AP Computer Science A@Beijing National Day School

Lab 16: `StringChecker`

Due date: Friday, April 26, 2019
Instructor: Mr. Alwin Tareen
Total Points: 15

Task Overview

- Implement a program that determines whether a string conforms to some defined criteria.

Background

- The `StringChecker` interface describes classes that check if strings are valid, according to some criterion.

```
1  public interface StringChecker
2  {
3      boolean isValid(String str);
4  }
```

- A `CodeWordChecker` is a `StringChecker`. A `CodeWordChecker` object can be constructed with three parameters:

    - The minimim code word length(an `integer`)
    - The maximum code word length(an `integer`)
    - A string that must **not** occur in the code word(a `String`)

- A `CodeWordChecker` object can also be constructed with a single parameter that specifies a string that must **not** occur in the code word. In this case, the minimum and maximum lengths with default to `6` and `20`, respectively.

- The following examples illustrate the behavior of `CodeWordChecker` objects.

Example 1

```
StringChecker sc1 = new CodeWordChecker(5, 8, "$");
```

Valid code words have 5 to 8 characters, and must not include the string `"$"`.

Method call	Return value	Explanation
`sc1.isValid("happy")`	`true`	The code word is valid.
`sc1.isValid("happy$")`	`false`	The code word contains `"$"`.
`sc1.isValid("Code")`	`false`	The code word is too short.
`sc1.isValid("happyCode")`	`false`	The code word is too long.

Example 2

```
StringChecker sc2 = new CodeWordChecker("pass");
```

   Valid code words must not include the string `"pass"`. Since the word length bounds are not specified, then the word length bounds are 6 and 20, inclusive.

Method call	Return value	Explanation
`sc2.isValid("MyPass")`	`true`	The code word is valid.
`sc2.isValid("Mypassport")`	`false`	The code word contains `"pass"`.
`sc2.isValid("happy")`	`false`	The code word is too short.
`sc2.isValid("1,000,000,000,000,000")`	`false`	The code word is too long.

The `CodeWordChecker` Class

(a) Write the complete `CodeWordChecker` class. Your implementation must meet all specifications, and conform to all examples.

Specification

The Information Box Which Includes Your Name[5 points]

- Type your English and Pinyin name into the `Author` field, where it says: `YOUR NAME HERE`

Verify that a String Meets the Defined Criteria [10 points]

- Write a `Java` program in the file `HorseBarn.java` that verifies that a String meets the defined criteria.

- You will write your solution in a class called: `public class CodeWordChecker` right below the place where it says: `YOUR CODE HERE`.

- Make sure that you run your `Java` program, and ensure that it is free of errors.

Testing

- The file `CodeWordCheckerJUnitTest.java` contains the `JUnit` test cases which verify the correct functionality of the program.

Submission

- Submit your `Java` program by uploading it to the `Web-CAT` automated grading platform: `http://ec2-54-65-207-33.ap-northeast-1.compute.amazonaws.com:8080/Web-CAT/WebObjects/Web-CAT.woa`