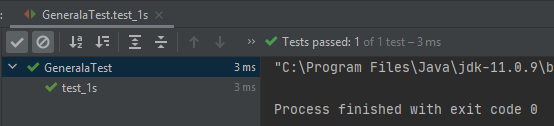
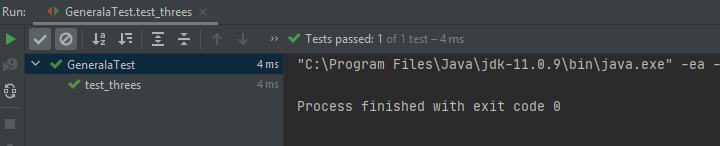
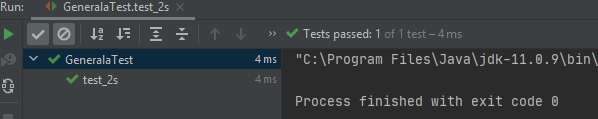
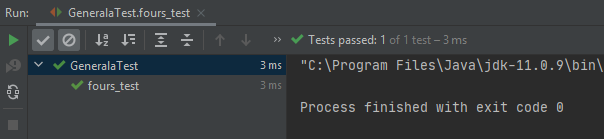
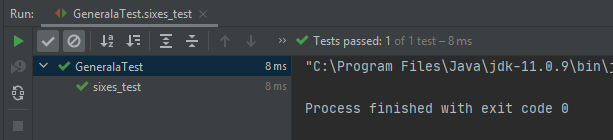
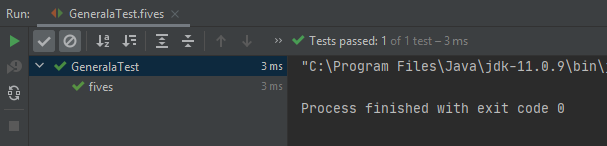
# Prueba 1

1. El primer code smell encontrado, está en los métodos ones, twos y threes, ya que usa un if para cada dado dependiendo del valor que tenga (cinco en total) y además tienen código duplicado, ya que la manera de calcularlos es la misma. Para arreglarlos cree un método sumar, el cual calcula la suma de los valores dependiendo de la categoría.

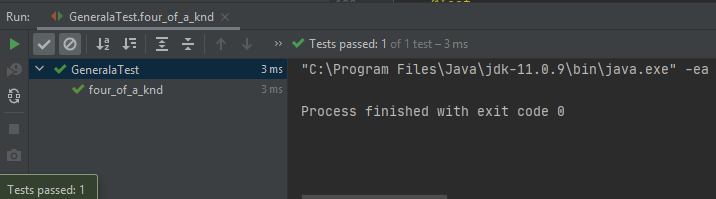
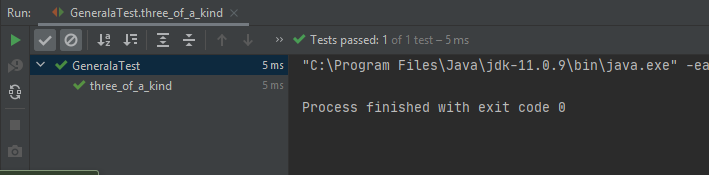


1. Este code smell estaba presente en los métodos fours, fives y sixes. Dicho code smell consistía en que prácticamente los métodos estaban duplicados, y además se repetia la variable sum o s en repetidas ocasiones de forma innecesaria. Para arreglarlo usé el método sumar (creado anteriormente para las clases ones, twos y threes) , reduciendo en gran medida la cantidad de código.

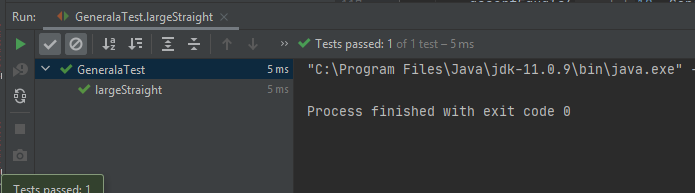
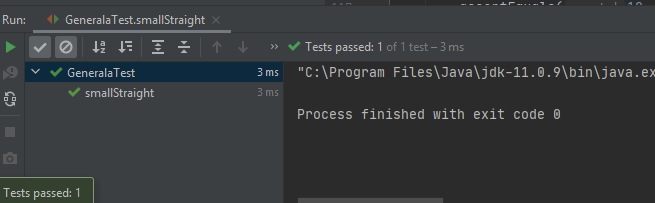




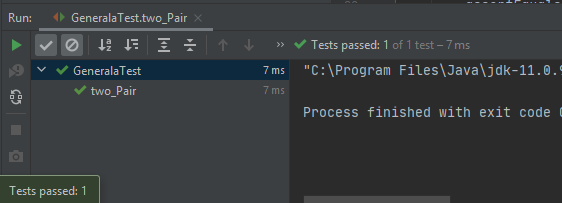
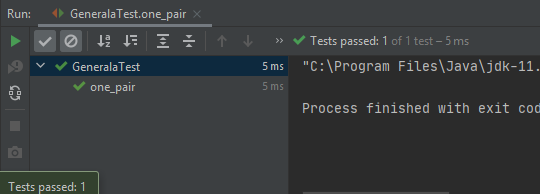
1. Este code smell estaba presente en three of a kind y en four of a kind. Dicho code smell era que ambos métodos realizaban acciones similares y que, además, ambos realizaban dos acciones cada uno: primero asignarle valores a un arreglo y luego ver si el arreglo tenia números pares o no. Por lo que use un for en ambos métodos para asignarle valores al arreglo y luego creé un nuevo método llamado numerosIguales, el cual solo se encarga de revisar si existe el número repetido de dados (3 o 4) que se está buscando.



1. El siguiente code smell estaba presente en el método smallStraight, el cual primero asignaba valores a un arreglo pero en muchas líneas de código, y luego se comparaban las celdas de dicho arreglo de forma individual (una por una). Para arreglarlo realice un for que recorriera el arreglo y asignara sus valores. Luego creé un método llamado encontrarValores, encargado de comprobar si los valores del arreglo daban un smallStraight. El método creado también se usó para el método largeStraight, ya que funcionan de forma similar.

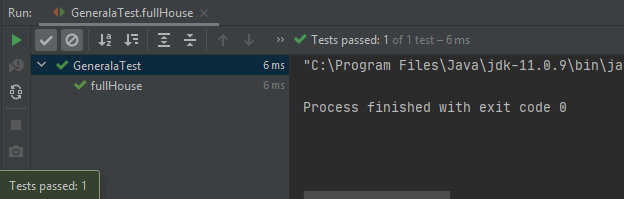


1. Este code smell estaba presente en score\_pair y two\_pair, el cual primero asignaba valores a un arreglo, de forma individual, usando muchas líneas de código, y luego se realizaba otra acción, que consistía en comprobar si existía un par o dos pares, respectivamente. Para refactorizarlos primero use un for que recorre el arreglo y le asigna los valores, y para comprobar los pares cree dos métodos, uno para score\_pair y otro para two\_pair.



1. Este code smell estaba presente en el método fullHouse. Dicho code smell consistía en que primero definia variables con nombres que no representaban del todo en que consistían, y además definia la variable “i” de los for. Luego de esto, se asignaba valores a cada celda de un arreglo, de forma individual, y por ultimo se ejecutaban dos for, uno para verificar si existía un par de dados con valores iguales y además guardar el numero del dado, y otro for para hacer lo mismo pero con un trio de dados.

Para refactorizarlo hice un for para asignar los valores al arreglo, y cambie los nombres de las variables por nombres mas representativos. Luego creé un método para comprobar la existencia de un trio o un par de dados, y creé otro método para obtener el valor de los dados.



1. Este code smell estaba presente en el método chance, el cual llamaba muchas veces a la variable total solo para irle sumando los valores de los dados. Para arreglarlo meti el total dentro de un for y le fui sumando ahí los valores.

