XJTLU | LEARNING MALL ONLINE Need help? English (en)

Home - My courses - CPT204(S2) - Sections - Week 5 : 29 March - 2 April — Specification, Linked List, Generics, Linked-based Deque - Lecture Quiz 5

Started on	Wednesday, 31 March 2021, 20:38
State	Finished
Completed on	Sunday, 4 April 2021, 15:25
Time taken	3 days 18 hours
Grade	**53.33** out of 110.00 (**48**%)

Question 1

Incorrect Mark 0.00 out of 10.00

Consider the two methods to find the value `val` in an integer array `a` below.

```
static int findFirst(int[] a, int val) {
    for (int i = 0; i < a.length; i++) {
        if (a[i] == val) return i;
    }
    return a.length;
}
```

```
static int findLast(int[] a, int val) {
    for (int i = a.length -1 ; i >= 0; i--) {
        if (a[i] == val) return i;
    }
    return -1;
}
```

If clients only care about calling the find method when they know that `val` *always occurs exactly once* in `a`, `findFirst` and `findLast` are behaviorally equivalent.

Select one:

○ True

● False ✗

The correct answer is 'True'.

Question 2

Incorrect Mark 0.00 out of 10.00

Consider the two methods to find the value `val` in an integer array `a` below.

```
static int findFirst(int[] a, int val) {
    for (int i = 0; i < a.length; i++) {
        if (a[i] == val) return i;
    }
    return a.length;
}
```

```
static int findLast(int[] a, int val) {
    for (int i = a.length -1 ; i >= 0; i--) {
        if (a[i] == val) return i;
    }
    return -1;
}
```

If clients only care that the find method should return any index `i` such that `a[i] == val`, if `val` is in `a`;
and any integer `j` where `j` is ***not*** a valid index of array `a`, otherwise;
then `findFirst` and `findLast` are behaviorally equivalent.

Select one:

○ True

● False ✗

The correct answer is 'True'.

Question 3

Correct

Mark 10.00 out of 10.00

Suppose we're working on the method below:

```
/**
 * Requires: tiles has length 7 & contains only uppercase letters.
 *           crossings contains only uppercase letters, without duplicates.
 * Effects: Returns a list of words where each word can be made by taking
 *          letters from tiles and at most 1 letter from crossings.
 */
public static List<String> scrabble(String tiles, String crossings) {
    if (tiles.length() != 7) { throw new RuntimeException(); }
    return new ArrayList<>();
}
```

Which one is a part of the *postcondition* of `scrabble`?

Select one:

● a. scrabble returns a list of strings

○ b. tiles has only uppercase letters

○ c. crossings has no duplicates

○ d. scrabble takes two arguments

Your answer is correct.

The correct answer is: scrabble returns a list of strings

Question 4

Correct

Mark 10.00 out of 10.00

Suppose we're working on the method below:

```
/**
 * Requires: tiles has length 7 & contains only uppercase letters.
 *           crossings contains only uppercase letters, without duplicates.
 * Effects: Returns a list of words where each word can be made by taking
 *          letters from tiles and at most 1 letter from crossings.
 */
public static List<String> scrabble(String tiles, String crossings) {
    if (tiles.length() != 7) { throw new RuntimeException(); }
    return new ArrayList<>();
}
```

Which one is **not** a part of the *precondition* of `scrabble`?

Select one:

● a. scrabble returns an empty ArrayList

○ b. tiles has length 7

○ c. crossings is a string of uppercase letters

○ d. scrabble's arguments are of type String and String

Your answer is correct.

The correct answer is: scrabble returns an empty ArrayList

Question 5

Incorrect

Mark 0.00 out of 10.00

Suppose we're working on the method below:

```
/**
 * Requires: tiles has length 7 & contains only uppercase letters.
 *           crossings contains only uppercase letters, without duplicates.
 * Effects: Returns a list of words where each word can be made by taking
 *          letters from tiles and at most 1 letter from crossings.
 */
public static List<String> scrabble(String tiles, String crossings) {
    if (tiles.length() != 7) { throw new RuntimeException(); }
    return new ArrayList<>();
}
```

Which one is the part of the spec that are ***checked statically*** by Java?

Select one:

○ a. scrabble takes two arguments

○ b. tiles is a string of uppercase letters

● c. crossings has no duplicates

○ d. when tiles.length() != 7, scrabble throws a RuntimeException

Your answer is incorrect.

The correct answer is: scrabble takes two arguments

Question 6

Incorrect

Mark 0.00 out of 10.00

Which of the following is **not** part of a method's specification?

Select one:

○ a. restrictions on used data types

○ b. return type

○ c. restrictions on return values

○ c. restrictions on return values

● d. number of arguments

○ e. argument types

○ f. restrictions on argument values

Your answer is incorrect.

The correct answer is: restrictions on used data types

Question 7

Incorrect

Mark 0.00 out of 10.00

Alice writes the following code:

```
public static int gcd(int a, int b) {
    if (a > b) {
        return gcd(a-b, b);
    } else if (b > a) {
        return gcd(a, b-a);
    }
    return a;
}
```

Bob writes the following test:

```
@Test public void gcdTest() {
    assertEquals(6, gcd(24, 54));
}
```

Which of the following statement is **incorrect** ?

Select one:

○ a. If Alice adds `a > 0` to the precondition, Bob should test negative values of a

○ b. If Alice does not add `a > 0` to the precondition, Bob should test negative values of a

○ c. Alice should write `a > 0, b > 0` in the precondition of `gcd`

● d. Alice should not write `a and b are integers` in the precondition of `gcd`

Your answer is incorrect.

The correct answer is: If Alice adds `a > 0` to the precondition, Bob should test negative values of `a`

Question 8

Partially correct

Mark 13.33 out of 20.00

Given the following specification :

```
static int find(int[] arr, int val)
    requires: arr[0] == val
    effects:  returns index i such that arr[i] == val
```

Which are the valid test cases for `find` ?

Select one or more:

☑ `find([ 1, 2, 3 ], 1)` must return `0`

□ `find([ 4, 4, 5 ], 4)` must return `0`

□ `find([ 4, 4, 5 ], 4)` must return 1

□ `find([ 6, 7, 8 ], 2)` throws an exception

☑ `find([ 3 ], 3)` must return `0`

☑ `find([ 4 ], 5)` must not return `0`

Your answer is partially correct.

You have selected too many options.

The correct answers are: `find([ 1, 2, 3 ], 1)` must return `0`, `find([ 3 ], 3)` must return `0`

Question 9

Correct

Mark 10.00 out of 10.00

What is a condition that must be preserved and guaranteed to be true during a method's execution called ?

Answer: Invariant ✓

The correct answer is: invariant

Question 10

Correct

Mark 10.00 out of 10.00

To allow types such as Integer, String, and user-defined types to be a parameter to methods, classes, and interfaces, we use generics

Using it, we can create classes that work with different data types.

Finish review

◀ Lab 5 Recording

Jump to...

Lab Exercise 5.1 LLDeque EMP