EECS2030Z Test 2

Version B

GETTING STARTED

1. Start eclipse; use the workspace suggested by eclipse (or remember the directory name of your workspace).
2. Import the test project by doing the following:
    1. Under the **File** menu choose **Import...**
    2. Under **General** choose **Existing Projects into Workspace** and press **Next**
    3. Click the **Select archive file** radio button, and click the **Browse...** button.
    4. Select the file **test2B.zip** and click **OK**
        - If you do not see the file named **test2B.zip** in your directory, then open a terminal and copy and paste the following command:

          ```
          cp /eecs/dept/www/course/2030/labtest/test2B.zip .
          ```

          and re-import the project.
    5. Click **Finish**.
3. All of the files you need for this test should now appear in eclipse.
4. Open a terminal. You will use this terminal to submit your work.
5. Copy and paste the command `cd workspace/Test2B/src/test2` into the terminal and press enter.

Question 1 (20 marks total)

SOLUTION

Implement the utility class described by this API. You do not have to include javadoc comments.

```
submit 2030 test2B Utility2B.java
```

Question 2 (10 marks total)

SOLUTION

Implement the constructors for the class described by this API. Use constructor chaining where possible when implementing your constructors. You do not have to include javadoc comments in your code, but see Question 3C below.

```
submit 2030 test2B Fraction.java
```

Question 3 (20 marks total)

A.

State the definition of a method postcondition.

A method postcondition is a condition that the method promises will be true immediately after the method finishes running.

Grading scheme: 3 marks total.

B.

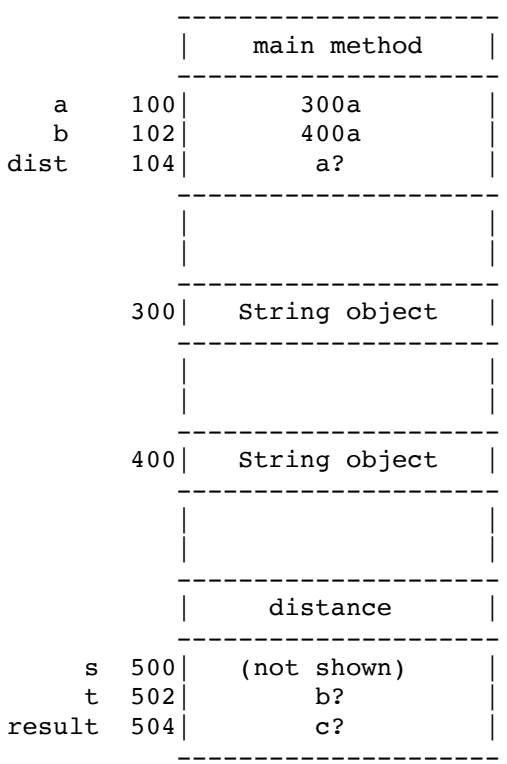
Suppose that the method name `distance` from Question 1 was implemented like so:

```
public static int distance(String s, String t) {
  int result;
  /* code not shown here that assigns the correct value to result */
  return result;
}
```

Suppose a client writes a `main` method that includes the following two lines of Java code:

```
String a = "zzzz";
String b = "walk";
int dist = Utility2B.distance(a, b);
```

The memory diagram illustrating the state of memory for the three lines of client code is shown below. What suitable values of a, b, and c would complete the memory diagram?



a = 4, b = 400a, c = 4

a and c are both equal to the value of the distance between the two strings. b is equal to the address of the second string.

Grading scheme: 3 marks total (1 for each of a, b, and c).

C.

Provide the Javadoc necessary to *exactly* reproduce the API documentation for the constructor `Fraction(int numer, int denom)` from Question 2.

```
/**
 * Initializes the fraction so that its numerator is
 * equal to the given numerator and its denominator is
 * equal to the given denominator.
 *
 * @param numer the numerator of the fraction
 * @param denom the denominator of the fraction
 * @throws ArithmeticException if the denominator is equal to zero.
 */
```

Grading scheme: 4 marks total (1 for method description, 1 for the the first parameter description, 1 for the second parameter description, 1 for the exception). The answer must appear exactly as shown above (missing or extra white space is acceptable, and the leading * are not required.).

D.

Provide 3 test cases for the method `Utility2B.distance`. Make sure that each test case tests a different feature of the method; try to include one boundary test case. For each test case, provide a one sentence explanation of what the test case is testing.

```
The boundary case occurs when both strings are empty.

s                     : ""
t                     : ""
expected return value: 0
explanation           : tests strings of zero length

s                     : "and"
t                     : "and"
expected return value: 0
explanation           : tests two strings that are equal

s                     : "ate"
t                     : "ace"
expected return value: 1
explanation           : tests two strings that are not equal

s                     : "a"
t                     : "and"
expected return value: IllegalArgumentException
explanation           : tests that an exception is thrown if strings have different lengths
```

Grading scheme: 6 marks total.

2 marks for each test case (1 mark for providing appropriate inputs, 0.5 mark for providing the expected return value or result, 0.5 mark for the explanation). Deduct 1 mark if there are 3 valid test cases that do include the boundary case.

E.

Consider the following implementation of `equals(Object)` for the class `Fraction` from Question 2:

```
public class Fraction {

  private int numer;
  private int denom;

  // constructors not shown

  @Override
  public boolean equals(Object obj) {
    if (this == obj) {
      return true;
    }
    if (this.getClass() != obj.getClass() || obj == null) {
      return false;
    }
    Fraction other = (Fraction) obj;
    if (this.numer != other.numer) {
      return false;
    }
    if (this.denom != other.denom) {
      return false;
    }
    return true;
  }

}
```

Explain whether or not the implementation shown above satisfies the equals contract.

The implementation does not satisfy the equals contract because `x.equals(null)` throws an exception. The exception is thrown because `obj` is used to invoke `getClass` before it is checked for equality to `null`:

```
    if (this.getClass() != obj.getClass() || obj == null)
```

Grading scheme: 4 marks total.

2 marks for stating `x.equals(x)` throws an exception. 2 marks for the explanation.

submit 2030 test2B answers.txt