Programming Assignment 9

Due: 11:59pm, Saturday, March 5

Overview

The goals of this assignment are to:

1. Write a program with no starting code
2. Handle exceptions
3. Make use of GUIs
4. Make use of external JARs and APIs
5. Create a website to promote your product

Setup

There are no starter files for this assignment. However, there are test cases for you to run and external libraries that you'll need to use. You will need to make use of the Apache POI libraries. For your convenience, I've provided the required external jars (similar to objectdraw.jar in PA8). Copy the test case files and libraries for HW9 to your directory (or your PC if developing locally):

```
$ mkdir ~/HW9
$ cd ~/HW9
$ cp /home/linux/ieng6/cs11wb/public/HW9/* .
$ cp /home/linux/ieng6/cs11wb/public/HW9_libs/* .
```

Setup Eclipse

If you haven't already, read the "Set Up Eclipse" document on the course website. To set up Eclipse for this assignment (highly recommended) you'll need to first copy the required Apache POI libraries to your local desktop. The following instructions assume you have Eclipse open.

1. Select File -> New -> Java Project
2. In the pop up dialog, name the project HW9 and hit the *Next >* button
3. Select the *Libraries* tab
4. Select *Add External JARs* button
5. Find and select the 4 provided Apache POI JAR files
6. Click to expand the *JRE System Library*
7. Select *Access Rules: No rules defined*
8. Hit the *Edit* button
9. Select *Add…* button
10. Change the Resolution to *Accessible* and Rule Pattern to *javafx/***
11. Select *Ok, Apply, Ok*

Part I: Small business needs (25 pts)

This homework is a small project I picked up from a friend that runs a musical instrument resale website. His problem consisted of:

1. He has a spreadsheet (Microsoft Excel, .xlsx format), let's call it the *master sheet*, that has all of his items that he currently sales on his website. The sheet contains multiple columns/fields, e.g.: SKU (Stock Keeping Unit), MAP (Minimum Advertised Price), List Price, Description, items in stock, etc. Of these fields, for this project we are only interested in the SKU and MAP. The SKU is guaranteed to be unique for each item. See the master.xlsx made available for you from the Setup.
2. Every month he gets an updated spreadsheet (also in .xlsx format) from each vender that he purchases items from (see vendor1.xlsx, vendor2.xlsx, vendorFail.xlsx).
    a. It is important to note that there are many items from the vender that he does not purchase, but the vender sheet lists all items sold by the vendor. For example, the vendor may sell drum sticks, pianos, and guitars, but his company only sells this vendor's drum sticks.
    b. The sheet from the vendor also has many fields, but they may have different titles or description. However, there is always a field for SKU and MAP.
        i. The column for SKUs and MAPs changes with each vendor file.
    c. The SKUs from the vendor match the SKUs in the master. For example, if the vendor's SKU for drum sticks is `a147362`, then the SKU entry in the master sheet for drum sticks would also be `a147362` (examples below).
    d. The MAPs are updated every so often by the vendor.

Every month when a new vendor sheet comes in with updated Minimum Advertised Prices, he has to go through the Master sheet and manually match SKUs found in the Vendor sheet. Once he finds a match, he then updates the appropriate MAP in the Master sheet. Due to the large number of items that his company sells, it is a huge waste of time to do this every time a new vendor sheet comes in. This is where you come in!

Your goal is to create a GUI that allows the user to select the master sheet, vendor sheet, and where to save the updated master sheet. I'm giving you full flexibility with how you want your GUI to look, however there are a few restrictions:

1. The user must be able to select a file graphically (FileChooser)
2. The user must be able to select where to save the output graphically (DirectoryChooser)
3. The user should see the path of the files (input and output) selected (TextField)
4. The user should be notified if there was an error at any stage of the program (Alert)
5. The user needs to be able to hit a "run" button to create the output files (Button)
6. The user should be notified once the output file is generated successfully (Alert)
7. You must use JavaFX. You cannot use java.swing or java.awt.

Program Input

In a perfect example the user will select a master file, a vendor file, and a path on where to save the files (or if you prefer, the path and filename of where to save the output file). However,

many things can go wrong at this stage and your program must handle any exception that comes up. For example:

A. The user selects a file that is not of type .xlsx
B. They select a .xlsx file, but it is not related to this project
C. They user doesn't select 2 input files, or where to store the file, they just hit run
D. They select an appropriate file, but there is no column for either SKU or MAP (e.g. vendorFail.xlsx)

Input Assumptions

- You can assume that even if the master/vendor file have multiple sheets, the program will only care about the first sheet.
- A valid input file will have type .xlsx (not .xls, .csv, .txt)

Program Output

Once the user hits "run", your program should find the matching SKUs in the master and vendor sheets and create a new output file in the path specified by the user. This output file can be auto named by you (the programmer), or you could allow the user to select the file name output. Here's a simple example output:

Example

Master Sheet

	A	B	C	D	E
1	SKU	MAP	Description	List Price	Stock
2	a14352	5	drums	50	30
3	c1242	6	guitar	60	23
4	d3424	7	amp	70	10
5	d2321	8	piano	80	5
6	c23414	25	saxaphone	250	3
7	z332234	30	trumpet	300	1

Vendor Sheet

	A	B	C
1	MAP	Description	SKU
2	7	drums	a14352
3	32	violin	f5532
4	39	piano	d2321
5	20	saxaphone	c23414
6	30	trumpet	z332234

Updated Master – Sheet1

	A	B	C	D	E
1	SKU	MAP	Descption	List Price	Stock
2	a14352	7	drums	50	30
3	c1242	6	guitar	60	23
4	d3424	7	amp	70	10
5	d32321	39	piano	80	5
6	c23414	20	saxaphone	250	3
7	z332234	30	trumpet	300	1

Updated Master – Sheet 2

	A	B	C	D	E	F
1	SKU	MAP	Descption	List Price	Stock	Detail
2	a14352	7	drums	50	30	MAP Increased
3	c1242	6	guitar	60	23	Not found
4	d3424	7	amp	70	10	Not found
5	d32321	39	piano	80	5	MAP Increased
6	c23414	20	saxaphone	250	3	MAP Decreased
7	z332234	30	trumpet	300	1	MAP Unchanged

You'll see that:

1. The updated master contains 2 sheets. The first sheet has the same format as the original master sheet. The MAPs for drums, piano, saxophone have been updated.
2. The second sheet is the exact same as (sheet 1) but there is a column added at the end that tells the user what was changed. You don't have to use these words exactly, but each SKU will fall under one of these four categories.

3. Not seen here is that you should have more descriptive names for the output sheets, e.g. sheet 1 can be named "master" and sheet 2 "detailed".

The user will then be notified if the updated master sheet was successfully generated, or if something went wrong. The program will stay open so that the user can continue to select additional files to run.

Apache POI

In order to work with excel files in Java, you'll need to use the external Apache POI libraries. When you're ready to test on the ieng6.ucsd.edu servers, this will be how you compile (assuming HW9.java is your starter file).

```
javac -cp "/home/linux/ieng6/cs11wb/public/HW9_libs/*:." HW9.java
java -cp "/home/linux/ieng6/cs11wb/public/HW9_libs/*:." HW9
```

Part II: Durability (10 pts)

We will be throwing every odd case we can think of at your program to try to break it. Do your best to catch every possible error a user could introduce and don't allow your program to exit with an error.

Part III: Website (10 pts)

This section will give you a chance to develop a website to promote and "sell" your product. Complete the HTML tutorial at CodeAcadamy to learn HTML + CSS basics.

Your website should have at a minimum:

- A user guide for your product (including screen shots of the program in use) and how to compile and run your product on the ieng6.ucsd.edu servers
- A main page for your tutor to open, *index.html*
- Information on costs, how to buy, marketing
- Information about you
- A list of bugs/anything you weren't able to get to work

All files for your website must be within a subdirectory *website* when submitted. See turnin instructions below for example. Note: the website doesn't have to be hosted anywhere, we just need all the files required to view your website in your *website/* directory.

Part IV: User readability (10 pts)

Same as always. Be sure to remove the Apache POI libraries from your directory before submitting your assignment.

Turnin Instructions

Remember the deadline to turn in your assignment is Saturday, March 5, by 11:59pm.
Make sure the program works correctly on the ieng6 linux servers. If you developed on your desktop with Eclipse, be sure that your program compiles and runs on the ieng6 servers before

submitting. This is the only way the tutors have to check your assignment. If it doesn't compile, you will lose almost all points.
Because there is flexibility in the file names that you use for your program, you'll need to create a README file that outlines how to compile and run your program on the ieng6 servers.

```
$ cat README
@Authors: Your Name
To compile: javac -cp "/home/linux/ieng6/cs11wb/public/HW9_libs/*:." HW9.java
To run: java -cp "/home/linux/ieng6/cs11wb/public/HW9_libs/*:." HW9
```

When you are ready to turn in your program in, type in the following command and answer the prompted questions:

```
# this is a comment, i.e. don't enter lines w/ a '#'
$ cd ~/HW9
# remove any files you don't wish to submit, e.g. *.class
$ rm *.class *.xlsx *.jar
$ cd ~/
$ bundleP9
```

POW Challenge

Your section lead will be looking for the best submission from your group. Criteria will be on correctness, user interface, and website design. The top submissions from each team will be presented during class and voted on for best in class. Some ideas to take this project even further:

- Have a splash screen when your program starts
- Use css + javafx to make your application stand out
- Use a tool to package your final product into an executable jar that can be downloaded from your webpage and run on any system. Eclipse makes this easy.