Java – Strings and I/O

10.01 ChopEmUp.java

Write a utility class called ChopEmUp and implement the static methods below. This lab focuses on using the Scanner class to process Strings.

static int **countEm**(String str) – returns the count of the tokens contained within *str*.

ChopEmUp.countEm("1 hi 12 bye 101"); // => returns 5

static String **condenseSpaces**(String str) – returns the same String but with extra spaces removed. Given " Have   no fear   of perfection,   you'll   never reach   it. ", return "Have no fear of perfection, you'll never reach it.".

static int **addEm**(String str) – returns the sum. Precondition: str contains only +, - operators, integers and is a valid expression.

ChopEmUp.addEm("5 - 4 + 12 - -2"); // => returns 15

static int **smallest**(String str) – returns the smallest number in str. Precondition: size > 0.

static double **range**(String str) – returns the range(largest – smallest) of doubles in str. Precondition: str contains only numbers and size > 0.

static String **cumulativeTotals**(String str) – returns the cumulative totals as a String. Given "1 5 8 3", return "1 6 14 17".

static String **flipCoins**(String str) – returns the percentage of heads rounded to two decimal places and "win" or "lose". You win if there are more heads than tails.

chopEmUp.flipCoins("H T T H H"); // -> returns "60.00% win"

static int **adjacentDuplicatesCount**(String str) – returns the count of duplicates that are next to each other. Given "1 1 2 7 7 7 7 3 3 5 3", return 3.

static String **stats**(String str) – returns the number of tokens and the longest token. Given "When life gives you lemons, make lemonade.", return "7 tokens, longest is 8". Precondition: str contains only words, commas, and periods.

\\s : a whitespace character

[abc] : a, b, or c (simple class)

+ : greedy quantifier which means one or more times

Invoking useDelimiter("[a,eo]+") on "hello,World" splits into the tokens "h" "ll" "W" and "rld"

static String **halloween**(String str) − the input String will contain alternating Halloween costumes for guys and girls and a count. Return a String with the count of different costumes for guys, girls, and the absolute difference. Given "IronMan 4 Leia 2 SpiderMan 1 BlackWidow 3 Hulk 9 Catwoman 13 Superman 4", return "guys 4, girls 3, diff 0"

CHOP’EM UP SAMPLE DATA

```
// Sample Data

public static void main(String[] args) {
  out.println(countEm("1 hi 12 bye 101") + " == 5");
  String spaces = "  Have     no fear    of perfection,      ";
  spaces += "you'll    never  reach     it.  ";
  String res = "Have no fear of perfection, ";
  res += "you'll never reach it.";
  out.println(condenseSpaces(spaces).equals(res)  + " == true");
  out.println(addEm("5 - 4 + 12") + " == 13");
  String s = "99 3 66 44 12 78 -4 102";
  out.println(smallest(s) + " == -4");
  s = "99.4 3 66.2 4.4 12 78 -4 102.0";
  out.println(range(s) + " == 106.0");
  out.println(cumulativeTotals("1 5 8 3") + " == 1 6 14 17");
  out.println(flipCoins("H T T H H") + " == 60.00% win");
  s = "1 1 2 7 7 7 7 3 3 5 3";
  out.println(adjacentDuplicatesCount(s) + " == 3");
  s = "When life gives you lemons, make lemonade.";
  out.println(stats(s) + " == 7 tokens, longest is 8");
  s = "IronMan 4 Leia 2 SpiderMan 1 BlackWidow 3 ";
  s += "Hulk 9 Catwoman 13 Superman 4";
  out.println(halloween(s) + " == guys 4, girls 3, diff 0");
}
```

CHOP’EM UP SAMPLE OUTPUT

```
5 == 5
true == true
13 == 13
-4 == -4
106.0 == 106.0
1 6 14 17 == 1 6 14 17
60.00% win == 60.00% win
3 == 3
7 tokens, longest is 8 == 7 tokens, longest is 8
guys 4, girls 3, diff 0 == guys 4, girls 3, diff 0
```

10.02 ProcessingFiles.java

Write a utility class called ProcessingFiles and implement the static methods below. This lab focuses on using the Scanner class to read and write files and process Strings. Several text files have been provided for testing.

Each IDE is unique to where the default location resides for reading and writing files. Generally, the IDE's that require a project (Intellij, Netbeans, Eclipse, etc.), the default location is inside the root directory of the project. Use file explorer on windows to paste the files inside the project folder. Typically, IDE's that support small scale software development (DrJava, JCreator, BlueJ, Atom, etc.), the default location is in the folder where the java file resides. Alternatively, you can pass the fully qualified path(i.e., C:\\text_folder\\someText.txt - requires extra backslash) or set up a resource folder. It is recommended to use the default location.

static String **stringifyFile**(String filename) – returns the contents of the file as a String. The formatting should remain the same.

static void **writeNumbers**(String filename, int n) – creates and writes numbers to a file called *filename*, starting at 1 and ending at *n* inclusive. Place each number on its own line.

NUMBERS.TXT OUTPUT FILE

```
1
2
3
4
5
```

static String **piggyBank**(String filename) – read a file which contains pairs of tokens where the first token is the denomination and the second is the amount. Return the total. Given "Quarters 4 pennies 3 DIMES 8 quarters 1 nickels 2", return "Piggy Bank Total: $2.18".

PIGGYBANK.TXT INPUT FILE

```
Quarters 4
        pennies 3 DIMES 8
quarters 1          nickels 2
```

static void **alternate**(String inputFileName, String outputFileName) – writes the contents of the input file to the output file but with each line swapped.

FROST.TXT INPUT FILE

```
Some say the world will end in fire,
Some say in ice.
From what I’ve tasted of desire
I hold with those who favour fire.
But if it had to perish twice,
I think I know enough of hate
To say that for destruction ice
Is also great
And would suffice.
```

FROSTSWAPPED.TXT OUTPUT FILE

```
Some say in ice.
Some say the world will end in fire,
I hold with those who favour fire.
From what I’ve tasted of desire
I think I know enough of hate
But if it had to perish twice,
Is also great
To say that for destruction ice
And would suffice.
```

static void **removeTags**(String inputFileName, String outputFileName) – read a file containing valid html tags and write the contents to the specified file but with the tags removed.

A tag is any text between a set of angle brackets. Lots of ways to solve this one. One approach is to read a line and remove tags by repeatedly finding an open angle bracket and the corresponding closed angle bracket. The following two methods may be useful.

indexOf(ch, fromIndex)

substring(beginIndex, endIndex.

Notice the whitespace has been preserved on the output.

INDEX.HTML INPUT FILE

```

    <head>
       <title>Title of the page</title>
    </head>
    <body>
       <h1>Heading</h1>
       <form action="some_action" method="POST">
             <label for="lname">Last name:</label><br>
             <input type="text" id="lname" name="lname">
       </form>
       <p>This is a paragraph</p>
    </body>
    <footer>
       <p>This is the footer</p>
    </footer>

```

NOTAGS.TXT OUTPUT FILE

```

  Title of the page


      Heading

                Last name:


      This is a paragraph


      This is the footer

```

static void **wrap**(String inputFileName, String outputFileName, int MAX_LINE) – given a file with no new lines, return a wrapped representation where the words have been wrapped around at a line length of MAX_LINE characters.

Do not split words and wrapped lines should not have any leading whitespace.

```
wrap("OfDeath.txt", "OfDeathWrap20.txt", 20);
```

OFDEATH.TXT INPUT FILE

```
MEN fear death, as children fear to go in the dark; and …
```

OFDEATHWRAP20.TXT OUTPUT FILE

```
MEN fear death, as
children fear to go
in the dark; and as
that natural fear in
children, is
increased with
tales, so is the
other. Certainly,the
contemplation of
death, as the wages
of sin, and passage
to another world, is
holy and religious;
but the fear of it,
as a tribute dueunto
nature, is weak. Yet
in religious
meditations, there
is sometimes mixture
of vanity, and of
superstition. You
shall read, in some
of thefriars' books
of mortification,
that a man should
think with himself,
what the pain is, if
he have but his
finger's end
pressed, or
tortured,and thereby
…
--Extinctusamabitur
idem.
```

SAMPLE DATA

```
    // sample test cases

    public static void main(String[] args) throws IOException {
        System.out.println(stringifyFile("RollingStone.txt") + "\n");
        writeNumbers("numbers.txt", 5);
        System.out.println(piggyBank("piggyBank.txt") + "\n");
        alternate("Frost.txt", "FrostSwapped.txt");
        removeTags("index.html", "noTags.txt");
        wrap("ExcellentLord.txt", "ExcellentLordWrapped50.txt", 50);
        wrap("OfDeath.txt", "OfDeathWrap20.txt", 20);
}
```