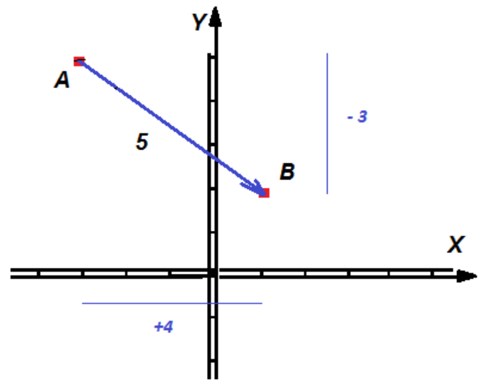
Характеристиками их называют потому, что они неизменны, если вектор передвигать (смещать) параллельно самому себе. (При вращении вектор меняет направление, при масштабировании меняется значение модуля, при проецировании и то, и другое, при сдвиге – точка привязки вектора относительно системы координат).

Например, даны точки в 2D: A(-3,5) и B(1,2). Координаты вектора $\underline{AB}(1-(-3)),(2-5))=\underline{AB}(4,-3)$. Модуль: $|AB|=\sqrt{(4^2+(-3)^2)}=5$. Направление: $\cos(alfa)=(1-(-3))/5=0.8$, $\cos(beta)=(2-5)/5=-0.6$.

$\underline{AB}(4/5,-3/5)$.

Проекции вектора AB на оси координат OX и OY: $AB_x=(xB-xA)=1-(-3)=4$ (проекция - положительное число, так как вектор идет вправо, совпадая с направлением оси OX),

$AB_y=(yB-yA)=2-5=-3$ (проекция – отрицательное число, так как вектор идет вниз, против направления оси OY). По сути координаты вектора задаются его проекциями на оси координат.



В программировании и математике (что одно и то же, ну или почти так) есть действия и сущности, над которыми производятся эти действия. В языках им соответствуют глаголы и существительные. Поэтому займемся теперь действиями, глаголами.

С векторами можно производить **операции**. Наиболее распространенные операции – сложение (вычитание) векторов, умножение (деление) векторов.

Сложение векторов: $\underline{AD}=\underline{AB}+\underline{BD}$.
Слагаемые векторы для проведения операции **сложения** необходимо плоскопараллельно самим себе передвинуть так, чтобы конец первого слагаемого вектора совместился с началом второго слагаемого вектора. (Как было сказано ранее, от их перемещения ни модуль, ни направление результирующего вектора не изменятся). Тогда результирующий вектор соединяет начало первого вектора с концом второго. В нашем случае это точки A и D.

Чтобы узнать направление и модуль результирующего вектора, необходимо:

$|\underline{AD}| = \quad ((xD-xA)^2+(yD-yA)^2)$ и $\cos(alfa)=(xD-xA)/|\underline{AD}|$, $\cos(beta)=(yD-yA)/|\underline{AB}|$.

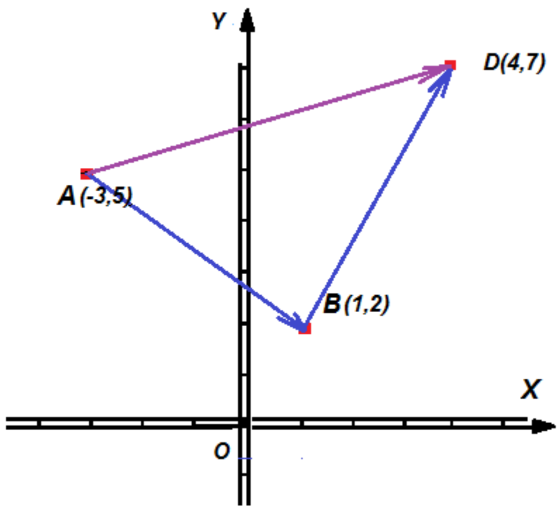
Координаты вектора $\underline{AD}$: $\underline{AD}=\underline{AB}+\underline{BD}=((xB-xA)+(xD-xB),(yB-yA)+(yD-yB))=(xD-xA,yD-yA)$.

Обратите внимание! В компьютерном варианте затратно и не нужно перемещать вектора (менять координаты точек), достаточно сразу работать с координатами векторов. (Сейчас мы думаем о технологичности компьютерных преобразований: быстродействие, простота и надежность графических программ).

Например, зададим точку D(4,7). Вектор $\underline{BD}$(3,5), так как $\underline{BD}$=(4,7)-(1,2).

Тогда $\underline{AD}=\underline{AB}+\underline{BD}=(4,-3)+(3,5)=(4+3,-3+5)=(7,2)$.

$|\underline{AD}|=\sqrt{(7^2+2^2)}=7.28$.



Без обратных операций математика немыслима, в этом ее сила. А необратимость программ – главный недостаток программирования, его слабость. (Эту беду исправляет искусственный интеллект).

Итак, обратная операция сложению - вычитание.

Чтобы **вычесть** из одного вектора другой достаточно сложить эти два вектора, только второй вектор надо взять с обратным знаком, то есть повернуть его направление на противоположное.

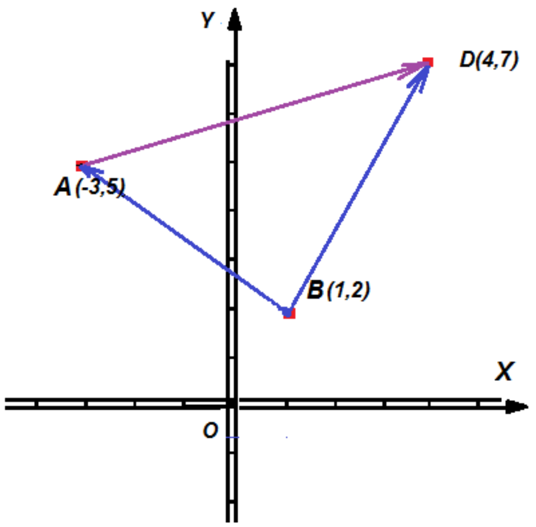
Например, $\underline{AD}=\underline{AB}+\underline{BD}$, то есть $\underline{AB}=\underline{AD}-\underline{BD}=\underline{AD}+(-\underline{BD})$.

Вектор «-BD» образуется из BD ((xD-xB),(yD-yB)) так (0,0)-(xD-xB,yD-yB)=(0-(xD-xB),0-(yD-yB))=(xB-xD,yB-yD). Запись использует координаты точек.

Говоря о характеристиках вектора, все еще проще - модуль остается тем же, а направление меняет знаки на противоположные.

Подставим координаты вектора BD в выражение. $\underline{AB}$=(xD-xA,yD-yA)+(xB-xD,yB-yD)= (xD-xA+xB-xD,yD-yA+yB-yD)=(xB-xA,yB-yA). Итак: $\underline{AD}-\underline{BD}$ = (xB-xA,yB-yA).

Например, $\underline{AB}=\underline{AD}-\underline{BD}=\underline{AD}+(-\underline{BD})=(7,2)+(-(3,5))=(7-3,2-5)=(4,-3)$.



В мире формальных конструкций есть два типа фундаментальных констант (0 и 1) и два типа действий (сложение и умножение). И обратные к ним.

Умножение векторов. Различают скалярное и векторное умножение векторов.

Скалярное умножение двух векторов: AB*AD приводит нас к углу между векторами (точнее, косинусу этого угла): Cos(alfa) ↔ AB*AD. Поясню!

Так как известны два определения (выражения) для скалярного произведения – «сумма попарных произведений проекций векторов-сомножителей на оси координат» (по определению) и «произведение модулей векторов-сомножителей на косинус угла между ними» (что вытекает из знаменитой теоремы косинусов $c^2=a^2+b^2-2*a*b*\cos(C)$), то, записав и то, и другое выражение и приравняв их друг другу, имеем:

|AB|*|AD|*cos(alfa) = ABx*ADx + ABy*Ady.

Откуда сразу получаем:

cos(alfa)= (Abx*Adx+Aby*Ady)/(|AB|*|AD|).

Рецепт готов. Чтобы вычислить угол между двумя векторами достаточно сложить произведения проекций векторов на соответствующие оси координат и разделить сумму на произведение модулей этих векторов.

Например, alfa – угол между векторами(AB,AD)=((4,-3),(7,2)).

Cos(AB,AD)= (Abx*Adx+Aby*Ady)/(|AB|*|AD|)=(4*7+(-3)*2)/(5*7.28)=0,6.

Alfa=arcos(0.6)=53 град.

Векторное умножение двух векторов: $\underline{Ab}\mathbf{x}\underline{AD}$ – соответствует направленному углу между векторами (точнее, синусу этого угла): sin(alfa) ↔ $\underline{Ab}\mathbf{x}\underline{AD}$. Скажу больше: векторное произведение порождает (как всегда) новое понятие: площадь. А у этой площади появляется даже направление!

Важнейшее объяснение!

Например, у ладони есть площадь. У ладони есть также две стороны – внешняя и внутренняя (куда сгибаются пальцы), то есть направление. Площадь внутренней стороны ладони будет положительной величиной, а площадь внешней стороны ладони – отрицательной величиной.

Это обстоятельство дает нам возможность понять в случае с трехмерными телами, где у тела «внутри» и где «снаружи». Иными словами – где внутренние точки тела, а где – внешние к телу точки. Внутренние точки находятся там, где точка пространства принадлежит и пространству, и телу. Внешние точки находятся там, где они принадлежат пространству, но не принадлежат телу.

Трехмерное тело сплошное внутри себя, то есть содержит точки. Если точка принадлежит пространству, но не принадлежит телу, то здесь тела нет, точка находится снаружи тела.

Плоскость (позже будет поверхность) – граница тела и среды. Среда – это пространство, в котором заключено тело. И эта плоскость нам должна быть понятна с точки зрения, где у нее внутренняя сторона, обращенная к телу, а где у нее внешняя сторона, обращенная к среде. Для компьютерной графики это повод понять: рисовать плоскость тела или нет, если глаз находится в заданной точке пространства и смотрит на тело.

Если площадь плоскости **положительна** (для краткости говорят, что «плоскость положительна») и глаз смотрит на положительную плоскость, то рисовать ее **не надо**, так как глаз находится внутри тела. Если глаз смотрит на **отрицательную плоскость**, то рисовать ее **надо**, так как глаз находится вне тела.

Сейчас поясню механизм вычислений площади и ее знака!

Так как известны два выражения для векторного произведения – «детерминант проекций векторов-сомножителей на оси координат» (по определению) и «произведение модулей векторов-сомножителей на синус угла между ними», то:

$$|\underline{\mathbf{AB}}|*|\underline{\mathbf{AD}}|*\mathbf{sin(alfa)}=\mathbf{(Abx*Ady-Aby*Adx)}\,\underline{\mathbf{k}}.$$

Откуда сразу получаем:

$$\mathbf{sin(alfa)}=[\mathbf{(Abx*Ady-Aby*Adx)}/(|\underline{\mathbf{AB}}|*|\underline{\mathbf{AD}}|)]*\underline{\mathbf{k}}.$$

Обратите внимание: и синус, и его угол стали векторами, поскольку имеется направление! Его в вычислении задает вектор $\underline{k}$, он присутствует в формуле. Необычно! Но еще со школы нам говорили, что важно: в каком направлении отсчитывать угол. Вот оно и проявилось. Если аргумент синуса отсчитывается от 0 по часовой стрелке, то направление угла отрицательное. Если аргумент синуса идет от 0 против часовой стрелки, то направление угла положительное.

А вот скалярное произведение, скаляр и косинус направления не имеют, так как, идя от 0 хоть против, хоть по часовой стрелке, косинус имеет один и тот же знак, положителен (90 град.).

k – это третья орта, перпендикулярная одновременно и к первому вектору, и ко второму, угол между которыми мы ищем. Это направление площади! Если направление площади совпадает с ортой k, то площадь положительна, если не совпадает – отрицательна.

Это я ввел в курс дела кратко, теперь стоит пояснить сказанное.

1. Надеюсь, что Вы не забыли, как записывается матрица с векторами:

Abx	Aby
Adx	Ady

и как вычисляется ее детерминант **Abx*Ady-Aby*Adx** (!)

Тут я рекомендую Вам вспомнить (посмотреть в конспекте тему про симплексы) формулу вычисления площади треугольника по координатам трех точек. Сильно поможет понять, о чем это я сегодня …

Там, в зависимости от того, как переставить точки в матрице местами, может получаться то положительная площадь, то отрицательная.

Тогда объяснять этот факт было рано, а сегодня – уже в самый раз. Меняя точки в матрице местами, мы меняем их порядок: 1-2-3 или 1-3-2 при обходе вершин треугольника (говорят «обход по контуру»).

Нарисуйте треугольник, пронумеруйте его точки-вершины и нарисуйте стрелкой их обход в том и другом случае. Убедитесь, что обход против часовой и по часовой стрелке совпадает со знаком площади! Идем по вершинам против движения часовой стрелки – получается положительная площадь, идем по часовой стрелке – получается отрицательная площадь.

Например, A(-3,5,1),B(1,2,1),D(4,7,1) – три точки на плоскости в однородных координатах.

	-3	5	1
S = 0.5*	1	2	1
	4	7	1

det(S)=14.5>0, площадь положительна, орта <u>k</u> совпадает с направлением среднего пальца правой ладони, плоскость невидима.

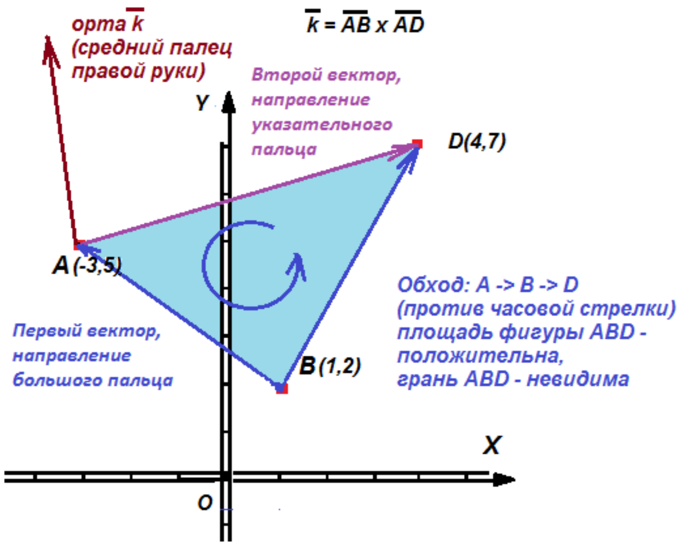
Нарисовав точки A,B,D на плоскости, можно увидеть, что их обход от A к B и дальше от B к D идет против часовой стрелки.

Если переставить точки B и D местами:

	-3	5	1
S = 0.5*	4	7	1
	1	2	1

то det(S)=-14.5<0, площадь отрицательна, орта <u>k</u> имеет направление, обратное к направлению среднего пальца правой ладони, плоскость видима.

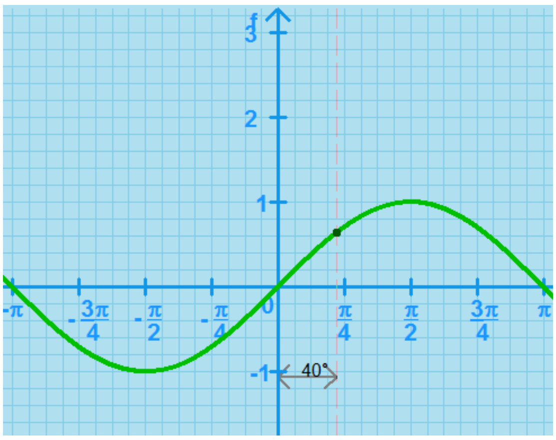
Нарисовав точки A,B,D на плоскости, можно увидеть, что их обход от A к D и дальше от D к B идет по часовой стрелке, а от A к B и от B к D идет против часовой стрелки.

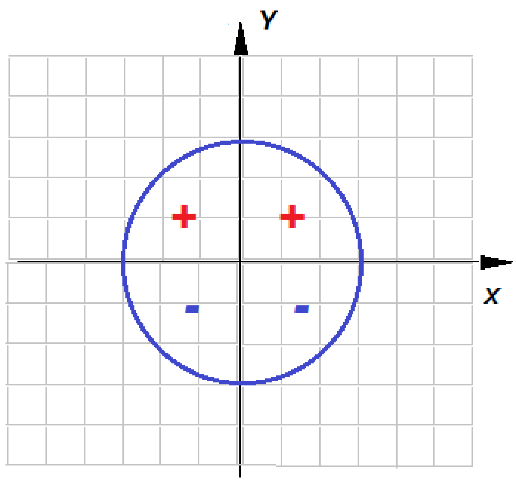


Вывод: площадь – индикатор показателя «внутри-снаружи». По ее знаку можно судить, куда смотрит плоскость (граница тела), куда она направлена!

2. Второе важное замечание.

Синус-то и дает знак угла. Синус положительного угла – положителен, синус отрицательного угла – отрицателен. Это мы знаем, помня, как идет график синуса.



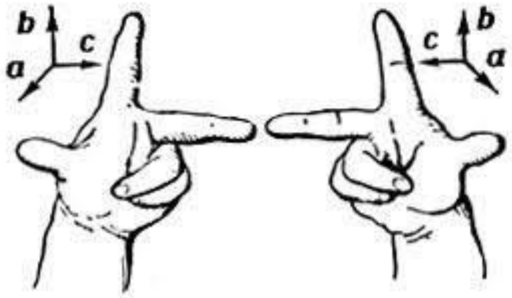


Детерминант с его вычитанием (ABx*ADy-ABy*ADx) подтверждает это. Если ABx*ADy>ABy*ADx, то обход против часовой стрелки и детерминант положителен, и синус положительного угла между двумя векторами положителен. Если ABx*ADy<ABy*ADx, то идет обход по часовой стрелке и детерминант отрицателен, и синус отрицательного угла отрицателен.

Вывод: если обходите против часовой стрелки вершины треугольника (три вершины двух векторов с началом обеих в одной из них), то площадь треугольника положительна, векторное произведение векторов положительно, поверхность ладони внутренняя и невидимая, синус положителен. Если наоборот, то все отрицательное и плоскость, внешняя к телу, видима.

3. Про пальцы на правой руке.

Физики назвали положительное направление правилом буравчика, или правым винтом, или правой тройкой. Направьте большой палец **правой** руки вдоль **ПЕРВОГО** вектора (от 1 ко 2 точке-вершине), указательный – вдоль **ВТОРОГО** вектора (от 1 к 3 точке-вершине), оттопырьте перпендикулярно им средний палец этой же руки. Он покажет по отношению к ладони, где ее положительное направление, где невидимая ее сторона. Средний палец – это орта k, он же указывает на положительное направление ладони. Большой палец – орта i, указательный - орта j.



Конечно, без разницы, что объявлять положительным, а что - отрицательным. Важно, что есть нечто и обратное к нему. Мир дуален (например, «да и нет», «правое и левое», «положительное и отрицательное», «мужское и женское», «большое и малое», «верх и низ», «дискретное и непрерывное», «статическое и подвижное», «мокрое и сухое», «теплое и холодное» и так далее).
Но интересно, что все живое в биологическом мире – левое. А ядовитое для нас – правое, факт, установленный еще Луи Пастером в 19 веке. Что значит, левое? Если плоскость световой волны закручивается влево по ее ходу, проходя через раствор сахара в воде, то сахар – левый. Если вправо, то сахар - правый. То есть правый сахар то же есть, химики синтезировали его. Его молекула - полная зеркальная копия молекулы левого сахара. И правый сахар совсем не сладкий.

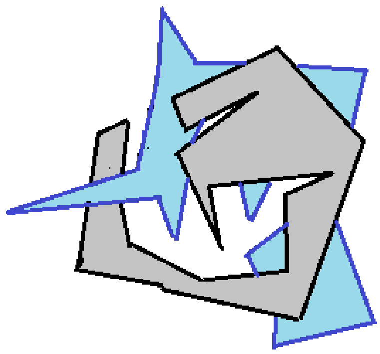
Задание 12

Теория

Поскольку трехмерные объекты чрезвычайно сложны, то невозможно (по ресурсам времени) определить, что именно, из-за чего и насколько является видимым на общей сцене.
Во-первых, для этого пришлось бы сравнивать каждый элемент каждого объекта с каждым элементом каждого другого объекта, что требует написания комбинаторного алгоритма, время выполнения которого растет необычайно быстро.

Например, при 50 простейших объектах на сцене количество комбинаций, которые надо обработать, становится равно $50!=10^{64}$, а при 100 объектах: $100!=10^{157}$. Чтобы Вам было легче понять масштаб этих чисел, замечу, что наша Вселенная существует всего 10^{26} наносекунд. А реальные сцены содержат по 50 000 объектов. Данное явление получило название комбинаторный взрыв.

Во-вторых, крайне сложно понять, как написать универсальный алгоритм для обработки любых возможных сцен. На сегодняшний момент (за 40 лет существования компьютерной графики) изобретено всего 12 алгоритмов такого типа (и каждый имеет свои вполне определенные недостатки). Посмотрите на рисунок и прикиньте, как составить алгоритм, который будет искать пересечения граней и отрисовывать видимые глазу их участки.

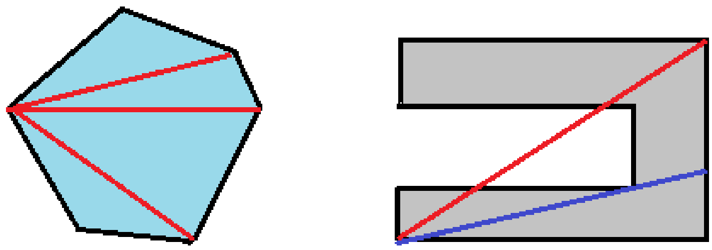


Поскольку прямой путь решения задачи (на машине фон Неймана) невозможен, то поступают следующим образом: разбивают сложные объекты на простые. В результате разбиения форма объектов упрощается, количество объектов на сцене увеличивается.

1. **Сложные объекты разбивают сначала на невыпуклые объекты: плоскости-грани-полигоны**
2. **Далее невыпуклые объекты разбивают на выпуклые**

3. **Выпуклые объекты разбивают на симплексы**

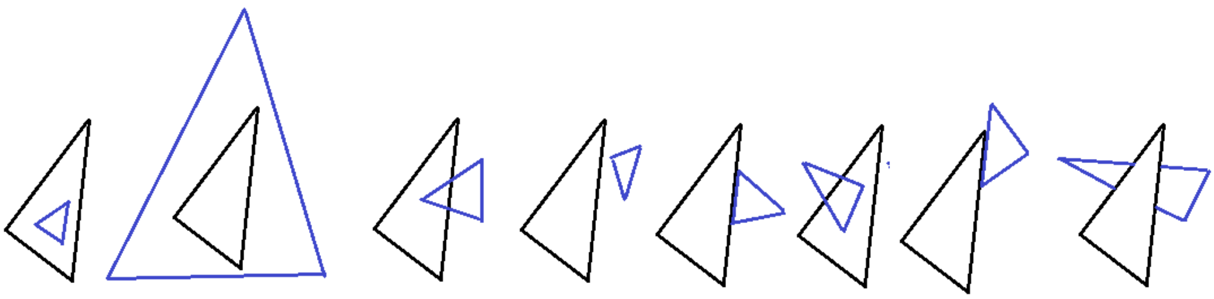
Примечание: сразу разбить любую невыпуклую фигуру на симплексы невозможно.



Пример: выпуклая фигура легко бьется на симплексы, невыпуклая – нет

У симплекса всегда известно количество вершин (и оно минимально) и симплексы всегда выпуклы.

Поэтому количество отношений двух сравниваемых на видимость симплексов сцены всегда одно и то же. У двух треугольников их (основных, обобщенных) 8.



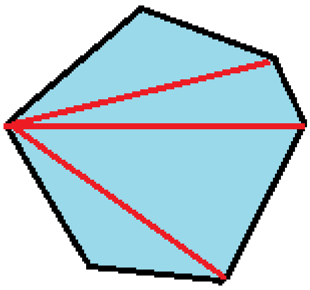
Алгоритм, описывающий каждое из этих отношений, становится стандартным и простым – 8 параллельных веток с 3-4 формулами в каждой ветке. Стандартный и простой алгоритм, выраженный через матричную алгебру, может быть зашит в процессор. На уровне «железа» алгоритм выполняется значительно быстрее, чем с использованием компилятора.

Развитие вычислительной техники с ее двумя ресурсами (память и быстродействие) показывает, что в долгосрочной перспективе выигрывают алгоритмы, которые ориентированы на унификацию процедуры обработки графических сцен и «железо».

Итак, решим 3 задачи, начиная с последней (на примере 2D).

Задача 3 «Разбиение выпуклых объектов на симплексы»

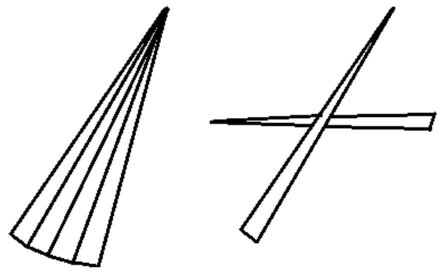
Если фигура выпукла, то, проводя из любой одной ее вершины прямые линии к остальным вершинам фигуры (кроме двух соседних), мы получаем набор треугольников.



Разбиение выпуклой фигуры на симплексы

Задача решена.

Примечание. Алгоритм модифицируется, подбирая в качестве исходной такую вершину, треугольники разбиения от которой получаются менее остроугольные, то есть с более равномерным распределением углов в них. Это повышает точность расчетов сцены.



Контрпример: неудачное разбиение симплекса на треугольники

Задача 2 «Разбиение невыпуклых фигур на выпуклые»

Исходные данные: невыпуклая фигура представлена полигоном с пронумерованными точками вершин, которые соединены ребрами.

Вычисляем векторное произведение V смежных сторон каждой вершины полигона, при этом достаточно вычислить знак векторного произведения. По сути векторное произведение двух ребер – это знак вершины, из которой исходят эти два ребра.

Если знаки всех вершин V равны 0, то это отрезок. Разбивать не надо.

Если знаки вершин – разные, то это невыпуклая фигура. Разбивать надо.

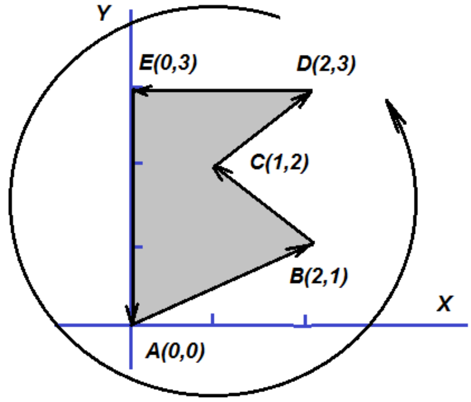
Если знаки вершин одинаковые, то это выпуклая фигура. Ее разбивают на симплексы задачей 3.

Кроме этого следует учесть, что если знаки всех вершин полигона положительные – то грань невидима.

Если знаки всех вершин полигона отрицательные – то грань может быть видима (все зависит от того, где находится глаз наблюдателя по отношению к этой грани).

Пример.

Дан полигон: A(0,0), B(2,1), C(1,2), D(2,3), E(0,3). Ребра задаются в порядке упоминания точек: AB, BC, CD, DE, EA так, как их вносит пользователь на сцену, и, следовательно, в реестр объектов сцены.



Полигон – невыпуклая фигура, подлежащая разбиению на выпуклые. Стрелкой показано направление обхода вершин

Вычислим векторы-ребра, вычитая из координат точки конца вектора координаты точки его начала, например: EA=(0,0)-(0,3)=(0,-3).

Итак, EA(0,-3), AB(2,1), BC(-1,1), CD(1,1), DE(-2,0).

Знак вершины вычисляется как знак векторного произведения смежных с ней ребер. Вершин 5, подсчитаем знак каждой (голубое выделение полей соответствует матрице):

V_A=EA**x**AB=	0	-3	0*1-2*(-3)=6>0 (положительная вершина)
	2	1	

V_B=AB**x**BC=	2	1	2*1-(-1)*1=3>0 (положительная вершина)
	-1	1	

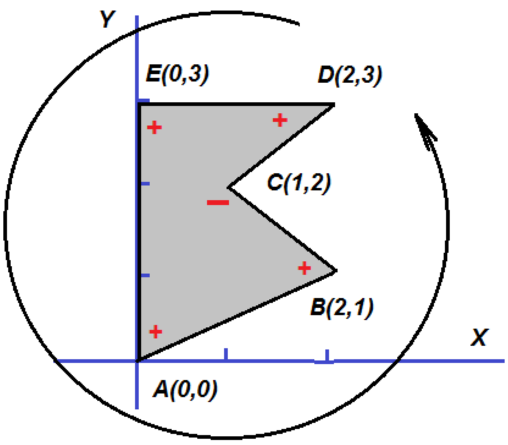
V_C=BC**x**CD=	-1	1	(-1)*1-1*1=**-2<0** (**отрицательная вершина**) -> так как знак вершины сменился, то полигон **невыпуклый** и нуждается в **разбиении**!

	1	1	

$V_D = \underline{CD}\mathbf{x}\underline{DE}$=	1	1	1*0-(-2)*1=2>0 (положительная вершина)
	-2	0	

$V_E = \underline{DE}\mathbf{x}\underline{EA}$=	-2	0	(-2)*(-3)-0*0=6>0(положительная вершина)
	0	-3	

Так как при вычислении знаков вершин произошла смена их знаков, то полигон – невыпуклая фигура и нуждается в разбиении.



Невыпуклая фигура с разметкой знаков вершин

Запускается **процедура разбиения невыпуклых фигур на выпуклые** (сложного на простое).

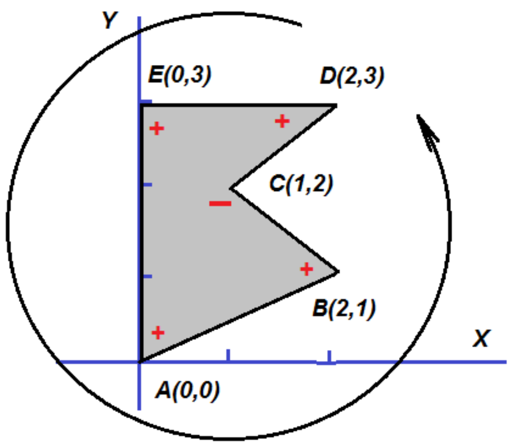
Принцип процедуры: поворот и перенос.

<u>Поместите</u> вершину, которая предваряет в реестре полигона отрицательную вершину в начало координат и *<u>поверните</u>* полигон так, чтобы первое ребро отрицательной вершины совместилось с осью абсцисс. (Найдите угол между осью координат и ребром через косинус).

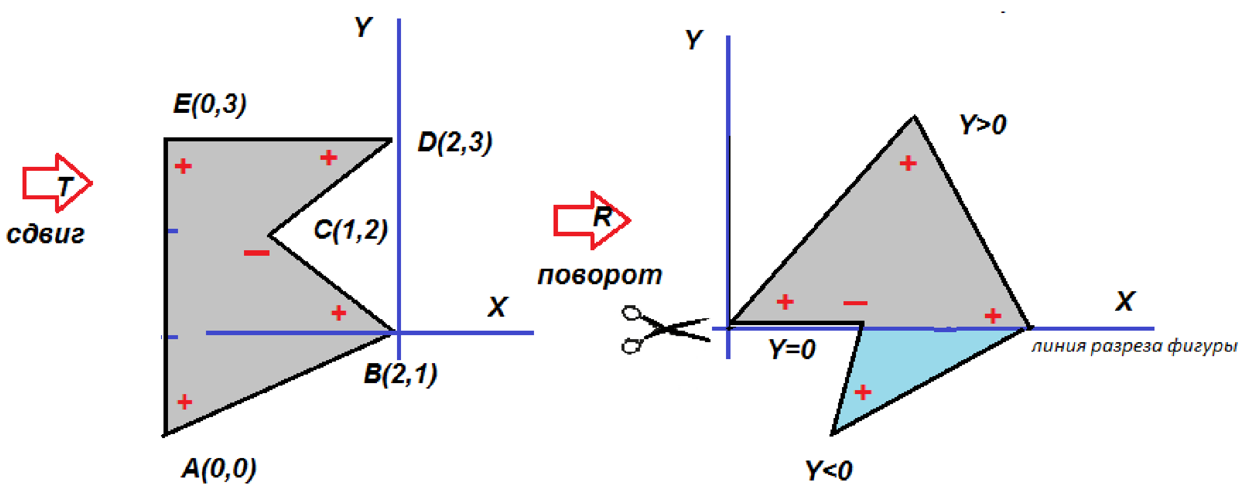
В нашем примере в (0,0) надо поставить вершину B(2,1), сместив полигон на вектор (-2,-1). Далее, повернуть вокруг начала координат все вершины (и ребра) на угол alfa, чтобы совместились вектор BC(-1,1) с осью OX (1,0):

$\cos(alfa)=((-1)*1+1*0)/(\sqrt{((-1)^2+1^2)}*\sqrt{(1^2+0^2)})= \quad -1/\sqrt{2} \quad ->$ alfa=135 град. (против часовой стрелки)

Принцип проверки: рассечь полигон по оси абсцисс. Для этого надо определить знаки y_i всех остальных вершин полигона. Отрицательные знаки (вершина ниже оси OX) – сигнализируют об отделившихся треугольниках или более сложных фигурах.



Исходная невыпуклая фигура, подлежащая разбиению

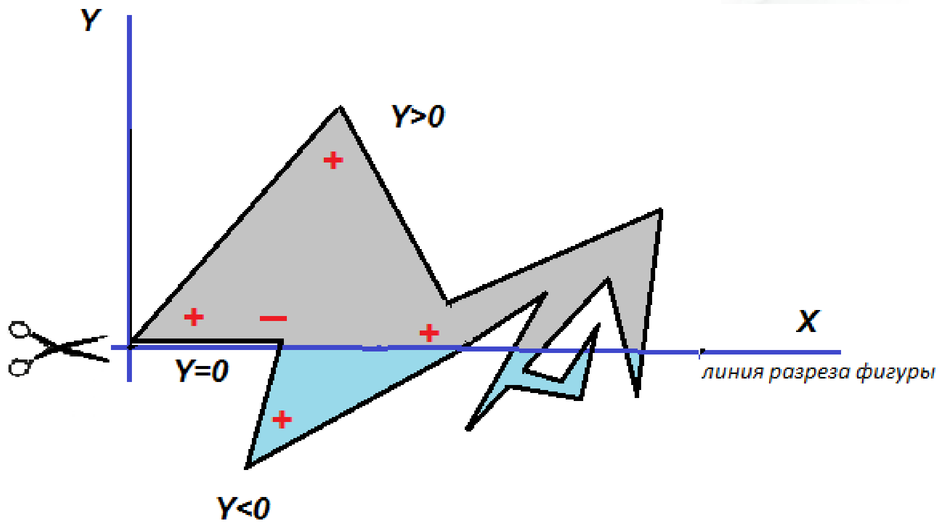


Поворот и перенос фигуры. Определяем отрицательные по Y вершины

Сверху остается исходный (его y_i>=0), но обрезанный полигон. Одновременно снизу образуется новый полигон или даже несколько полигонов с y_i<=0.

Для каждого нового полигона, включая старый, процедура разбиения рекурсивно вызывает сама себя.

Примечание: во время разбиения могут образоваться **несколько (много!!!)** новых фигур. Сколько их будет – заранее сказать **невозможно.** Искать их надо будет перебором всех сторон (отрезков) фигуры. Плюс, к тому же, надо будет искать их точки пересечения с осью X.

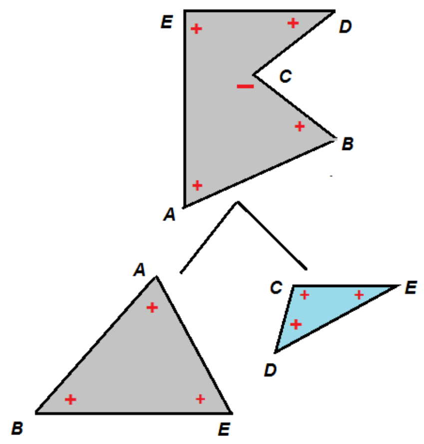


Пример осложнения при разбиении

Каждый вновь образовавшийся полигон снова проверяется на выпуклость.

Если полигон выпуклый, то осуществляется вызов для него задачи 3. Если полигон снова невыпуклый (имеет в своем составе вершины разных знаков), то он снова дробится. Если все полигоны, на которые разбилась невыпуклая фигура, выпуклы, то Задача 2 передает их Задаче 3 для разбиения полигонов на треугольники.

В результате в памяти компьютера строится дерево. Корнем дерева является исходный полигон, а листьями – симплексы.



Дерево разбиения полигона на симплексы

Задание.

Начертите жирной линией произвольного вида полигон из 7-8 вершин в тетради (по клеткам) на половине страницы.

Задайте и подпишите номера его вершин. Определите координаты вершин, занесите их в таблицу данных (реестр полигона). Далее удобно вести расчет в этой же таблице. Определите координаты векторов-сторон полигона. Определите знаки каждой вершины. Запустите для тех вершин, которые отличаются знаком от большинства вершин, процедуру разбиения полигона на невыпуклые фигуры.

Определите смещения и углы поворота. **Проведите** разбиение и отделение полигонов (найдите уравнения новых сторон и координаты новых вершин, как точки их пересечения). При необходимости повторите процедуру рекурсивно до тех пор, пока все фигуры не станут выпуклыми. Разбейте образовавшиеся выпуклые полигоны на симплексы.

Определите после разрезания координаты всех вновь появившихся вершин и их ребер. **Нарисуйте** дерево разбиения своей исходной фигуры на выпуклые фигуры, а потом на симплексы так, как бы оно появилось в памяти компьютера. **Нарисуйте** разбиение тонкими линиями (линии разреза) на своем исходном рисунке с обозначением номеров новых вершин.