Seth Gilbert

CS2020: Algorithms and Data Structures (Accelerated)

Exercise 1: Numbers and Strings

Problem Set Description

Exercise 1: Numbers and Strings

Attempt

This exercise contains two parts: **PizzaCircles** (which focuses on manipulating numbers) and **StringManipulation** (which is self-explanatory).

For each part, you need to complete the specified method, and run the associated JUnit test. Once you pass all the tests (and only after you pass all the tests), submit the Java file.

(a) PizzaCircles

You are having a party, and you need to decide how big a pizza to get. You know that each person requires a certain amount of pizza (e.g., 500 square centimeters). Your task is to write a method MinSizedPizza which takes an input an area and returns the minimum integral radius of a pizza that is sufficiently large to serve everyone. (Pizzas are only sold in integer, non-negative sizes, e.g., 0, 1cm, 2cm, 3cm, 4cm, ...) If the user asks for an invalid value (e.g., a negative area), then return 0.

For example, if the input is 1000 square centimeters (e.g., enough pizza to serve two), then you need a pizza with a radius of at least 18cm. (A 17cm pizza is not big enough, since it only would have an area of 907.9.)

See the attached file **PizzaCircles.java**, and the JUnit tests are available in **PizzaCircleTest.java**. Submit PizzaCircles.java.

Hints:

Notice that the input is a double and the output is a long. Be careful since you are manipulating large values to use large enough data types. If you store your radius in an integer, for example, you may overflow.

You are going to have to round your real number to an integer. There are two ways to do this. The java.lang.Math library has a method **round** that returns the closest integer. Also, you can *cast* a floating point number to an integer and you get the floor. For example: **long f = (long)d** takes the floor of d and stores it in f. Look up "type casts" in a Java reference book.
Read the Oracle page on java.lang.Math at http://docs.oracle.com/javase/6/docs/api/java/lang/Math.html.

(b) StringManipulation

In this problem, you are given two strings, and an instruction for how to combine the two strings. There are three possible instructions:

join: concatenate the two strings
outer: concatenate the first part of the first string and the second part of the second string
inner: concatenate the second part of the first string the first part of the second string

The instruction itself is hidden inside a longer string. Thus, the method you have to implement looks like:

public static String InstructionSearch(String instructions, String first, String second){

The instructions string contains the command indicating what to do to the first and second strings. For example:

InstructionSearch("abcdJOINefgh", "Seth", "Gilbert")

should return the string "SethGilbert". Here are the precise rules for how it should behave:

The instructions input contains one of three commands (join, inner, outer).

The command can be in upper or lower case, or mixed case. If the instructions input contains no command, or more than one command, the method should return the string "ERROR". This error requirement should take priority over all others.
If the command is "join", then the output should be the concatenation of the two strings. For example, if the two strings are "xyz" and "123" then the joined output is "xyz123".
If the command is inner or outer, then the instruction string will also contains a single digit in the range of [0,1,3, ..., 9]. If the instruction string contains no digits or more than one digit, then it should return the string "NO_NUMBER".
Let d be the digit in the instructions.

If the command is outer, then the method should return the first d characters of the first string and the last d characters of the second string. For example, if d=2, and the strings are xyz and 123, then the method should return xy23.
If the command is inner, the the method should return the last d characters of the first string and the first d characters of the second string. For example, if d=2, and the strings are xyz and 123, then the method should return yz12.
If either string is not long enough (i.e., contains less than d characters), then the entire string should be included. For example, if d=4 and the strings are xyz and 123, then the method should return xyz123 (for either command inner or outer).

See the attached file **StringManipulation.java**, and the JUnit tests are available in **StringManipulationTest.java**. Submit StringManipulation.java.

Hints:

The String class contains a large number of useful methods for manipulating strings. Read the specifications for the String class at: http://docs.oracle.com/javase/7/docs/api/java/lang/String.html. Be careful to note that the method should be case-insensitive, i.e., the capitalization of the characters does not matter.

Exp:	1
Max grade:	1
Open at:	20-01-2014 18:00
Close at:	04-02-2014 23:59

Problem Set files :

- PizzaCircles.java
- PizzaCirclesTest.java
- StringManipulation.java
- StringManipulationTest.java