Property-based Testing

Design Property Tests for each of the problems:

1. `passingScore` method which takes in an integer and returns a boolean value (`true` if the score is greater or equal to 60).

Valid T: generate a positive integer between 60 and 100 (inclusive)
Valid F: generate a positive integer between 0 and 59 (inclusive)
Invalid: generate a negative integer or an integer greater than 100

2. `unique` method that returns a descending array of integer values included in the input array.

generate more integers than the range to ensure duplicates and check that the result has no duplicates and it's in reverse order.

3. `indexOf` method that returns the index of the value within the array, or –1 if not found or null.

```
public static int indexOf(final int[] array, final int valueToFind, int startIndex)
```

Randomly generate:
- a list of numbers in a range,
- a target value outside the range,
- an index to add it to,
- an index to start.

4. `NaturalUtils add` method.

Valid: generate two positive numbers
Invalid: generate two positive numbers and negative the first or the second

5. `NaturalUtils subtract` method.

Valid: generate two positive numbers, filter for where the first is greater than the other
Invalid: generate two positive numbers, filter for where the second is greater than the other or negate the first or the second

6. `isTriangle` method:

```java
public class Triangle {
  public static boolean isTriangle(int a, int b, int c) {
    boolean hasABadSide = a >= (b + c) || c >= (b + a) || b >= (a + c);
    return !hasABadSide;
  }
}
```

Valid: generate three positive numbers, filter for where the first is less than the sum of the other two
Invalid: generate three positive numbers, filter for where the first is greater or equal than the sum of the other two (or negate either)

7. `Basket add and remove` methods from last week's homework.

Create actions separately for add and remove and then combine them in randomly generated sequences (lists).