Exercise 1

Finger Exercises due Aug 5, 2020 20:30 -03 *Completo*

Exercise 1

1/1 point (graded)

ESTIMATED TIME TO COMPLETE: 4 minutes

Consider the following code specification:

```
def size(aSet):
    """
    aSet is a collection of objects, which might be empty.
    Objects are assumed to be of the same type.
    """
```

Here is a set of possible test cases to include in a black box test suite. Indicate which of the following conditions would make a good black box test suite for the function `size` by clicking on the appropriate choice(s).

Review: Black Box Test Suites

Black-box testing is a method of software testing that tests the *functionality* of an application. Recall from the lecture that a way to think about black-box testing is to look at both:

- The possible paths through the specification.
- The possible boundary cases.

Undoubtably many - if not all - of the listed tests look like they would be pretty good for testing the function `size`. However, we want you to think critically about the way `size` is specified - including possible boundary cases - and pick a set of tests that

adequately and fully tests all paths and boundary conditions. Be sure the set of tests you pick does not have extraneous, useless, or repetitive tests.

- [x] Empty set
- [x] Set of size 1
- [ ] Set of odd size
- [ ] Set of even size
- [x] Set of size greater than 1
- [ ] Set whose size is a prime number

✔

Explanation:
A good black box test suite would contain tests for the following conditions: Empty set, Set of size 1, and Set of size greater than 1.
Black-box testing is a method of software testing that tests the functionality of an application. Recall from the lecture that a way to think about black-box testing is to look at both the paths through the specification and the possible boundary cases. In this example, the boundary cases all have to do with the size of `aSet`. Specifically these boundary cases are when `aSet` contains zero, one, or many items.
The remaining conditions would not further test the functionality of the `size` function because an odd, even, or prime sized set are all sets of size greater than 1. Nothing in the function specification suggests there is anything special or unique about odd, even, or prime sized sets, so testing those cases specifically simply repeats the test "Set of size greater than 1".

Enviar

Answers are displayed within the problem

Exercise 1

Ocultar discussão

Topic: Lecture 7 / Exercise 1

Show all posts por atividade recente

Take from this question
Black box test is like finding the different cases of available test cases, say None, equal to 1 and mor... 3

? SPOILER
Why is it important to check size of 1? Thanks! 2

? Should we test types?
Should we test if actually all the objects have the same type? 1 new 3