ArrayToSort ...*// Надо отсортировать*New SortedArray *// В будующем здесь будет отсортированые штуки  
  
//начнем с 0 штуки***int** iпоSortedArray=0  
*// сперва идут прямоугольники патом квадраты  
  
//Вначале стоят прямоугольники***for**(GeometObject ГдетоПямоугольник: ArrayToSort ){  
 *//Записать в SortedArray вначале пямоугольники* **if**( ГдетоПямоугольник **instanceof** Прямоугольник){  
 SortedArray[iпоSortedArray]=ГдетоПямоугольник;  
  
 *//прямоугольник на 0 месте* iпоSortedArray++;*// следующий на 1 месте* }  
}  
*//Все прямоугольники на начале  
//SortedArray заполнен прямоугольниками  
//iпоSortedArray == гдето в середине  
  
//Вконце стоят квадраты***for**(GeometObject ГдетоКвадрат: ArrayToSort ){  
 *//Записать в SortedArray вначале пямоугольники* **if**( ГдетоКвадрати **instanceof** Квадрат){  
 SortedArray[iпоSortedArray]=ГдетоКвадрати;  
  
 *//прямоугольник на 0. месте, следующий на 1.* iпоSortedArray++;  
 }  
}  
  
*// Теперь прямоугольники вначале, квадрати вконце*SortedArray заполнен, никто так и не сортирован  
*// сортируем прямоугольки  
// после них квадраты  
  
// если соседей проверять значит обязятельно начинать  
// или заканчивать на +1 больше в начале(на -1 меньше в конце)  
// если проверять 2 предыдущих - +2 в начале, -2 в конце***for**(**int** i=1; i<SortedArray.length; i++) {  
  
 *//только прямоугольники сортировать* **if** (SortedArray[i] **instanceof** Прямоугольник){  
 *//Проверить соседа сверху(слева)( i-1 )  
 //Надо Точно знать класс для таких штук* **if**(--**true**?  
 ( (Прямоугольник) SortedArray[i-1]).length/width >=  
 ( (Прямоугольник) SortedArray[i]).length/width  
  
 --**true**?){ Правда--> Тогда поменять местами  
  
 }  
 *// i++* }  
  
}  
*//Теперь прямоуголькики сортированы, тоже самое с квадратами***for**(**int** i=1; i<SortedArray.length; i++) {  
 *//сортировать соседние квадрати SortedArray[i-1] ... и SortedArray[i]*}  
  
*// Готово*SortedArray сортирован.