SWE 619 Assignment 5
Fall 2019

Goal: Iteration Abstraction, and a bit of Type Abstraction.

The point of this exercise is to show some of the power of interfaces in Java. In particular, the `Iterable` interface allows for the writing of extremely safe and compact for loops.

Build a version of the `String` class called `MyString` that implements the Java `Iterable` interface. Like `String`, `MyString` should be an immutable class. The `iterator()` method that you supply in `MyString` should produce the characters in the string (as `String` objects), one at a time, from left to right. For example, iterating over the integer "cat" should produce a "c", and then an "a", and then a "t". The `MyString` class doesn't actually have to have much additional functionality; you may find a simple `public String get()` method sufficent.

Provide a quality set of JUnit tests to test your implementation. One of your tests should iterate over a `MyString` and produce the palindrome (ie reverse) of that string. For example:

```
MyString m = new MyString("bat");
String res = "";
for (String i : m) {
   res = i + res;
}
// res has the value "tab";
```

Note: If you wish, you can implement the iterator "from scratch". Or you can use an existing iterator. The latter option is much shorter, and hence is probably better. But you have to be careful: The fact that `MyString` is immutable tells you how you have to implement the `remove()` method.

Turn in a **story**. This means that it is possible to grade your assignment simply by looking at a document. Hence, the document needs to include code fragments, individual tests, screenshots, etc. -- as needed -- to show that you have satisfied each item in the assignment.