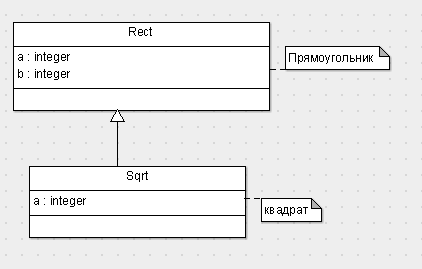
**Solid LSP**

LSP – поведение наследников не должно противоречить поведению базового класса

Думаю, проще показать на примере, предположим, что мы пишем test для нашего кода и у нас есть вот такая структура классов

(Так нельзя)



Мы написали test и говорим, вот, чтобы проверить тестер нашего кода давайте создадим List с 4 прямоугольниками и 1 квадратом, переберем их и изменим их стороны с помощью функции SetA(int a) и SetB(int b). Все пишут SetA(67) – все хорошо, но когда прописывают SetB(7), то у всех прямоугольников сторона меняется, а у квадрата ее тупа нет и мы получаем NullReferensException в ебало. Это пример, который наглядно демонстрирует нарушение принципа LSP.

!!! Также я вам запрещаю выкидывать null или исключение в методах, которые мы не используем в наследнике