/\*

\* CounterDemo.java

\* Hunter Damron

\* Honor Code: I didn't cheat

\* Purpose: Utilizes the Counter class to make a counter capable of clicking,

\* undoing, reseting, and displaying the counter's value

\*/

**import** javax.swing.JOptionPane;

**public** **class** CounterDemo {

**public** **static** **void** main(String[] args) {

Counter click = **new** Counter();

String input = "junk";

**while**(!(input.matches("quit"))) {

**if**(input.matches("")) {

click.click();

}

**if**(input.matches("reset")) {

click.reset();

}

**if**(input.matches("undo")) {

click.undo();

}

**if**(input.matches("-h")) {

JOptionPane.*showMessageDialog*(**null**, "This program is used as a basic counter.\n"

+ "Commands are:\n\"quit\" to end the program\n"

+ "\"undo\" to undo the last click\n"

+ "\"reset\" to set the counter back to zero\n"

+ "and \"-h\" to show this message dialog");

}

input=JOptionPane.*showInputDialog*("Counter reads " + click.getValue() +

"\nPress enter to coninue. (-h for help dialog and list of commands)");

}

}

}

Output:

Before input:

Counter reads 0

Press enter to continue. (-h for help dialog and list of commands)

With blank input:

Counter reads <counter + 1>

Press enter to continue. (-h for help dialog and list of commands)

With input “undo”:

Counter reads <counter – 1, if counter is greater than 0>

Press enter to continue. (-h for help dialog and list of commands)

With input “reset”:

Counter reads 0

Press enter to continue. (-h for help dialog and list of commands)

With input “-h”:

This program is used as a basic counter.

Commands are:

“quit” to end the program

“undo” to undo the last click

“reset” to set the counter back to zero

And “-h” to show this message dialog

R3.1. Public interface is the range of methods the user can call: getValue(), click(), and reset(); plus the constructor. Implementation is the content of the class which determines how the methods will execute when called by the user.

R3.3. The word *private* protects the variable because it can only be manipulated by methods of the same class.

R3.4. *private double* <number grade> and *private int* <grading scale (points between each letter)>

R3.7. No, there should not be a setValue() mutator because that allows the user to mutate the variable however he chooses, and if that is what is desired in the code, then it should just have *value* as a public variable