FEYZİYE MEKTEPLERİ VAKFI
1885
IŞIK ÜNİVERSİTESİ

COMP 2502 HCI Homework II Report

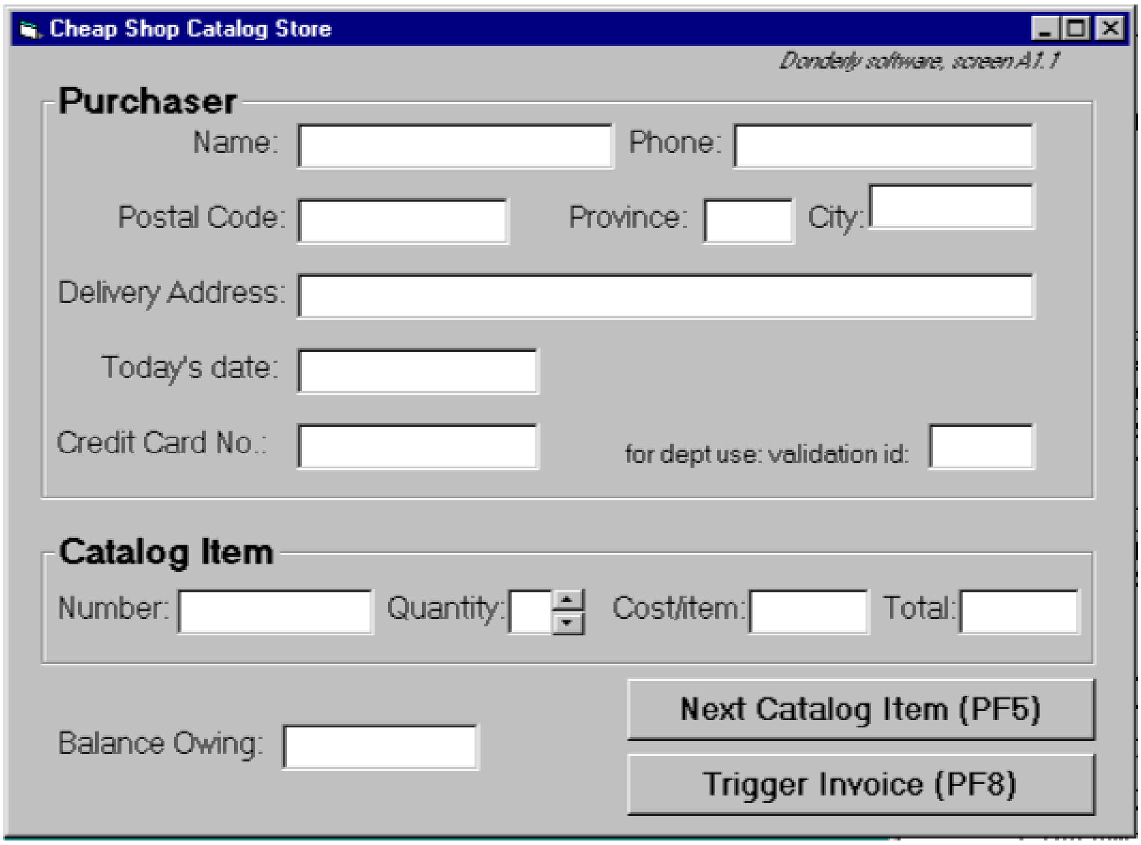
by Yağız Efe Atasever – 19COMP1035

In this report, I will try to explain how I did the requested homework, in two parts: Design and implementation.

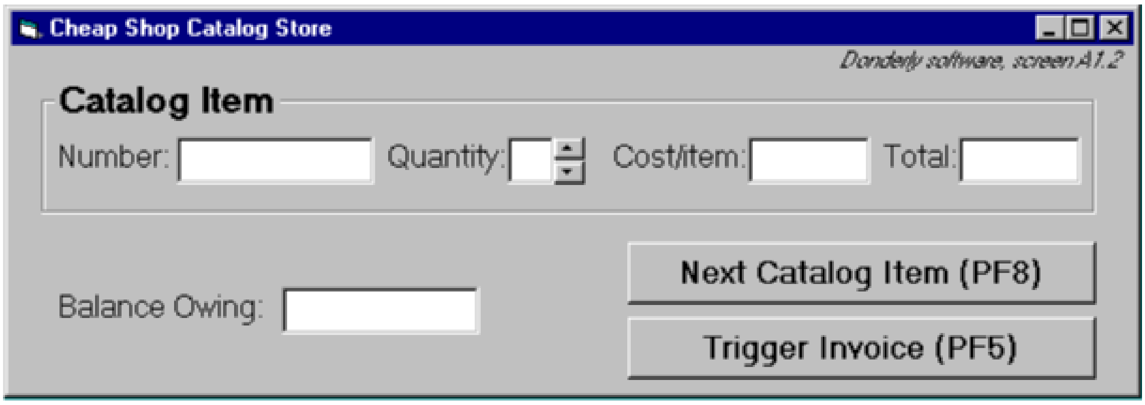
DESIGN

Everything started with me, thinking "how can I do such program?". So I opened the Homework doc file, which tries to explain what the students have to do in this homework. In the second page, there were two screen examples.

Screen 1



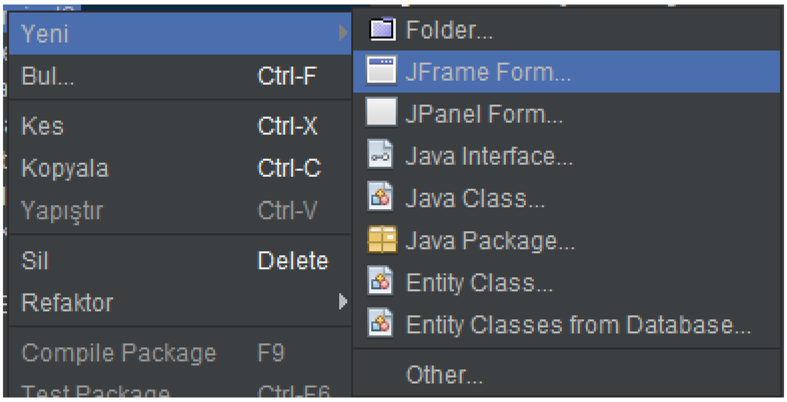
Screen 2



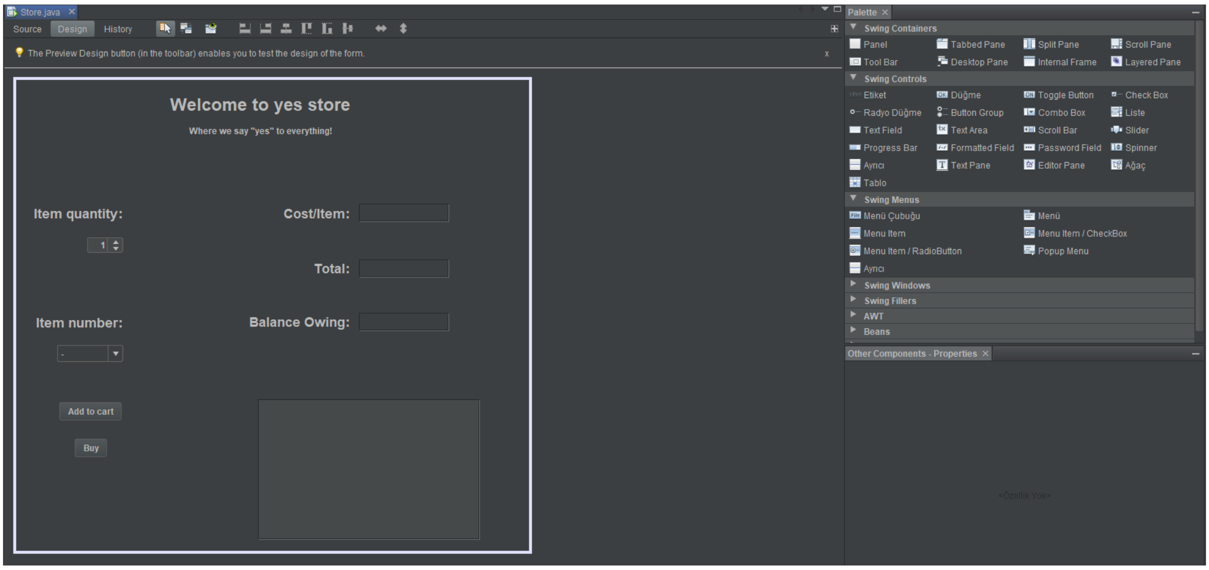
In the lecture hours, our teacher Mr. Sadık Fikret Gürgen told us the first screen is not necessary. So I do not have to do that, that is nice. So I started the designing part of the second screen which is a lot similar to the one in the homework explanation doc file. Of couse there will be some differences but the general structure is like that.

Implementation

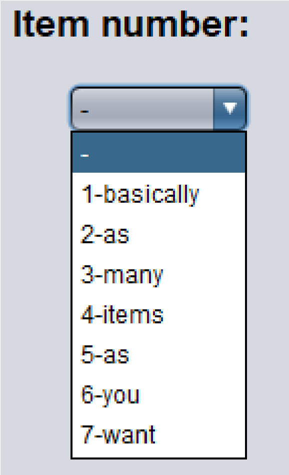
In the first homework, I did everything from scratch, every single line of code by myself. But then I realised that this is not necessary, at all. I discovered I don't have to create a class. If I create a JFrame form, my job would be more simplier, so I did it.



After that, I put some buttons, text panes, one spinner, one combo box, and one text area. Then I took the "editable" and "focusable" properties from the text panes and text area. I renames all the variables for ease coding, by right clicking them and clicking "change variable name".

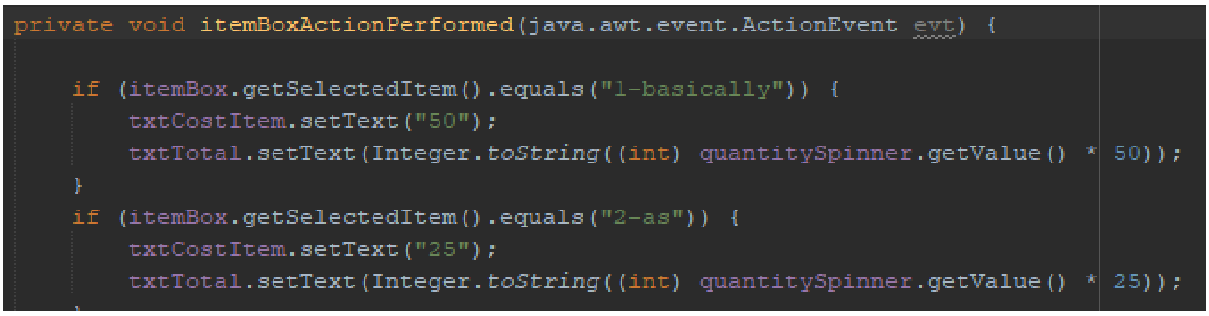

The Preview Design button (in the toolbar) enables you to test the design of the form.

First off all, I limited the Item quantity spinner, so it won't go below zero. Because you can't buy –1 items in one store. After that I added some items to buy, in the Item number combo box.



As a small funny thing, I named the items like that. Every item is a word in the sentence "Basically as many items as you want.". Apart from the humor, one other reason of the naming the items like this, is show you I can put as many items as I want to the store, even though they would be in a combo box.

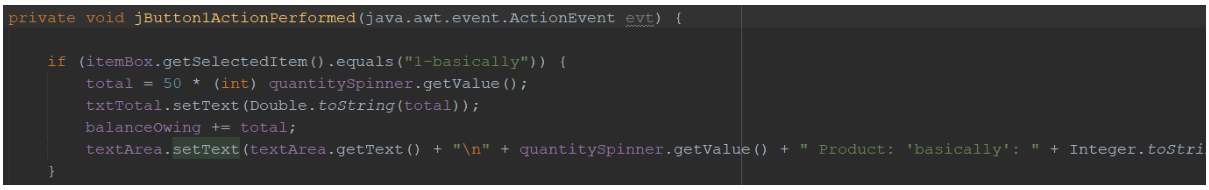
As you know, the person would want to know the prices, before they buy the item. So I wrote the code below in the combo box, so when the customer chose an item in the combo box, they will see the price int the "Cost/Item" part. (I wrote the code blow for every item.)



```
private void itemBoxActionPerformed(java.awt.event.ActionEvent evt) {

    if (itemBox.getSelectedItem().equals("1-basically")) {
        txtCostItem.setText("50");
        txtTotal.setText(Integer.toString((int) quantitySpinner.getValue() * 50));
    }
    if (itemBox.getSelectedItem().equals("2-as")) {
        txtCostItem.setText("25");
        txtTotal.setText(Integer.toString((int) quantitySpinner.getValue() * 25));
```

Then I wrote the "Add to cart" button, which shows us to total price of your items. For example you added your cart 2 of the same item, you will see the price of that item doubled. Apart from that this button also shows us what we added to cart in a text area, which will be our invoice. Below you can see a glimpse of the code. Of course I wrote this code for every item. I couldn't fit all of the code because of the low resolution of the picture.

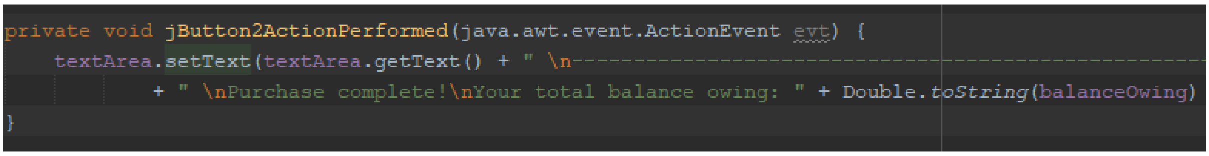


```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    if (itemBox.getSelectedItem().equals("1-basically")) {
        total = 50 * (int) quantitySpinner.getValue();
        txtTotal.setText(Double.toString(total));
        balanceOwing += total;
        textArea.setText(textArea.getText() + "\n" + quantitySpinner.getValue() + " Product: 'basically': " + Integer.toStri
    }
```

Well, while approaching to the end, we have everthing working. Of course with a small problem. Customer has to pick the quantity first. Otherwise the “total” value sometimes won’t show proper results but It is not a big problem. The interface designed for the customer to pick the quantity first by butting the quantity spinner on top of everything.

Finally I added a “Buy” button to show proper invoice. Only thing it does is showing “balance owing” to the customer as a normal invoice would do.



```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    textArea.setText(textArea.getText() + " \n---------------------------------------------------
            + " \nPurchase complete!\nYour total balance owing: " + Double.toString(balanceOwing)
}
```

So, for a small recap, I used a JFrame form, than I added the necessary elements and modified them. Added my items and prices of them. Solved “the small problem” by making customer pick the quantity first, and gave him a invoice.

Most important thing of all, I learned so much while making the program. And not everything from the internet. I worked like trial and error method and by this way, I learned a ton of new and usefull information.

Source

https://www.youtube.com/watch?v=VH61xyKfAbo&list=WL&index=57&t=1726s

Other than this video (which I watched like 2 minutes max) I learned everything by myself.