Lab 8

Methods Lab

In this lab, you will be writing helper methods that perform a particular repetitive task common across multiple classes. These methods will avoid repeating the same piece of code in different classes.

This lab is written somewhat differently. Be sure to read the instructions carefully!

Like your past labs, **you need to have a menu in your main method that will allow testing of all of these methods**. Note that because the methods you will be writing will take parameters, you will need to make some decisions to call the methods correctly. You can do this in any way that makes sense to you.

Please keep in mind how you are naming your methods. Good names can be VERY helpful when using them later (and you may want to use them for later labs and maybe even your final project!).

**Method 1** is intended to get a string from the user. You will need to create code that takes a scanner (which reads from [System.in](http://system.in/)) and a string as parameters. The string will be output to the console as a prompt for the user to enter a string value. The method should then return the string entered by the user.

**Method 2** is intended to get a valid int from the user. You should write a method that takes a scanner (that is reading from [System.in](http://system.in/)) and a string as parameters. The string should be output to the console as a prompt to enter an int value. The method will verify that the value entered is an int and return it. If it is not a valid int, it will re-prompt the user for a valid value until it is valid.

**Method 3** is intended to get a valid int value from the user that is between the values of X and Y. This method will use Method 2 above to ensure it gets a valid int. You should write a method that takes a scanner (that is reading from [System.in](http://system.in/)), a string, and two ints as parameters. This method will call Method 2 to get a valid int by passing the scanner and string. It will then determine if the int is between the values of the two int parameters that it was passed. If the value is between the two parameter values, return the int, otherwise call Method 2 again to get a new int that will be between the correct values.

**Method 4** is similar to Method 2, but for doubles.

**Method 5** is similar to Method 3, but for doubles.

**Method 6** is intended to test the other methods by using them to make kind of a mad-lib style input/output. You should write a method that does the following:

1. Call Method 1 and asks the user to enter a string for the name of an item. (I will reference whatever the user typed in as <item> for the rest of this method).
2. Call Method 1 and asks the user to enter a color name.
3. Call Method 1 and ask the user to enter a different color name.
4. Call Method 2 and ask the user to enter the max number of <item> someone can have.
5. Call Method 2 and ask the user to enter the minimum number of <item> someone can have.
6. Call Method 4 and ask the user to enter the max price of <item>.
7. Call Method 4 and ask the user to enter the min price of <item>.
8. Create a string that says that the system has (random number between min and max of <item>) of (random color from one of the colors they entered) <item>'s and ask them how many would you like to buy.
9. Pass that string to Method 3 and make sure you get a number back that is between the min number of possible items and the random number you said that the system had.
10. Call Method 5 and ask the user how much they are willing to pay for each one, specifying a range of the min and max prices.
11. Calculate and print the final price owed, what they are purchasing, and at what price each of them is being purchased for.

Example of method 6 running:

Enter a name for an item: *<user enters "phone" here>*

Enter a color: *<user enters "blue" here>*

Enter a different color: *<user enters "pikachu" here>*

Enter the maximum number of phones someone can have: *<user enters "5" here>*

Enter the minimum number of phones someone can have: *<user enters "1" here>*

Enter the maximum price of a phone: *<user enters "1000.0" here>*

Enter the minimum price of a phone: *<user enters "100.0" here>*

The system has 3 pikachu phones, how many would you like to buy? *<user enters "4" here>*

Sorry, that’s not within range!

The system has 3 pikachu phones, how many would you like to buy? *<user enters "-1" here>*

Sorry, that’s not within range!

The system has 3 pikachu phones, how many would you like to buy? *<user enters "2" here>*

How much are you willing to pay for each phone (between 100.0 and 1000.0)? *<user enters "500.0" here>*

The final price owed is $1000.0 for 2 phones at $500.0 each.

Thanks for your business!