Worksheet: Fragile Base Class Problem

Inheritance of code has its pitfalls. It is often forgotten that inheritance is also an interface between classes, which is very tight (the whole code is inherited), but is often miserably underspecified. This worksheet is intended to raise awareness of this problem.

Let's take a closer look at the following class. The example is taken from the Java API (package `java.io`) and has been simplified to make the problem clearer:

```
public class StringOutputStream extends FilterOutputStream {
    public StringOutputStream(OutputStream s) { … }
    void write (char ch) { … }
    void write (String s) { … }
}
```

Suppose you want to count the number of characters written by a `StringOutputStream`, regardless of which `write` method was used.

```
public class CountedOutputStream extends StringOutputStream {
    public CountedOutputStream(OutputStream s) {
        super(s);
    }
    private int c = 0;
    public int writtenChars() {
        return c;
    }
    …
}
```

Task:
Provide an implementation for the methods `write(char ch)` and `write(String s)` in class `CountedOutputStream`. Easy peasy? Well then:

Constraints:
You know nothing about the implementation of class `StringOutputStream`. Your solution should work properly independent of which of the following three implementations for `StringOutputStream.write(String)` was actually used.

```
a) void write(String s) {
       // writes the string directly into the stream
   }

b) void write(String s) {
     for(int i=0; i<s.length(); i++)
       write(s.charAt(i));
   }

c) void write(String s) {
     write(s.charAt(0));
     if(s.length() > 1)
       write(s.substring(1)); // remainder starting with the second character
   }
```

In project `14_Inheritance` on the AD you find a frame of class `CountedOutputStream` in package `patterns.inheritance`, which contains some `TODO`s. In the `test` source folder you will also find the unit test `AllOutputStreamTests`, which tests the given class `StringOutputStream` as well as your implementation of class `CountedOutputStream`.

Do not try to understand the tests! Focus on the implementation of class `CountedOutputStream`. The test classes are parameterized with different implementations of the base class (as the ones shown above).