Colorado Technical University

IT252-1402B-01: Intermediate Java Programming II

Instructor: Jeffery Sanford

Phase 5 Individual Project

Tim Paulson

June 21, 2014

Contents

[IP1 Sales Data App 3](#_Toc391303923)

[ScreenShot 3](#_Toc391303924)

[IP2 Sales Data Application Revised 4](#_Toc391303925)

[ScreenShot 5](#_Toc391303926)

[IP3 Sales Data Application Revised 6](#_Toc391303927)

[Runtime Screenshots 7](#_Toc391303928)

[IP4 Sales Data Application Revised 8](#_Toc391303929)

[Runtime Screenshot 9](#_Toc391303930)

[Class Diagram 9](#_Toc391303931)

[IP5 Final Sales Data Application 10](#_Toc391303932)

[FINAL APPLICATION SCREENSHOT 10](#_Toc391303933)

[Competency Test 11](#_Toc391303934)

[References 12](#_Toc391303935)

# IP1 Sales Data App

The swing components that I have decided to us for my application are the Jfram, Jlable, Jtextfield, Jradiobutton. The Jframe will be used to contain all of the GUI components. The Jlable will be used to describe all of the text fields and give instructions to the user. The Jtexfield will be used to gather the user input for ID, first and last name, and the total of supplies, books and paper. The Jradiobuttons will be used to allow the user to select which sales district and preferred method of contact. There will also be a clear button that will allow the user to clear all the data on the screen if they want to start over. The Quit button will allow the user to exit the program. Last but not least the Enter button when clicked will allow the user to submit and store all the data they have entered.

Pseudocode for QUIT button:

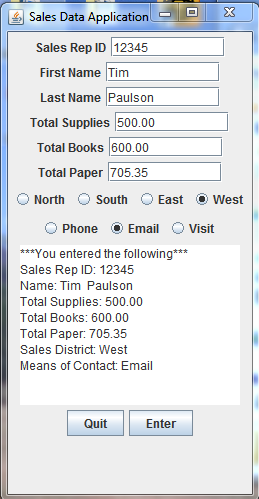
Exits the program

Pseudocode for ENTER button:

ID, first name, last name, total sold, radio buttons values are all stored into a variable

User input is displayed in the text area

## ScreenShot



# IP2 Sales Data Application Revised

At the request of Management, the programming team has revised the application to reflect the following implementations:

1. The application will now use a BufferedWriter to write and append user entered data to a file named “salesrep.txt” whenever the user clicks the ENTER BUTTON.
2. A command button, “Evaluate”, has been added to read each line of the file “salerep.txt”, and print it to the text area for review. Each time the user clicks Enter, the new data will be appended to the text file. Each time the Evaluate button is clicked, the current text displayed in the text area, if any, will be cleared and only the data currently contained within the text file will be displayed.

The following is the pseudocode for the Enter button:

All values entered are stored in variables

Try

Create file object

Create FileWriter object and allow for appending to file

Create BufferedWriter object

Use BufferedWriter object and concatenate variables to form one line of data

Close the bufferedWriter and fileWriter objects

Catch IOException ex

The following is the pseudocode for the Evaluate button:

Try

Create FileReader and BufferedReader objects

Clear text area of displayed text

While loop

Store each line of data in a variable. If the line is not null then

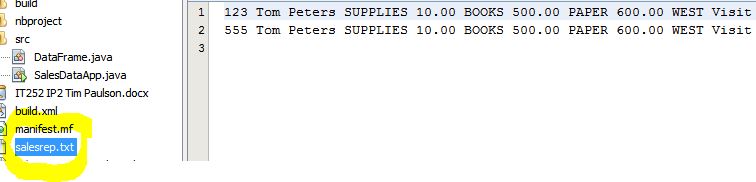
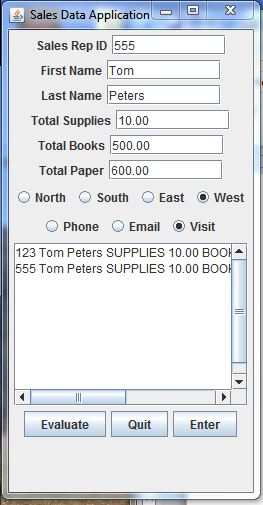
Keep appending the next line to the text area until the end of file

Is reached

Close the BufferedReader and FileReader.

Catch IOException ex

## ScreenShot



# IP3 Sales Data Application Revised

Management has again asked the programming department to revise the sale data application. The following revisions were made to satisfy the requirements set my management:

1. The Evaluate button provides additional functionality.
   1. The evaluate button will also along with previous functions, read in from the salesrep.txt file and evaluate the totals for supplies, books and paper.
   2. If the sum of each category is greater than or equal to $8000.00 then that sales reps data will be written to the stars.txt file for Bonus evaluation.
   3. Each entry in the stars.txt file will be separated by a line separator.

The following is the updated pseudocode for the Evaluate button ActionListerner, all other code remains the same:

Try

Create the file and buffered reader for the salesrep.txt file

Clear text area

Create the file and buffered writer for the stars.txt file

Enter While loop (keep reading next line as long as it is not null)

Append to the text area from salesrep.txt

Create a string tokenizer to tokenize each line of salesrep.txt

As the tokens come in parse if necessary and set tokens to variable values

Calculate the sum of sales categories and set to variable total

If >= 8000.00 then write that reps data to the stars.txt file

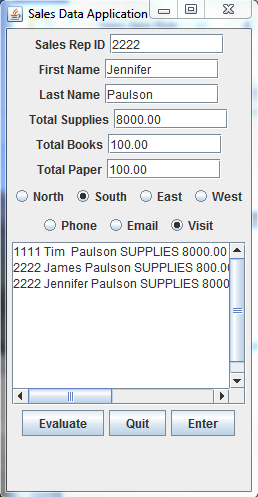
End while loop

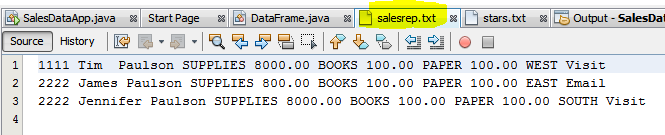
Close the buffered writer and readers and file writer and readers

End Try

Catch exceptions

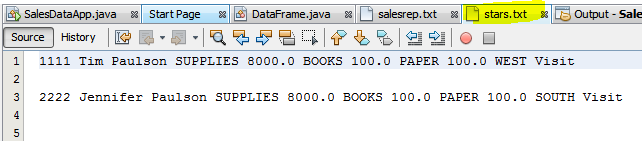
## Runtime Screenshots





Only Tim and Jennifer were written to the stars file because their sales were >= 8000.00

Line separator used.



# IP4 Sales Data Application Revised

Management has again asked the programming department to revise the sale data application. The following revisions were made to satisfy the requirements set my management:

1. A Display button was added to the GUI
   1. The display button will create an ArrayList to store objects of Type Sales Representative. It will read from the stars.txt file and create the object from each token read in. The objects will then be printed from the ArrayList and displayed in the text area using the toString method provided in the Sales Rep class.

The following is the updated pseudocode for the Display button ActionListerner, all other code remains the same:

Pesudocode:

Variable count and ArrayList are initialized

Try

FileReader and BufferedReader are instantiated to read the stars.txt file

Text area is cleared if anything is displayed

While the bufferedReader has something to read execute the following

Instantiate StringTokenizer with the bufferedReader

Set all tokens to variable values

Create the SalesRepStars object and pass variables to construct object

Add the constructed object to the arraylist

Increment the count variable

Exit while loop

Use for loop to print each object in the arrayList

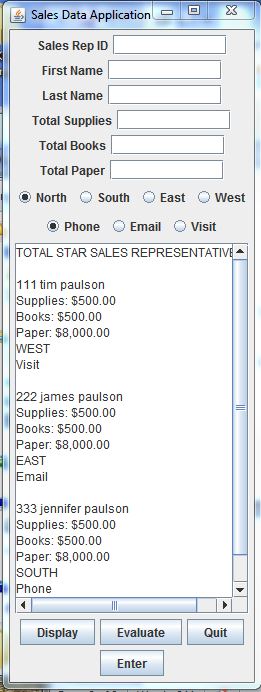
Close the bufferedReader and fileReader

## Runtime Screenshot

The following is a screenshot to demonstrate properly formatted display to the text area from the ArrayList and the created Display button.

## Class Diagram

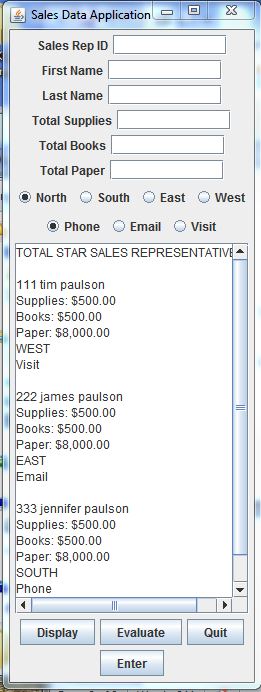
|  |
| --- |
| **SalesRepStars** |
| -books:double  -booksLable:string  -contact:string  -district:string  -fName:string  -lName:string  -id:int  -paper:double  -paperlabel:string  -supplies:double  -supplyLabel:string |
| +SalesRepStars()<<Constructor>>  +SalesRepStars(int,String,String,double,  Double,double string,  string)<<Constructor>>  +getBooks():double  +getContact():String  +getDistrict():String  +getFName():String  +getID():int  +getLName():String  +getPaper():double  +getSupplies():double  +setBooks(double):void  +setContact(string):void  +setDistrict(string):void  +setFName(string):void  +setID(int):void  +setLName(string):void  +setPaper(double):void  +setSupplies(double):void  +toString(): String |



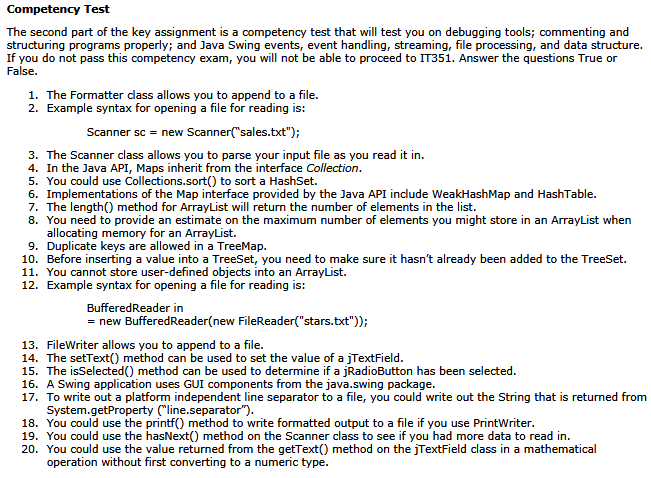
# IP5 Final Sales Data Application

The sales data application that management has requested is complete. The application will take the user’s input data and save it to file for evaluation. This information will then be read and evaluated to separate the star sales reps from the others based on total sales sold. The requirements and pseudocode for this application can be found in previous sections. By running the application you can see that the salesrep.txt and stars.txt file have been created. At the end of this document you will find the Competency Test required to proceed to the next course.

# FINAL APPLICATION SCREENSHOT



# Competency Test



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| .1 | .2 | .3 | .4 | .5 | .6 | .7 | .8 | .9 | .10 | .11 | .12 | .13 | .14 | .15 | .16 | .17 | .18 | .19 | .20 |
| T | T | T | F | F | T | F | F | F | F | F | T | T | T | T | F | F | T | T | F |

# References

Deitel, P & Deitel, H. (2012). Java™: How to Program, Ninth Edition.

Boston, Massachusetts, Pearson