



JAVA HOMEWORK ASSIGNMENT # 3



By: Eric Dockery

Date: 10-07-2013

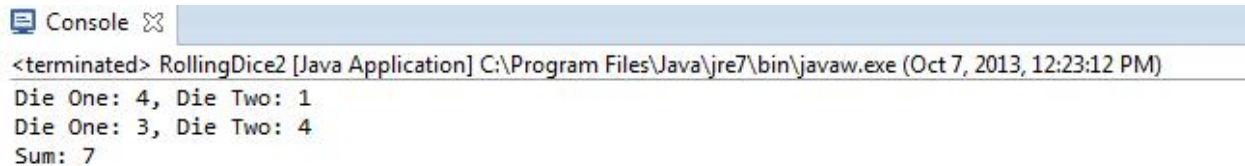
Problem 1 programming project 4.8.

Using the Die class defined in this chapter, design and implement a class called PairOfDice, composed of two Die objects. Include methods to set and get the individual die values, a method to roll the dice, and a method that returns the current sum of the two die values. Create a driver class called RollingDice2 to instantiate and use a PairOfDie object.

How to solve:

This program was in the chapter so for solving there was only a few minor changes to the original. I changed the name of the RollingDice program to RollingDice2 and then recreated the code along with the PairOfDice code given in the chapter.

Screen Captures:



```
Console ✕
<terminated> RollingDice2 [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Oct 7, 2013, 12:23:12 PM)
Die One: 4, Die Two: 1
Die One: 3, Die Two: 4
Sum: 7
```

Problem 2 Programming Project 4.13.

Modify the Fahrenheit program from this chapter so that it displays a button that, when pressed, also causes the conversion calculation to take place. That is, the user will now have the option of either pressing Enter in the text field or pressing the button. Have the listener that is already defined for the text field also listen for the button push.

How to Solve:

The modification of the Fahrenheit program will consist of adding a private JButton push; a push = new JButton (“Calculate!!”); a add (push); and a push.addActionListener(new TempListener()); This will change the program to also listen for the button press as well as the enter key. I would have liked to move the Calculate button below the Celsius line but I was having trouble moving it correctly.

Screen Captures:



Problem 3 Programming Project 5.10.

Modify the Account class from Chapter 4 so that it preforms validity checks on the deposit and withdraw operations. Specifically, don't allow the deposit of a negative number or withdrawal that exceeds the current balance. Print appropriate error messages if these problems occur.

How to Solve:

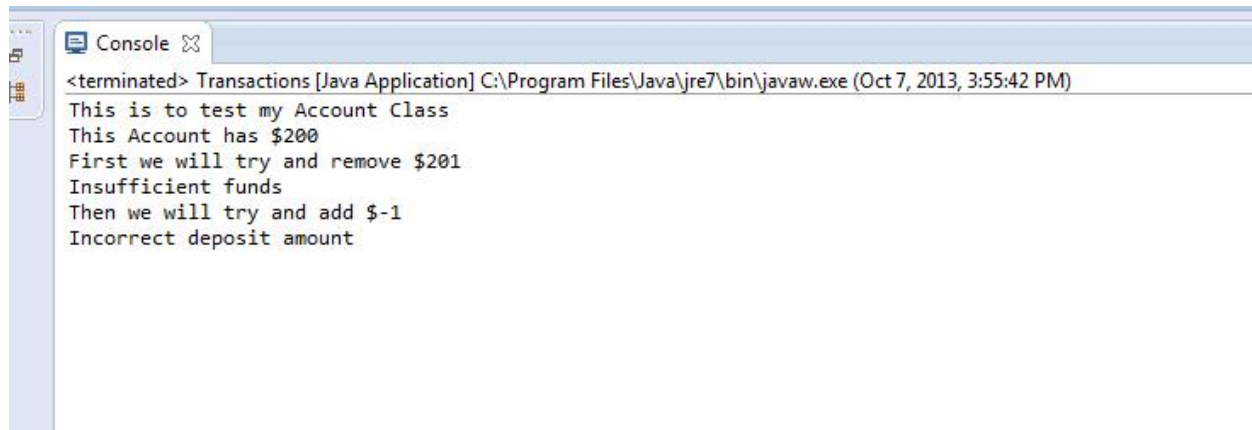
To modify this account class correctly the best method would be to include an if else series of statements for the withdraw and deposit classes. For withdraw we would use if (balance< (amount+fee)) {

System.out.println(“Insufficient funds”);} else{ balance= balance-amount-fee }

. For deposits we would use an if statement of if(amount<= 0){

System.out.println(“Incorrect deposit amount”); } else{ balance=balance + amount }.

Screen Captures:



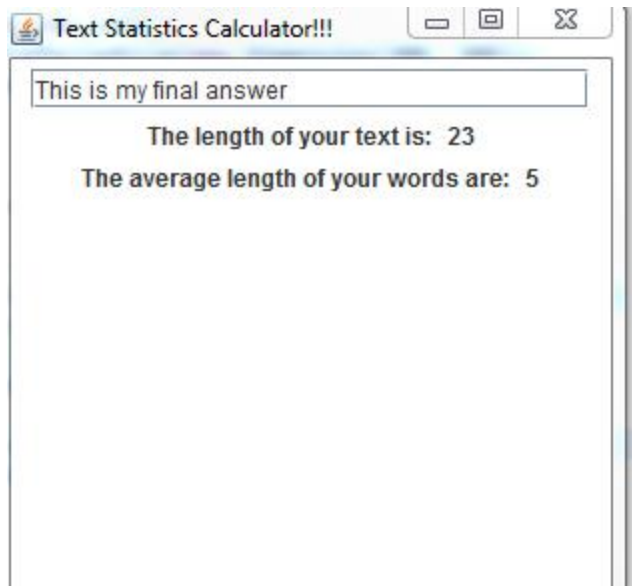
```
<terminated> Transactions [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Oct 7, 2013, 3:55:42 PM)
This is to test my Account Class
This Account has $200
First we will try and remove $201
Insufficient funds
Then we will try and add $-1
Incorrect deposit amount
```

Problem 4 Programming Project 5.16

Develop a simple tool for calculating basic statistics for a segment of text. The application should have a single window with a scrolling text box (a JTextArea) and a stats box. The stats box should be a panel with a titled border, containing labeled fields that display the number of words in the text box and the average word length, as well as any other statistics that you would like to add. The stats box should also contain a button that, when pressed, re-computes the statistics for the current contents of the text field.

(Special note for this problem) : I went online after having my program do everything but calculate the number of words in the String to see how this is supposed to be programmed into our code. I found a site that showed a more complex way to calculate this than what was covered in class. I then researched how this method worked and understood the method behind it. For this program most of what was calculated was just string manipulation methods covered in the book such as string.length() but for the number of words I found the function string.split("\\s+").length(). What this function does is it takes the number of spaces in the text and finds the length of them. So that one sentence " My name." returns the space between my and the end space.

Screen Captures:



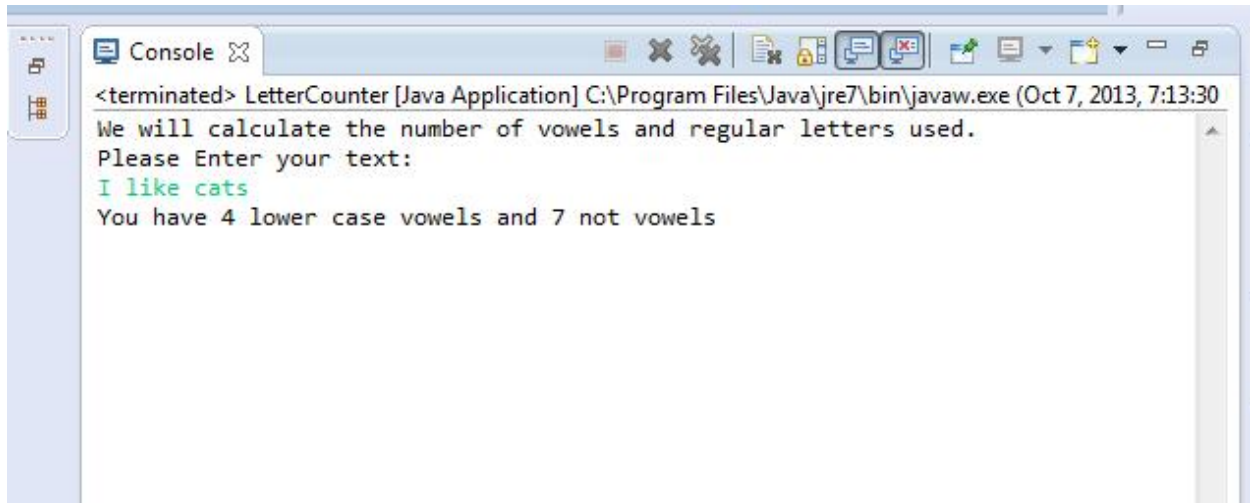
Problem 5 Programming Project 6.9

Design and implement an application that reads a string from the user, then determines and prints how many of each lowercase vowel (a, e, i, o, and u) appear in the entire string. Have a separate counter for each vowel. Also count and print the number of non-vowel characters.

How to solve:

To solve this programming assignment we have to take a look at loop structures. This loop will take the string input from the user and break each character down until the string ends. Then we will use the `string.substring(counter, counter+1)` function that will break down each variable in the string so that we can then use an if statement to compare the value to a vowel. We then increment the counter and report the value to the user. This could also have been written with nested loops, but the `substring(counter, counter+1)` is an easier method for me to use.

Screen Captures:



```
<terminated> LetterCounter [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Oct 7, 2013, 7:13:30)
We will calculate the number of vowels and regular letters used.
Please Enter your text:
I like cats
You have 4 lower case vowels and 7 not vowels
```

Problem 6 Programming Project 6.18

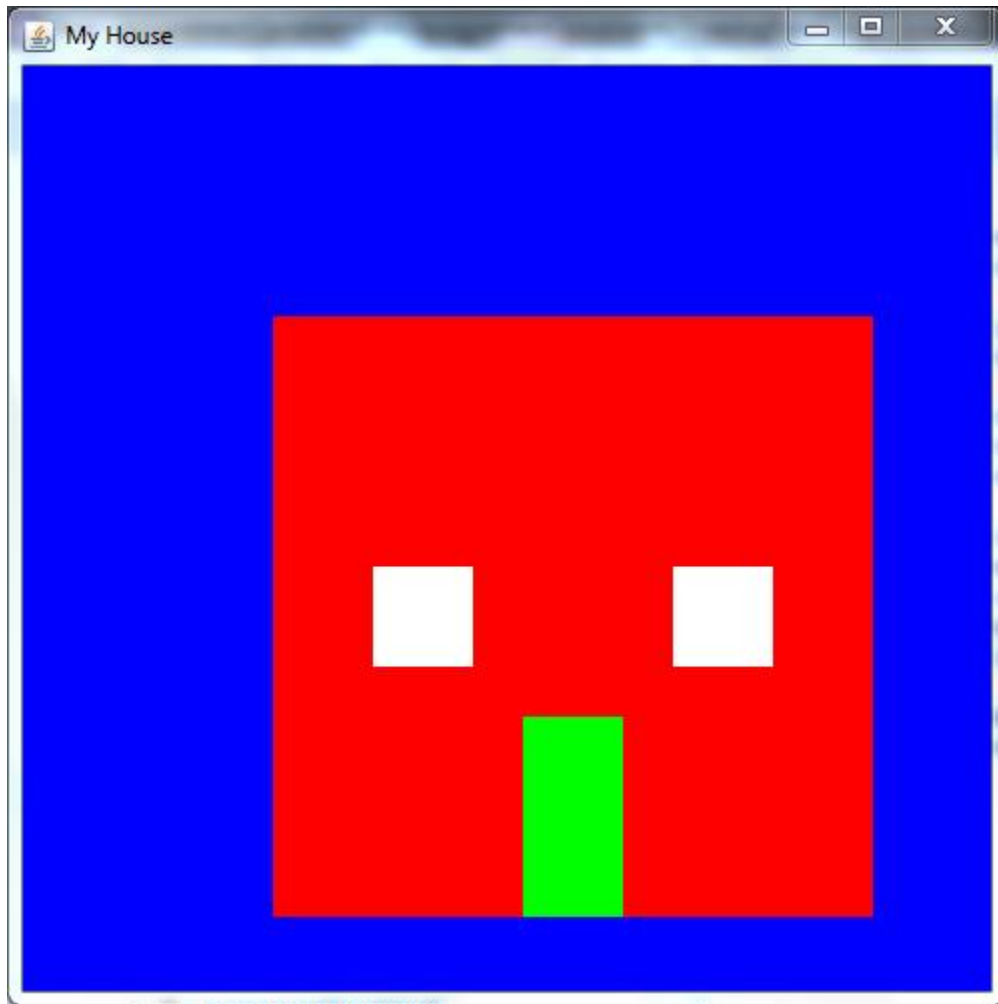
Design and implement a program that draws a simple fence with vertical, equally spaced slats backed by two horizontal support boards. Behind the fence show a simple house in the background. Make sure the house is visible between the slats in the fence.

How to solve:

First we want to design a horizon in the middle of the gui. This horizon will be defined in my example blue rectangle background. Then I will place rectangle representing a house with a triangle roof. For the house I will use the color red. I will also draw other squares for windows and color them white and place a door colored green on the house rectangle. Then using a loop and counter I will place the vertical fence pieces colored white and two horizontal fence pieces.

Screen Captures

(before loop fence to verify position)



(Completed House)

