IST 256
Lab
Week 6 – Monday, October 3, 2011

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 __| (button)

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 in class). (Note: a multi-line label has the text <html> label line 1
 label line 2.)

Form component names

As a form gets more complex, it is helpful to rename the components to have more meaningful names to the program than jTextField1, jTextField2, etc. So we will rename our components. For each TextField, right click on the TextField and select Change Variable Name. A box will come up where you can type a more meaningful name, for example, you may call the TextFields "numberBooksTF", "displayBooksTF", and "disaplyTotalTF". You can also change the name of the button to be "CheckoutButton".

Now create the Event -> Action -> actionPerformed method for the Checkout button, and notice that the name of the button method contains the new name of the button.
(We won't change the names of these labels because we are not using any of them in the program (with a setText()), but if you are going to use them, you can change the name in the same way.

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

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

```
/*
    * This method computes the costof 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:      cost of order
    */
```

The method that we wrote in class has the following:

We wrote the **method header line**, giving the scope keyword "public", the return type, the procedure name, and the formal parameter list.

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

- We created a local variable to keep the order price.
- We computed the order price by multiplying the number of items times the item price.
- We computed 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.
- We returned the order price as the result of the method.

Write the Checkout Button actionPerformed method

The button method must do the following steps:

1. get the number of books typed into the TextField called numberBooksTF,
2. compute the cost of the books by calling the computeCost method,
3. display the resulting cost both as a book cost in the TextField called displayBooksTF and as the total order cost in the TextField called displayTotalTF.

Here is more detail about how to do these steps:

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

1. If the TextField is empty (has value ""), the number of books is 0, otherwise get the number of books from the TextField called numberBooksTF.

2. Next, write a statement that will call the method computeCost to compute the cost 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 computeCost 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.)

3. Display the results, formatting as currency.

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 NumberFormat to create the currency formatter into your actionPerformed method; a good spot would be right after the variable declarations.

```
NumberFormat cf = NumberFormat.getCurrencyInstance();
```

Display the cost of books in the textfield named displayBooksTF, using cf.format(costBooks).

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

Display the total cost in the textfield named displayTotalTF, using cf.format(totalCost).

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
0	
1	
2	
3	

2. Extending the Program to also have magazine sales

Adding to the Form
Add a label (or two) and a TextField under the numberBooks label 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: |_____|

Rename your two new textfields to be something like numberMagsTF and displayMagsTF.

Add to the Checkout Button actionPerformed method:

Using the same button, we will also get the number of magazines and use the same computeCost method to gets its cost with discount. So in the source tab, add code to the actionPerformed method.

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

1. Add a statement to get the number of magazines from the numberMagsTF textfield.

2. Next, add a statement that will **call the method computeCost** again to compute the cost 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 computeCost method call to the variable for the cost of magazines.

3. Display the cost of magazines in the textfield named displayMags, formatting as currency.

Change the assignment statement for the total cost to be the cost of books added to the cost 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	0			
2	1			
2	2			
2	3			

3. (Optional Bonus Exercise) Design a method that will get the number from a textField but make it 0 if the text field is empty.

Note that a TextField can be a parameter to a method – here is an example of a method header with a TextField parameter. The type of a TextField can be given as javax.swing.JTextField:

```
private int getNumber(javax.swing.JTextField numberTF)
```

Call this method in your CheckoutButton to get the number from the numberBooksTF and the numberMagsTF.

Hand in your lab sheets with the program for the Checkout Button by the beginning of class next Monday, October 10.