

CITS5501 Software Testing and Quality Assurance

Semester 1, 2022

Week 7 exercise

Please refer to the CITS5501 website and the LMS for details of the due date and submission procedure.

Test plan for the `isHex()` method

Usually when writing integers, we represent them in base 10, but it can be convenient sometimes to write them in other bases, such as base 8 ([octal](#)) or base 16 ([hexadecimal](#)).

In software engineering, when an integer is represented in some base other than 10, the representation will usually have a prefix added to it to show what base it is in – for instance, “0x” for hexadecimal, or “0o” for octal.

A method `isHex()` that your team is testing has the following signature and Javadoc documentation:

```
1  /** Determine whether a String <code>s</code> represents an integer
2   * in hexadecimal notation.
3   *
4   * To represent an integer in hexadecimal notation, a string must
5   * satisfy all the following conditions:
6   *
7   * - It must start with the letters "0x"
8   * - All the characters after the initial two must be either
9   * digits (i.e. in the range '0'-'9') or lowercase letters
10  * from 'a' to 'f'
11  * - The string must have no leading zeroes - that is, the first
12  * character after the "0x" must be either a digit from '1' to
13  * '9', or a letter from 'a' to 'f'.
14  *
15  * @param s A string to be tested
16  *
17  * @return Returns true if <code>s</code> represents an integer in
18  * hexadecimal notation, and false if it does not.
19  */
20 public static boolean isHex(String s);
```

A colleague of yours is devising a test plan for the `isHex` method by applying the Input Space Partitioning technique. They have identified that in this scenario, `isHex` is the only

relevant function, and that it has no parameters other than the `String s` (and you may assume these decisions of theirs are correct).

They have proposed several characteristics to be used in partitioning the input space:

Test plan excerpt – ISP characteristics for `isHex(String s)`

- a. Is the `String s` equal to `null`?
 - Divides the input space into 2 partitions: situations where `s` is `null`, and those where it is not.

The following characteristics sub-partition the second partition of characteristic (a) (i.e., the situation where `s` is not `null`).

- b. Does the `String s` start with an `0`, with an `0x`, or with some other sequence of characters?
 - Produces 3 partitions: Strings starting with `0`, Strings starting with `0x`, and other Strings.
- c. Does `s` contain any characters besides the digits `'0'` through `'9'` and the letters `'a'` to `'f'`?
 - Produces 2 partitions: Strings which contain only these characters, and those which do not.
- d. Before the first non-zero hex digit (i.e. character in the range `'1'–'9'` or `'a'–'f'`), does the digit `'0'` appear zero times, or once, or twice or more?
 - Produces 3 partitions: Strings where `0` appears zero times, Strings where it appears once, and String where it appears twice or more.

Answer the following questions about your colleague’s proposed characteristics:

1. Has your colleague made a good choice of characteristics? Justify your answer. If you would modify or drop any of your colleague’s characteristics, state which of these you would do, and why. (Max. 500 words; 10 marks)
2. Are there additional characteristics you recommend be used? (Assume that any changes you have recommended for question (1) have now been made.) If there are, you may suggest up to two, explaining your reasoning. If there are not, explain why not. (Max. 500 words; 5 marks)