Questions

Bookmark this page

Unhelpful 1

2/2 points (graded)

Louis Reasoner doesn't want to use a helper method, so he tries to implement `subsequences()` by storing `partialSubsequence` as a static variable instead of a parameter. Here is his implementation:

```
private static String partialSubsequence = "";
public static String subsequencesLouis(String word) {
    if (word.isEmpty()) {
        // base case
        return partialSubsequence;
    } else {
        // recursive step
        String withoutFirstLetter = subsequencesLouis(word.substring(1));
        partialSubsequence += word.charAt(0);
        String withFirstLetter = subsequencesLouis(word.substring(1));
        return withoutFirstLetter + "," + withFirstLetter;
    }
}
```

What does subsequencesLouis("c") return?

- "c"
- ""
- [✓ ",c"]
- "c,"

✓

What does subsequencesLouis("a") return?

- "a"
- ""
- ",a"
- "a,"

● "c,ca"

✓

Explanation

The static variable maintains its value across calls to subsequencesLouis(), so it still has the final value "c" from the call to subsequencesLouis("c") when subsequencesLouis("a") starts. As a result, every subsequence of that second call will have an extra c before it.

Submit

Show Answer

Answers are displayed within the problem

Unhelpful 2

1/1 point (graded)

Louis fixes that problem by making partialSubsequence public:

```
/**
 * Requires: caller must set partialSubsequence to "" before calling subsequencesLouis().
 */
public static String partialSubsequence;
```

Alyssa P. Hacker throws up in her mouth when she sees what Louis did. Which of these statements are true about his code?

☐ partialSubsequence is risky -- it should be final

☑ partialSubsequence is risky -- it is a global variable

☐ partialSubsequence is risky -- it points to a mutable object

✓

Explanation

partialSubsequence is indeed a global variable. It can't be made final, however, because the recursion needs to reassign it frequently. But at least it doesn't point to a mutable object.

Submit

Show Answer

Answers are displayed within the problem

Unhelpful 3

7/7 points (graded)

Louis gives in to Alyssa's strenuous arguments, hides his static variable again, and takes care of initializing it properly before starting the recursion:

```
public static String subsequences(String word) {
    partialSubsequence = "";
    return subsequencesLouis(word);
}

private static String partialSubsequence = "";

public static String subsequencesLouis(String word) {
    if (word.isEmpty()) {
        // base case
        return partialSubsequence;
    } else {
        // recursive step
        String withoutFirstLetter = subsequencesLouis(word.substring(1));
        partialSubsequence += word.charAt(0);
        String withFirstLetter = subsequencesLouis(word.substring(1));
        return withoutFirstLetter + "," + withFirstLetter;
    }
}
```

Unfortunately a static variable is simply a bad idea in recursion. Louis's solution is still broken. To illustrate, let's trace through the call subsequences("xy"). It will produce these recursive calls to subsequencesLouis():

```
subsequencesLouis("xy")
    subsequencesLouis("y")
        subsequencesLouis("")
        subsequencesLouis("")
    subsequencesLouis("y")
        subsequencesLouis("")
        subsequencesLouis("")
```

When each of these calls starts, what is the value of the static variable partialSubsequence?

1. subsequencesLouis("xy")

empty string ✓ **Answer:** empty string

2. subsequencesLouis("y")

empty string ✓ **Answer:** empty string

3. subsequencesLouis("")

empty string ✓ **Answer:** empty string

4. subsequencesLouis("")

y ✓ **Answer:** y

5. subsequencesLouis("y")

yx ✓ **Answer:** yx

6. subsequencesLouis("")

yx ✓ **Answer:** yx

7. subsequencesLouis("")

yxy ✓ **Answer:** yxy

Explanation
Everything seems fine until call 5, where it becomes clear that the static variable is still clinging to letters like "y" that were added to it in deeper levels of recursion and never discarded.
The final (wrong) return value of this implementation can be read off from the base cases, calls 3,4,6,7: ",y,yx,yxy".
Static variables and aliases to mutable data are very unsafe for recursion, and lead to insidious bugs like this. When you're implementing recursion, the safest course is to pass in all variables, and stick to immutable objects or avoid mutation.

Submit

Show Answer

Answers are displayed within the problem

Previous Next