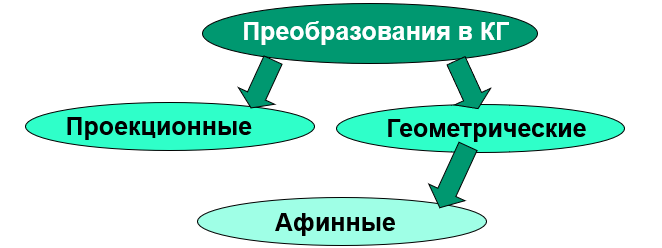
***Геометрические преобразования в*** компьютерной ***графике.***



**В машинной графике представлена большая группа геометрических преобразований (аффинная группа): перенос, поворот, масштабирование. Геометрические преобразования позволяют изменять и визуализировать объект**

**Объекты и геометрические преобразования**

* Компьютерная графика имеет дело со множеством геометрических объектов, таких как точки, многоугольники и многогранники. Все разнообразие геометрических объектов можно свести к ограниченному множеству *простейших сущностей*. Нам понадобятся для этого три базовых типа — **скаляры, точки и векторы**.
* Существуют, по крайней мере, три варианта определения этих сущностей, в зависимости от того, с какой точки зрения их рассматривать

— чисто математической (формальной),

— геометрической

— с точки зрения программной реализации.

* В конечном счете нам придется иметь дело со всеми тремя вариантами

Моделирование

Генерация моделей

линии, кривые, полигоны, сглаженные поверхности

цифровая геометрия

**Описание 2D объектов**

*Lines и Polylines*

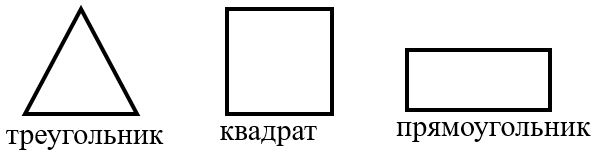
* *Polylines*: линия нарисованная по заданным точкам
* **Соединенные конечные точки дают замкнутую полилинию или *полигон***

Выпуклый и вогнутый многоугольники

*Выпуклый: Для каждой пары точек в многоугольнике линия между ними полностью содержится в многоугольнике.*

*Вогнутый: Не выпуклый: произвольные две точки соединенные в многоугольнике линией не полностью содержавшаяся в многоугольнике*

*Специальные полигоны*

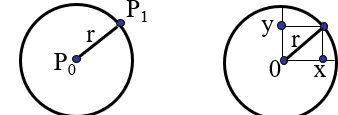


*Окружности*

Состоит из всех точек равноудаленных от единой точки – центра окружности

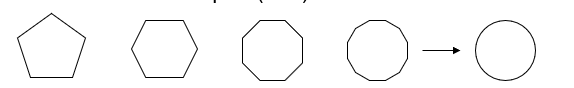
(радиус) *r* = *c,* где *c* const

Уравнение окружности *r2 = x2 + y2*



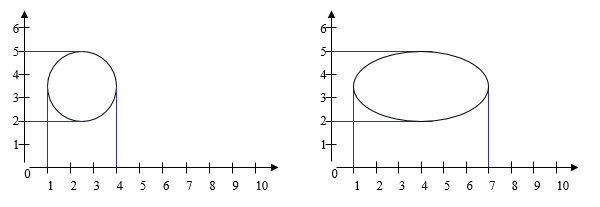
*Окружность как полигон*

Окружность может аппроксимироваться как полигон с множеством сторон (>15)



*Эллипс*

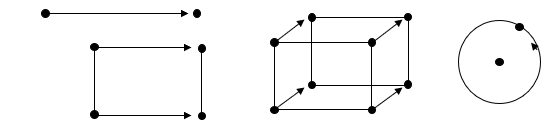
Окружность промасштабированная вдоль осей x или y



*Вершины в движении*

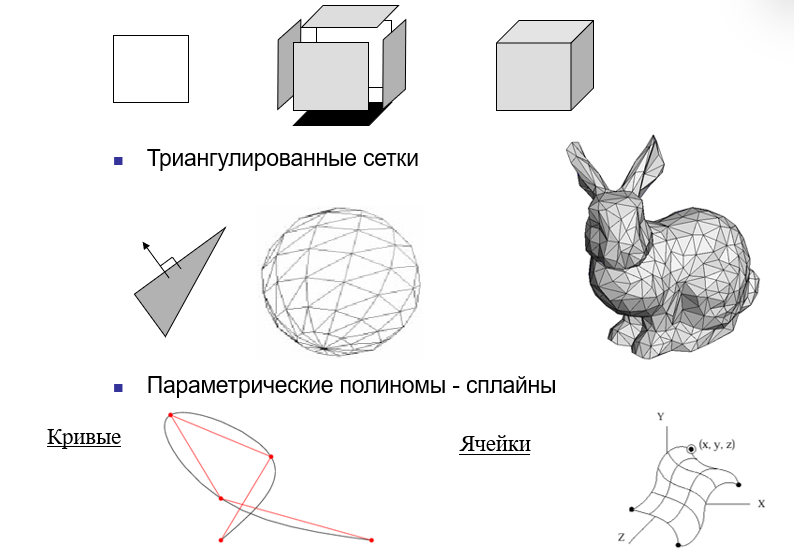
Линия нарисованная трассировкой пути точки при ее движении (1D)

Квадрат нарисованный трассировкой вершин линии при движении во взаимно терпендикулярном направлении (2 D)



Куб нарисованный трассировкой пути вершин квадрата при движении во взаимно терпендикулярном направлении (3 D)

Окружность нарисованная трассировкой движения точки вокруг центра



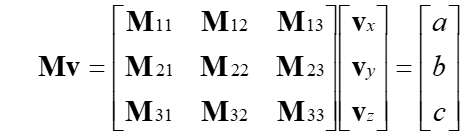
**Теоремы аффинной геометрии идентичны теоремам евклидовой геометрии, в которой важными понятиями являются параллельность и соотношение между параллельными линиями.**

**Аффинные преобразования формируют удобную подсистему билинейных преобразований, так как произведение двух аффинных преобразований также является аффинным. Преобразования связанные с некоторыми изменениями объекта.**

**Базовыми являются следующие преобразования:**

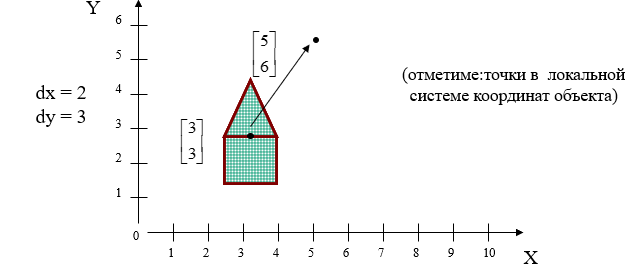
* + Перенос (Move/Translation);
  + Поворот (Rotate);
  + Масштабирование (Scale);
  + Отражение (Reflect);
  + Сдвиг по одной из координат (Shear).
* преобразование объекта = преобразование всех его точек
* преобразование полигона = преобразование всех его вершин

**Умножение матрица-вектор**

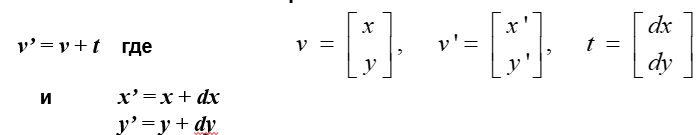


* Линейное преобразование вектора
  + преобразует один вектор в другой
  + сохраняет линейные комбинации
* Любое линейное преобразование м.б. представлено с помощью матриц
  + Масштабирование (scaling)
  + Поворот (rotation)
  + Перенос (translation)

**2D Перенос (Translation)**



**Сложение компонент векторов**



**Для переноса полигонов: перенесите вершины (вектора) и перерисуйте линии между ними**

**Сохраняются длины**

**Сохраняются углы**

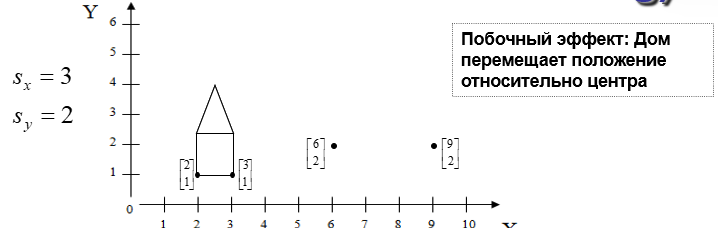
**2D Масштабирование**

* масштабирование координаты означает умножать каждый из ее компонентов на скаляр
* однородное масштабирование означает, что этот скаляр - одинаковый для всех компонентов:

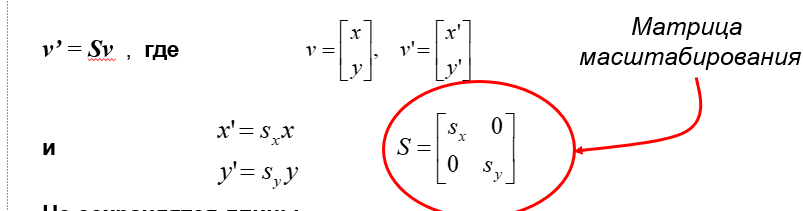
**2D Масштабирование (Scaling)**

* неоднородное масштабирование : различные скаляры для каждого компонента:

**2D Масштабирование (Scaling)**



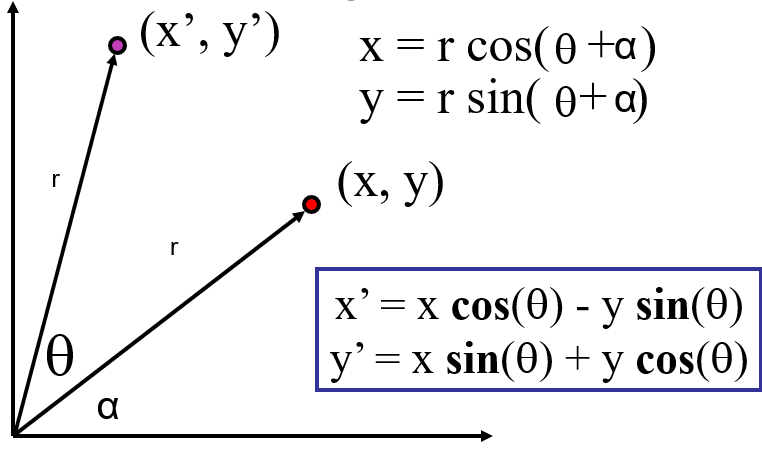
**скалярное умножение векторов**

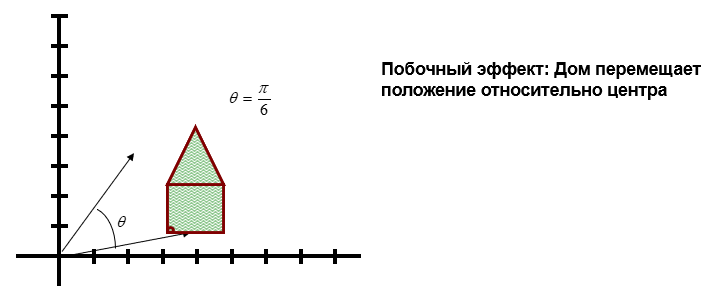
******

**Не сохраняятся длины**

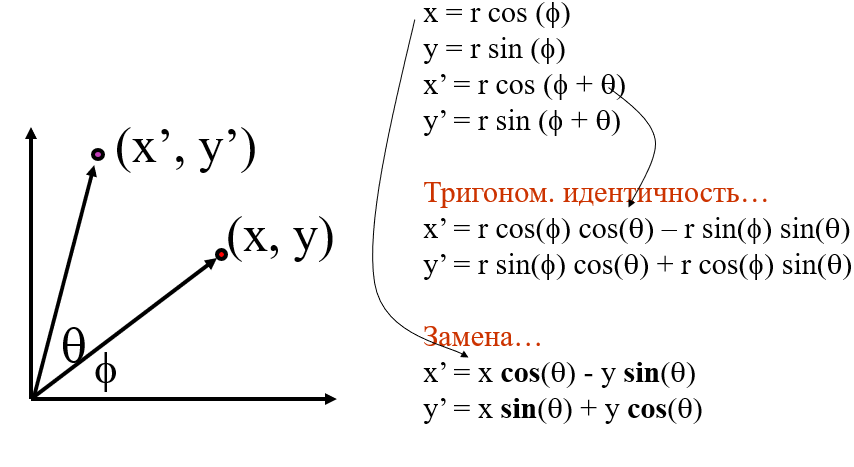
**Не сохраняются углы (кроме тех случаев, когда вычисление однородно)**

**2D поворот**



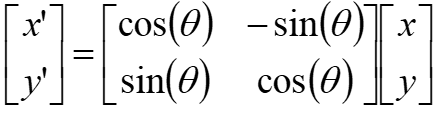


* **Доказательство синусом и формулами суммирования косинуса**
* **Сохраняются длины в объектах и углы между частями объектов**



2D матрица поворота

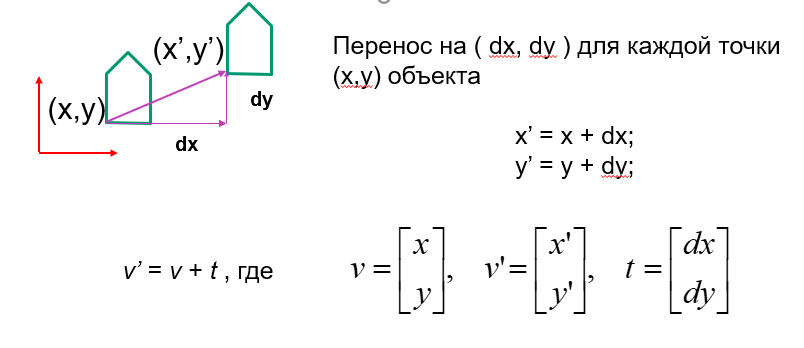
* легко выразить в матричной форме:



x’ линейная комбинация x и y

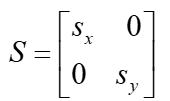
y’ линейная комбинация x и y

**2D перенос**

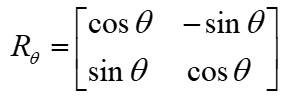


**Матрицы для аффинных преобразований**

* Масштабирование: x’ = xSx, y’ = ySy



* Поворот: x’ = x cos - y sin, y’ = x sin + y cos



**И отдельно для переноса x’ = x + dx**

**y’ = y + dy\**

* **Предположите, что объект не расположен в центре СК и мы хотим его масштабировать и вращать.**
* **Решение: переместитесь в НК, масштабируйте и/или вращайте *его в***  ***местной системе координат*, затем переместите объект обратно.**
* **Эта последовательность предлагает потребность составить Последовательные (составные) преобразования …**

**Однородные координаты (1)**

Однородные координаты - это математический механизм, связанный с определением положения точек в пространстве. Привычный аппарат декартовых координат, не подходит для решения некоторых важных задач в силу следующих соображений:

* В декартовых координатах невозможно описать бесконечно удаленную точку. А многие математические и геометрические концепции значительно упрощаются, если в них используется понятие бесконечности. Например, "бесконечно удаленный источник света".
* С точки зрения алгебраических операций, декартовы координаты не позволяют провести различия межу точками и векторами в пространстве. Действительно, (1,-2,5) - это направление или точка?
* Невозможно использовать унифицированный механизм работы с матрицами для выражения преобразований точек. С помощью матриц 3x3 можно описать вращение и масштабирование,

однако описать смещение (x¢=x+a) нельзя.

* Аналогично, декартовы координаты не позволяют использовать матричную запись для задания перспективного преобразования (проекции) точек.
* Для решения этих проблем используются **однородные координаты**.

**Однородные координаты (2)**

Однородные координаты - это математический механизм, связанный с определением положения точек в пространстве. Привычный аппарат декартовых координат, не подходит для решения некоторых важных задач в силу следующих соображений:

В декартовых координатах невозможно описать бесконечно удаленную точку. А многие математические и геометрические концепции значительно упрощаются, если в них используется понятие бесконечности. Например, "бесконечно удаленный источник света".

С точки зрения алгебраических операций, декартовы координаты не позволяют провести различия межу точками и векторами в пространстве. Действительно, (1,-2,5) - это направление или точка?

Невозможно использовать унифицированный механизм работы с матрицами для выражения преобразований точек. С помощью матриц 3x3 можно описать вращение и масштабирование, однако описать смещение (x¢=x+a) нельзя.

Аналогично, декартовы координаты не позволяют использовать матричную запись для задания перспективного преобразования (проекции) точек.

Для решения этих проблем используются однородные координаты.

Таким образом, преобразование из однородных координат в евклидовы эквивалентно проекции точки на плоскость w=1 вдоль линии, соединяющей точку с началом координат.

Из рисунка также видно, что если преобразование из однородных координат в евклидовы однозначно, то обратное преобразование – нет, потому что все точки на линии, соединяющей точку (x,y,w) и начало координат будут проецироваться в точку (x/w, y/w).

Наша цель состоит в том, чтобы собрать все рассмотренные выше линейные преобразования в одну общую матрицу преобразования. Для этого введем понятие *однородной системы координат*.

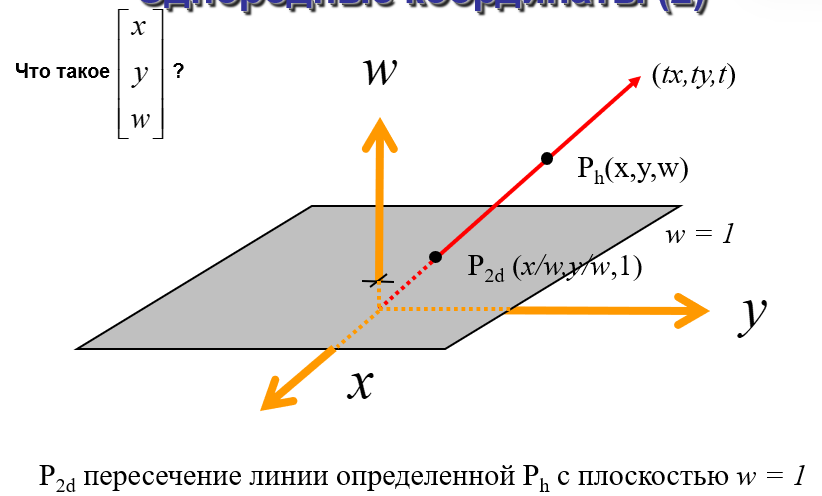
Однородные координаты - это математический механизм, связанный с определением положения точек в пространстве. Привычный аппарат декартовых координат, не подходит для решения некоторых важных задач в силу следующих соображений:

Как было сказано выше, с помощью однородных координат можно легко описывать бесконечность.  
  
 Рассмотрим точку с однородными координатами (x,y,w). Ей соответствует точка с евклидовыми координатами (x/w, y/w).

Зафиксируем x и y и устремим w к нулю. Точка (x/w, y/w) будет удаляться все дальше и дальше в бесконечность в направлении (x,y).

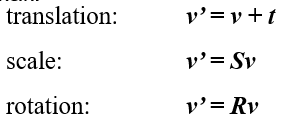
Когда w станет нулем, (x/w, y/w) уходит в бесконечность.

Следовательно, однородные координаты (x,y,0) – *идеальная точка* (ideal point) или, по-другому, *точка в бесконечности* (point at infinity) по направлению (x,y).

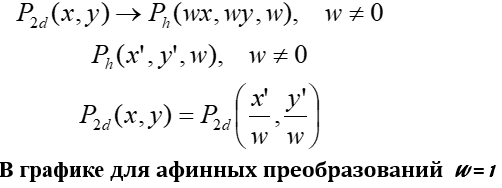


**Однородные координаты**

* Перенос, масштабирование и вращение выражены (неоднородно, негомогенно) как:

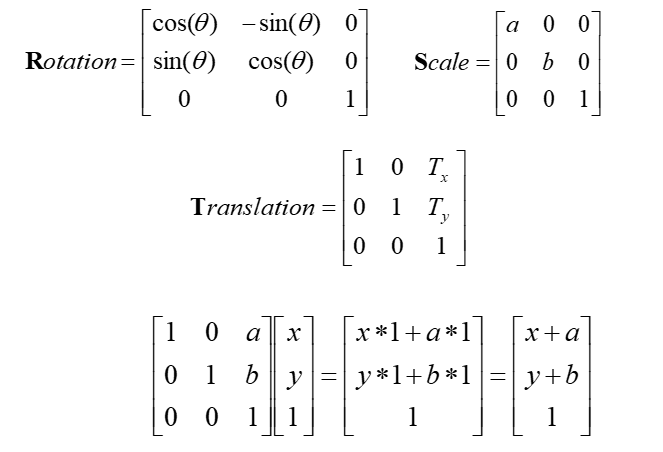


* Композицию преобразований трудным выразить, пока перенос не выражен как матричное умножение
* Однородные координаты позволяют всем трем преобразованиям быть выраженными однородно, позволяя композицию путем умножение матриц 3x3.



**Однородные координаты (7)**

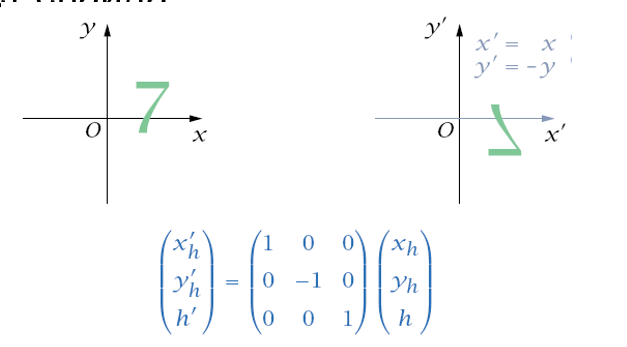
* наши 2D матрицы преобразования теперь 3x3:



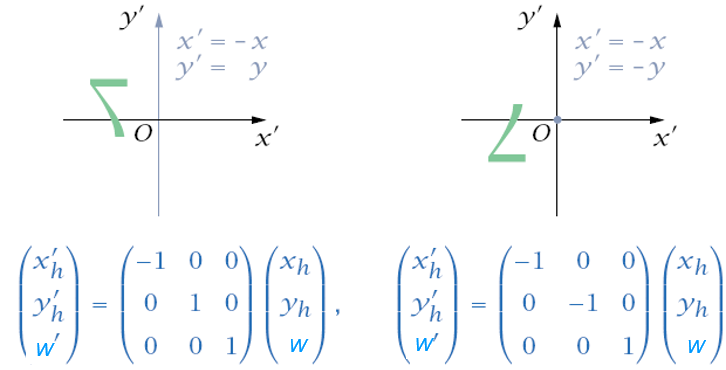
**Однородные координаты (8)**

* Набор преобразований, которые мы изучали, известен как *"аффинные" преобразования,* их свойства*:*
  + сохранены прямые линии;
  + сохранены параллельные линии
  + отношения длин вдоль сохраненных линий сохранены

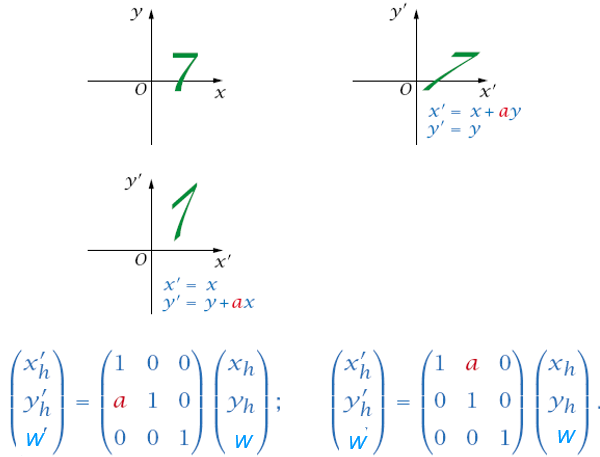
**Отражение (1)**



**Отражение (2)**



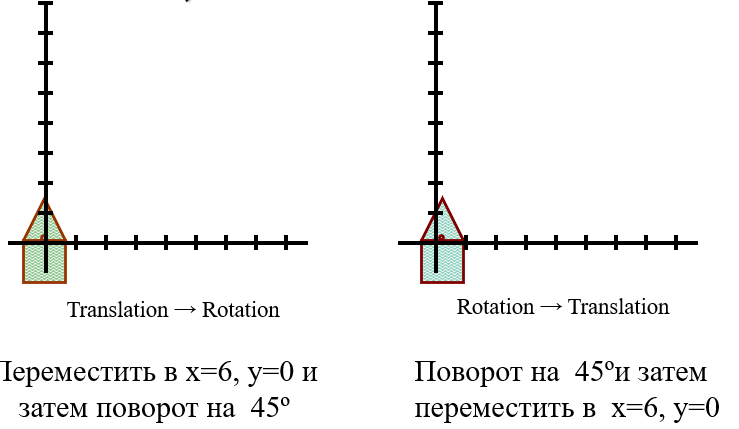
**Смещенние (**Shear)



**Порядок выполнения преобразований (1)**

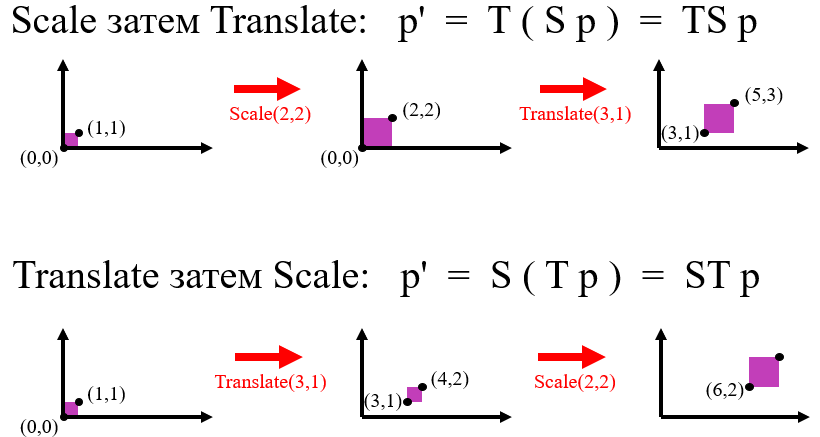
**Преобразования не коммутативны!!**

**Ta Tb = Tb Ta, но Ra Rb != Rb Ra и Ta Rb != Rb Ta**

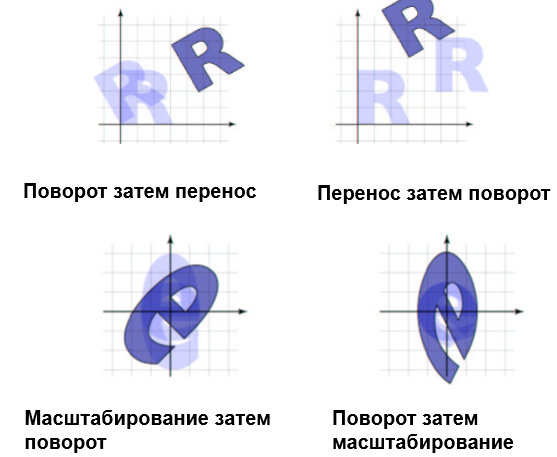


**Порядок выполнения преобразований**

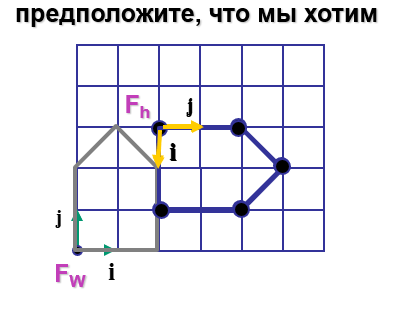
**Преобразования не коммутативны!! Тa Sb != Sb Ta**

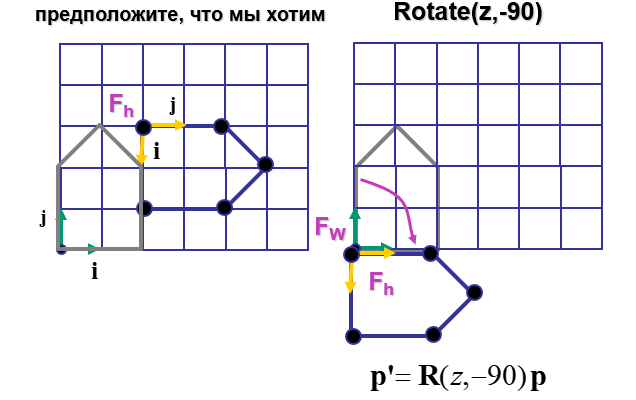


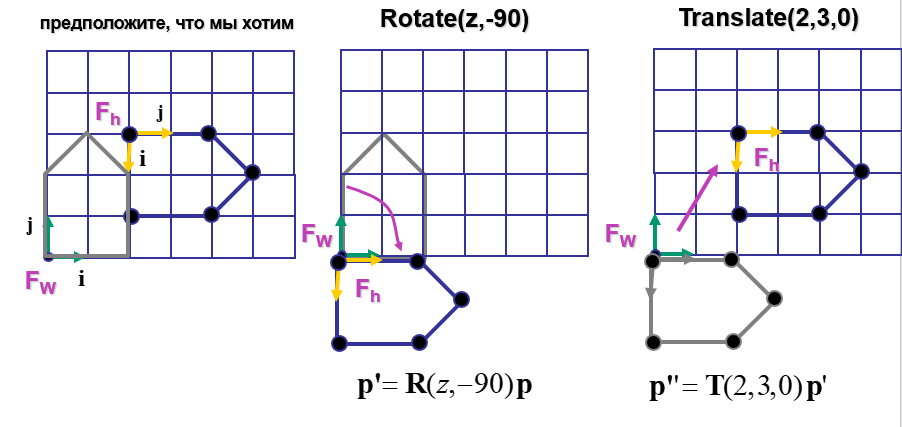
**Порядок выполнения преобразований**

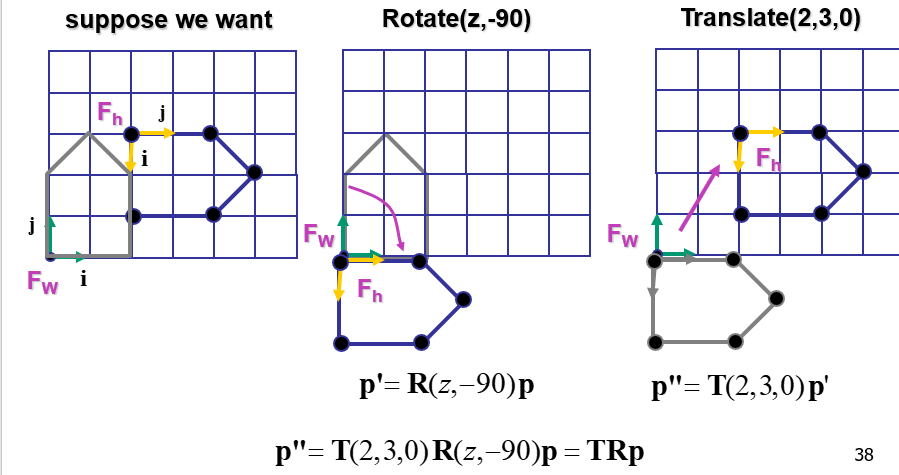


**Создание преобразований**

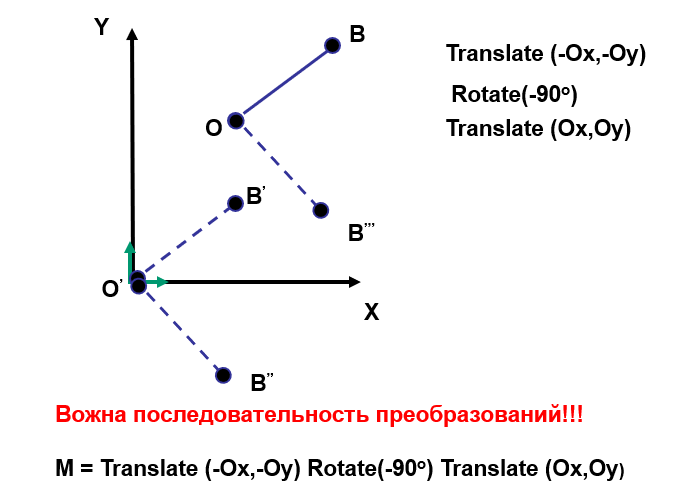




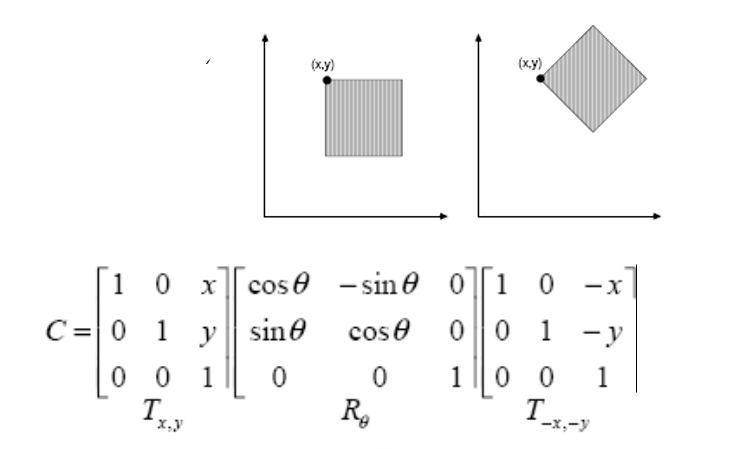




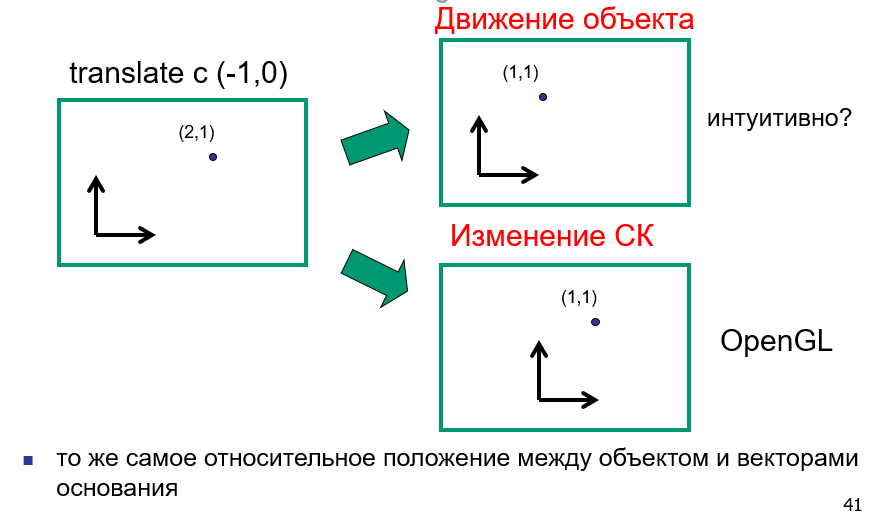
**Поворот вокруг произвольной точки О**



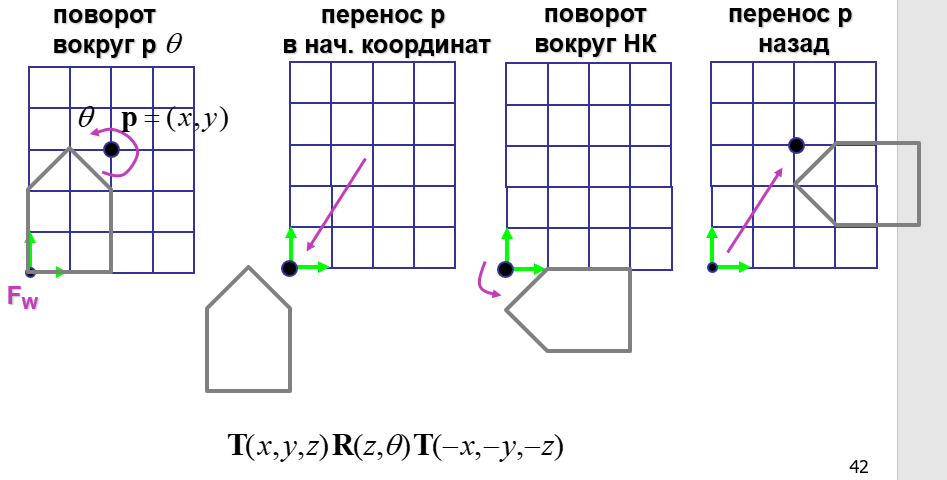
* Вращение объекта на угол θ вокруг точки (x, y)
  + Translate (x, y) в начало координат
  + Rotate
  + Translate в точку (x, y)



**Интерпретация Преобразований**



**Поворот вокруг точки: вращаем объект**



**Вращение: Изменение Систем координат**

* **тот же самый пример: вращение вокруг произвольного центра**
  + **шаг 1: переведите систему координат к центру вращения**
  + **шаг 2: выполните поворот**

**шаг 3: перевести назад систему координат**

**Общий подход к композиции преобразований**

* **преобразование геометрии объектов сцены в такую систему координат, где операция становится более простой;**
* **выполните операцию;**
* **преобразуйте геометрию объектов сцены назад к первоначальной системе координат.**

**Преобразование системы координат (СК)**

Зачем осуществлять преобразование системы координат?

* Часто объекты представлены в «удобной» или «естественной» системе координат
* Чтобы нарисовать объект с преобразованием Т, мы можем:

Зачем осуществлять преобразование системы координат?

* Часто объекты представлены в «удобной» или «естественной» системе координат
* Чтобы нарисовать объект с преобразованием Т, мы можем:

**Преобразование системы координат(CК)**

**Скажите системе однажды, как выводить объект, затем рисуйте в преобразованной CК, чтобы преобразовать объект.**

**Дом, выведенный в CК , которая была преобразована - translated, rotated и scaled**

**Преобразование системы координат(CК)**

В общем случае, если CS1 преобразована матрицей М, чтобы сформировать CS2, точка P в CS2 представлена с помощью MP в CS1

* Если CК1 преобразована последовательно с помощью M1, M2, …, Мn, чтобы сформировать CКn+1, то точка P в CКn+1 представлена с помощью M1 M2 … Мn P в CК1.
* Чтобы сформировать сложное преобразование между CКs, Вы последовательно умножаете на каждую соответствующую матрицу преобразования.