

Nov 10, 14 13:54

## AssEx1.java

Page 1/1

```

import javax.swing.JOptionPane;

public class AssEx1
{
    public static void main(String[] args)
    {
        // Create input dialog boxes to get customer name and balance
        String custName = null;

        /*Try-catch block for nullpointer exceptions (i.e. press cancel
or cross off) between these two,
        * the name dialog box works as specified, but i don't completel
y understand how they implement the functionality
        */
        try
        {
            custName = JOptionPane.showInputDialog( "Please tell me your na
me: ").trim();
        }
        catch (NullPointerException m)
        {
            System.exit(0);
        }

        //Test to determine if a non-empty name was input
        if (custName.isEmpty())
        {
            System.exit(0);
        }

        // Ask for initial balance, catch exceptions until user exits or
enters
        // a valid number
        Double initialBalance = null;

        while (initialBalance == null)
        {
            try
            {
                initialBalance = Double.parseDouble(JOptionPane.
showInputDialog( "Please tell me your current balance, using a '-' sign if you are in credit: " ));
            }

            //Catch non-double input in dialog box
            catch (NumberFormatException e)
            {
                JOptionPane.showMessageDialog(null, "Please enter a n
umeric balance", "Error", JOptionPane.ERROR_MESSAGE);
            }

            //Catch section for empty balance dialog
            catch (NullPointerException n)
            {
                System.exit(0);
            }
        }

        // Create a CustomerAccount object to store the input va
lues from the JOptionPane
        CustomerAccount nameBal = new CustomerAccount(custName,
initialBalance);

        // Instantiate the GUI, passing CustomerAccount object a
s parameter in constructor
        LWMGUI transaction = new LWMGUI(nameBal);
        transaction.setVisible(true);
    }
}

```

Nov 10, 14 13:54

## CustomerAccount.java

Page 1/2

```

public class CustomerAccount
{
    // Create instance variables
    private static final double serviceCharge = 1 - 0.2;
    private double costBottle = 0.0, currentBalance = 0.0;
    private int numBottles = 0, transactionCost = 0;
    private String custName = "";
    private Wine winTrans;

    // Constructor for customer name and initial balance. These values come
from
    // JOptionPane used only once each.
    public CustomerAccount(String name, double initBal)
    {
        custName = name;
        currentBalance = initBal;
    }

    // Helper method to calculate sale balance
    private void saleBalance()
    {
        double saleBal = currentBalance + ((transactionCost / 100) + ((t
ransactionCost % 100) / 100.0));
        currentBalance = saleBal;
    }

    // Helper method to calculate return balances
    private void returnBalance()
    {
        double returnBal = currentBalance - ((transactionCost) / 100 + (
(transactionCost) % 100) / 100.0)*serviceCharge;
        currentBalance = returnBal;
    }

    // Accessor methods

    // Return customer name and initial balance
    public String getName()
    {
        return custName;
    }

    public double getInitBal()
    {
        return currentBalance;
    }

    // Return total cost of most recent customer purchase
    public double getSaleCost()
    {
        return ((transactionCost / 100) + ((transactionCost % 100) / 100
.0));
    }

    // Return total cost of most recent customer return
    public double getReturnCost()
    {
        return ((transactionCost) / 100 + ((transactionCost) % 100) / 10
0.0)*serviceCharge;
    }

    // Return current balance after a customer return
    public double getReturnBal()
    {
        returnBalance();
        return currentBalance;
    }

    // Return current balance after a customer purchase
    public double getSaleBal()
    {
        saleBalance();
    }
}

```

Nov 10, 14 13:54

**CustomerAccount.java**

Page 2/2

```

        return currentBalance;
    }

    //Method to get Wine object as a parameter
    public void setWineObj(Wine newObj)
    {
        this.winTrans = newObj;
        costBottle = winTrans.getBottlePrice();
        numBottles = winTrans.getNumBottles();
        transactionCost = (int) (costBottle * 100) * numBottles;
    }
}

```

Nov 13, 14 16:06

**LWMGUI.java**

Page 1/6

```

/*Import the required java classes for constructing
 * and managing a GUI
 */

import javax.swing.*;

import java.awt.*;
import java.awt.event.*;

//Tell java we will be adding to JFrame class methods with our own methods

public class LWMGUI extends JFrame implements ActionListener

{
    // Instance variables
    private JButton saleButton, pReturnButton;
    private JLabel label;
    private JTextField wineNameInput, quantityBottlesInput, bottlePriceInput
    ,
        transAmount, currBalField;
    private String wineName;
    private double price;
    private double initialBalance;
    private int quantity;
    public Wine winTrans;
    public CustomerAccount saleReturn;
    private boolean validData;

    // Constructor for creating GUI with correct details
    public LWMGUI(CustomerAccount nameBal)
    {
        // Store CustomerAccount object
        saleReturn = nameBal;

        // Get variables from CustomerAccount object
        String customerName = saleReturn.getName().trim();
        initialBalance = saleReturn.getInitBal();

        // Create JFrame for adding layout components to
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(800, 400);
        setLocation(450, 300);
        setTitle("Lilybank Wine Merchants: " + customerName);

        // Call helper methods for layout
        layoutGuiTop();
        layoutGuiMiddle();
        layoutGuiBottom();
    }

    // Helper method (layoutGuiTop) to layout components at top of JFrame wi
ndow

    private void layoutGuiTop()
    {
        // Create JPanel and JTextField components
        JPanel northArea = new JPanel();
        northArea.setBackground(Color.red);
        wineNameInput = new JTextField(20);
        quantityBottlesInput = new JTextField(10);
        bottlePriceInput = new JTextField(10);

        // Wine name JLabel
        JLabel wineNameLabel = new JLabel("Wine Name:");
        wineNameLabel.setFont(new Font("Sans_Serif", Font.PLAIN, 20));
    }
}

```

Nov 13, 14 16:06

LWMGUI.java

Page 2/6

```

    // Bottle quantity JLabel
    JLabel quantityBottlesLabel = new JLabel("Quantity:");
    quantityBottlesLabel.setFont(new Font("Sans_Serif", Font.PLAIN, 20
));

    // Bottle price JLabel
    JLabel bottlePriceLabel = new JLabel("Price:£");
    bottlePriceLabel.setFont(new Font("Sans_Serif", Font.PLAIN, 20));

    // Add JPanel to JFrame NORTH
    this.add(northArea, BorderLayout.NORTH);

    // Add components to JPanel according to flow layout
    northArea.add(wineNameLabel);
    northArea.add(wineNameInput);
    northArea.add(quantityBottlesLabel);
    northArea.add(quantityBottlesInput);
    northArea.add(bottlePriceLabel);
    northArea.add(bottlePriceInput);
}

// Helper method layoutGuiMiddle to layout components in middle of JFram
e window
private void layoutGuiMiddle()
{
    // Create JPanel to hold two nested JPanels with box layout (cen
tral column)
    JPanel centerArea = new JPanel();
    centerArea.setLayout(new GridLayout(2,1));

    // Create JPanels for nesting
    JPanel centerArea1 = new JPanel();
    JPanel centerArea2 = new JPanel();

    // Create sales button and add ActionListener
    saleButton = new JButton("Process Sale");
    saleButton.addActionListener(this);
    centerArea1.add(saleButton);

    //Create returns button and add actionlistener
    pReturnButton = new JButton("Process Return");
    pReturnButton.addActionListener(this);
    centerArea1.add(pReturnButton);

    // Add JPanel to JFrame CENTER
    this.add(centerArea, BorderLayout.CENTER);
    centerArea.add(centerArea1, BorderLayout.NORTH);
    centerArea.add(centerArea2, BorderLayout.SOUTH);

    // Add display label to JPanels
    label = new JLabel("");
    label.setFont(new Font("Sans_Serif", Font.PLAIN, 20));
    centerArea2.add(label, BorderLayout.SOUTH);
}

// Helper method layoutGuiBottom to layout components to bottom of JFram
e
private void layoutGuiBottom()
{
    // Create JPanel
    JPanel southArea = new JPanel();
    southArea.setBackground(Color.red);

```

Nov 13, 14 16:06

LWMGUI.java

Page 3/6

```

    // JLabel for textfield displaying amount of most recent transac
tion
    JLabel transactionAmount = new JLabel("Amount of transaction: £");
    transactionAmount.setFont(new Font("Sans_Serif", Font.PLAIN, 20));

    // JLabel for textfield displaying account balance, including mo
st
    * recent transaction
    */
    JLabel currBalance = new JLabel("Current Balance: £");
    currBalance.setFont(new Font("Sans_Serif", Font.PLAIN, 20));

    // JTextfields for recent transaction and current balance both u
neditable by user
    transAmount = new JTextField(7);
    transAmount.setEditable(false);

    // Test to determine whether customer is in credit and format JT
extField accordingly
    if (initialBalance < 0.0)
    {
        String s = String.format("%.2f", (initialBalance/-1));
        currBalField = new JTextField(" + s + "CR", 9);
    }
    else
    {
        String s = String.format("%.2f", initialBalance);
        currBalField = new JTextField(" + s", 9);
    }
    currBalField.setEditable(false);

    // Add JPanel to JFrame SOUTH
    this.add(southArea, BorderLayout.SOUTH);

    // Add components to JPanel in SOUTH
    southArea.add(transactionAmount);
    southArea.add(transAmount);
    southArea.add(currBalance);
    southArea.add(currBalField);
}

//Action performed to tell the program what to do depending on which but
ton was pressed
public void actionPerformed(ActionEvent e)
{
    if (e.getSource() == saleButton)
        dataCaptureSales();
    else if (e.getSource() == pReturnButton)
        dataCaptureReturns();
}

/* Capture values entered into textfields as a sale, create wine object
* and use captured values as parameters
*/
private void dataCapture()
{
    wineName = wineNameInput.getText().trim();
    validData = true;

    //Obtain bottle price and check if a double was input
    try
    {
        price = Double.parseDouble(bottlePriceInput.getT
ext());
    }
    catch (NumberFormatException p)

```

Nov 13, 14 16:06

LWMGUI.java

Page 4/6

```

        {
            JOptionPane.showMessageDialog(null, "Please enter a valid price", "Error", JOptionPane.ERROR_MESSAGE);
            clearTextFields();
            validData = false;
        }

        //Obtain quantity of bottles and test whether an integer was entered
        try
        {
            quantity = Integer.parseInt(quantityBottlesInput.getText());
        }
        catch (NumberFormatException q)
        {
            JOptionPane.showMessageDialog(null, "Please enter a valid Quantity", "Error", JOptionPane.ERROR_MESSAGE);
            clearTextFields();
            validData = false;
        }
    }

    //Helper method to obtain values from a sales transaction
    private void dataCaptureSales()
    {
        //Error handling for checking input values

        dataCapture();
        //Test here for empty name and values less than zero, to prevent incorrect processing
        if (quantity < 0 || price < 0.0 || wineName.isEmpty() || !validData)
        {
            JOptionPane.showMessageDialog(null, "Please enter valid data", "Error", JOptionPane.ERROR_MESSAGE);
            quantity = 0;
            price = 0.0;
            wineName = "";
            clearTextFields();
        }
        else
        {
            saleMethod();
        }
    }

    //Helper method to obtain values from a return transaction
    private void dataCaptureReturns()
    {
        //Error handling for checking input values
        dataCapture();

        //Test here for valid data to prevent invalid processing
        if (quantity < 0 || price < 0.0 || wineName.isEmpty())
        {
            JOptionPane.showMessageDialog(null, "Please enter valid data", "Error", JOptionPane.ERROR_MESSAGE);
            quantity = 0;
            price = 0.0;
            wineName = "";
            clearTextFields();
        }
        else
        {
            returnMethod();
        }
    }

```

Nov 13, 14 16:06

LWMGUI.java

Page 5/6

```

    }

    //Helper method to display sales
    private void saleMethod()
    {
        // Create wine object, pass to CustomerAccount as a sale
        winTrans = new Wine(wineName, price, quantity);
        saleReturn.setWineObj(winTrans);

        //Get balance from CustomerAccount object
        double saleBalance = saleReturn.getSaleBal();

        // Test to see if balance is in credit (merchant owes customer)
        if (saleBalance < 0.0)
        {
            // Format and display credit balances
            String s = String.format("%.2f", saleBalance / -1);
            currBalField.setText("" + s + "CR");
        }
        else
        {
            // Format and display debit balances
            String s = String.format("%.2f", saleBalance);
            currBalField.setText("" + s);
        }

        // Format and display most recent sale cost
        String t = String.format("%.2f", saleReturn.getSaleCost());
        transAmount.setText("" + t);

        // Display name of wine bought and clear textfields
        label.setText("Wine purchased: " + wineName);
        clearTextFields();
    }

    //Helper method to display returns
    private void returnMethod()
    {
        //Set wine object in CustomerAccount as a sale
        winTrans = new Wine(wineName, price, quantity);
        saleReturn.setWineObj(winTrans);

        double returnBalance = saleReturn.getReturnBal();

        /*
         * Display transaction amount and current balance (from CustomerAccount)
         * method) Test to see if balance is in credit (merchant owes customer)
         * and format accordingly
         */
        if (returnBalance < 0.0)
        {
            String s = String.format("%.2f", returnBalance / -1);
            currBalField.setText("" + s + "CR");
        }
        else
        {
            String s = String.format("%.2f", returnBalance);
            currBalField.setText("" + s);
        }

        // Format and display return transaction amount
        String t = String.format("%.2f", saleReturn.getReturnCost());
        transAmount.setText("" + t);
    }

```

Nov 13, 14 16:06

LWMGUI.java

Page 6/6

```
        // Display name of wine returned and clear textfields
        label.setText("Wine returned: " + wineName);
        clearTextFields();
    }

    //Helper method to clear textfields
    private void clearTextFields()
    {
        wineNameInput.setText("");
        quantityBottlesInput.setText("");
        bottlePriceInput.setText("");
    }
}
```

Nov 11, 14 13:00

Wine.java

Page 1/1

```
public class Wine

//Create instance variables
{
    private String wineName = new String();
    private double unitPrice = 0.0;
    private int unitQuantity = 0;

    // Create constructor
    public Wine(String wName, double bPrice, int bQuantity)
    {
        wineName = wName;
        unitPrice = bPrice;
        unitQuantity = bQuantity;
    }

    // Create accessor methods
    public String getName()
    {
        return wineName;
    }

    public double getBottlePrice()
    {
        return unitPrice;
    }

    public int getNumBottles()
    {
        return unitQuantity;
    }
}
```