

IST 256
Lab
Week 5 – Wednesday, February 17, 2010

1. Writing a GUI Application with a method

In this lab, we will create a project with a GUI to manage the sales of books and magazines. Start a new project and name it something like SalesMethods (for sales with methods). Then create a JFrame Form and name it something like SalesMethodsGUI

In this first part of the lab, we will work together to create the form and the program that allows the user to buy books, and in the next part of the lab, you will add magazines to the form.

Make the Form

This project will have a user interface that assumes that the user can buy two different types of items, books and magazines. We will first create the form for books, with a user interface that looks something like this:

```
Enter Number Books at $20      |_____|  
(Get 10% off for 3 or more!)  
  
|__Checkout __|      Books: |_____|  
  
Total: |_____|
```

For the first instructions, you can either create two labels for the two lines of text, or write a multiline label using html tags to put the extra line (as demonstrated).

Create the Event -> Action -> actionPerformed method for the Checkout button.

Write the a method to compute the price of a book order

Go to the source tab and in the program, before the actionPerformed method, we will write a method called computePrice. Here is a description of the method that you can use as a comment block in your program.

```
/*  
 * This method computes the price of an order, where the discount is applied  
 * if the number of items is three or more  
 * parameters: number of items  
 *             price per item  
 *             discount rate  
 * result:    price of order  
 */
```

Write the method header line, giving the scope keyword “public”, the return type, the procedure name, and the formal parameter list. Try to pick good variable names for the formal parameters in the heading of the procedure.

Now the **body of the procedure** will compute the price of the order.

- Create a local variable to keep the order price.
- Compute the order price by multiplying the number of items times the item price.
- Compute the discount: if the number of items is greater than or equal to 3, an amount, which is the discount rate multiplied by the order price, is subtracted from the order price.
- Return the order price as the result of the method.

Write the Checkout Button actionPerformed method

The button method must get the number of books typed by the user into the TextField, compute the price of the books by calling the computePrice method, and display the resulting price both as a book price in the TextField labeled Books, and as the total order price in the TextField labeled Total.

Declare variables for the number of books, the price of the book order, and the total order. (What types are these variables?)

In getting the value for the number of books, we will first make sure that if the user doesn't type anything (which will result in any empty string in the TextField), we detect that and don't try to convert the empty string to an integer, which causes a run-time error. In the case that the user types nothing into the TextField, let's assume that means that the number of books is zero. An if test for this will start with the line:

```
if (jTextField1.getText().equals( "" ))
```

Of course, in this case we are using TextField1, but in general you can use the right number of textfield. Also note that if you copy this line from the pdf or doc file, you must re-type the quotes in NetBeans. The rest of the if statement will set the number of books to be 0 if the condition is true and get the number of books from the textfield otherwise.

Next, write a statement that will call the method computePrice to compute the price of the books order. The actual parameter for the number of books will be the variable that we used to get the number from the textfield. The actual parameter for the item price will be the price of books, 20.00, and the actual parameter for the discount rate will be the rate for books, .10. Assign the result of the computePrice method call to the variable for the price of books. (Always make sure that the number and types of actual parameters match the formal parameters in the method heading.)

Display the price of books in the textfield labeled Books.

Assign the total price to be the price of books (later we'll add magazines.)

Display the total price in the textfield labeled Total.

Test your program

Run the program for several values for the number of books and note the results. For one trial, just leave the textfield empty, i.e. don't type anything.

TextField contains	Total
<nothing>	
0	
1	
2	
3	

2. Extending the Program to also have magazine sales (work with your programming partner).

Adding to the Form

Add a label (or two) and a TextField under the Books label and textfield for the user to enter the number of magazines: (Note that you can copy and paste the GUI elements.)

Enter Number Mags at \$3 | _____ |
(Get 5% off for 3 or more!)

Then add the label Magazines: and a textfield after the Books textfield in the checkout section of the form:

Magazines: | _____ |

Add to the Checkout Button actionPerformed method:

In the source tab, add code to the actionPerformed method.

Add variable declarations for a variable to have the number of magazines and a variable to have the price of magazines.

Add a similar if statement to check if the textfield is empty and to get the number of magazines.

Next, add a statement that will call the method computePrice to compute the price of the magazines order. The actual parameter for the number of magazines will be the variable

that we used to get the number from the textfield. The actual parameter for the item price will be the price of magazines, 3.00, and the actual parameter for the discount rate will be the rate for magazines, .05. Assign the result of the computePrice method call to the variable for the price of magazines.

Display the price of magazines in the textfield labeled Magazines.

Change the assignment statement for the total order to be the price of books added to the price of magazines.

Test the Program

Run the program and test the results for magazines and the total. Always type the number 2 for the number of books and test the number of magazines.

Book TextField	Mag TextField	Books	Magazines	Total
2	<nothing>			
2	0			
2	1			
2	2			
2	3			

3. (Optional) Format the price numbers as currency.

Look at the currency format example shown in class on Monday, and modify your program to format all the monetary amounts as currency whenever they are displayed in a textfield.

Put the statement “import java.text.*;” in your program near the top. It should come after the package statement and before the first public class statement.

Next, put the statement that starts DecimalFormat to create the currency formatter into your actionPerformed method; a good spot would be right after the variable declarations.

Finally, change each statement of the form jTextFieldn.setText(...) to use the currencyFormatter.format method to convert the double variable to a string, replacing the method String.valueOf.

Be prepared to answer questions about your program and your results.