In this lab, you will learn JavaFX, inner classes, and event-handlers.

Problem statement: You created a DocAnalyst in Lab3 that parsed regular text files to give wordcount, and CSV files to give rowCount and columnCount. In this lab, you will change it to a GUI application.

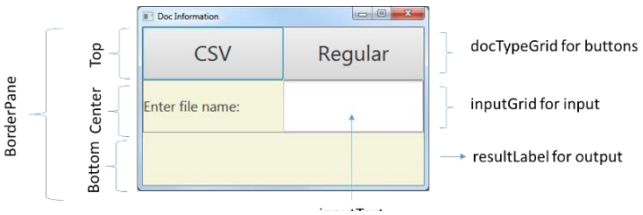


Figure 1: GUI components in opening screen

DocAnalyst analyzes text documents. It supports two types of text documents: CSV and Regular (i.e. non-CSV text file). In the given GUI (Figure 1), user enters a file name and presses a button to get appropriate information. When CSV button is pressed, the DocAnalyst gives row and column count(Figure 2). When Regular is pressed, it gives word count (Figure 3). Figure 4 and 5 show the scenarios when CSV is pressed for non-csv file, and Regular is pressed for a csv file.

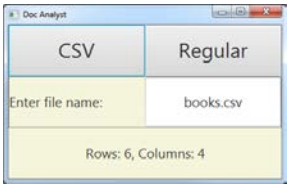


Figure 2: Enter CSV file name and press CSV button

Figure 3: Enter regular file name and press Regular button

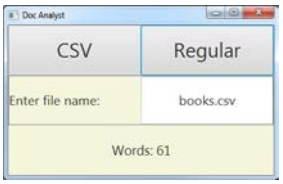


Figure 4: Enter CSV file name and press Regular button

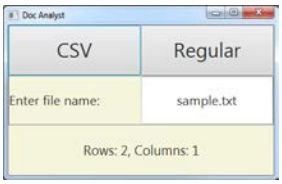
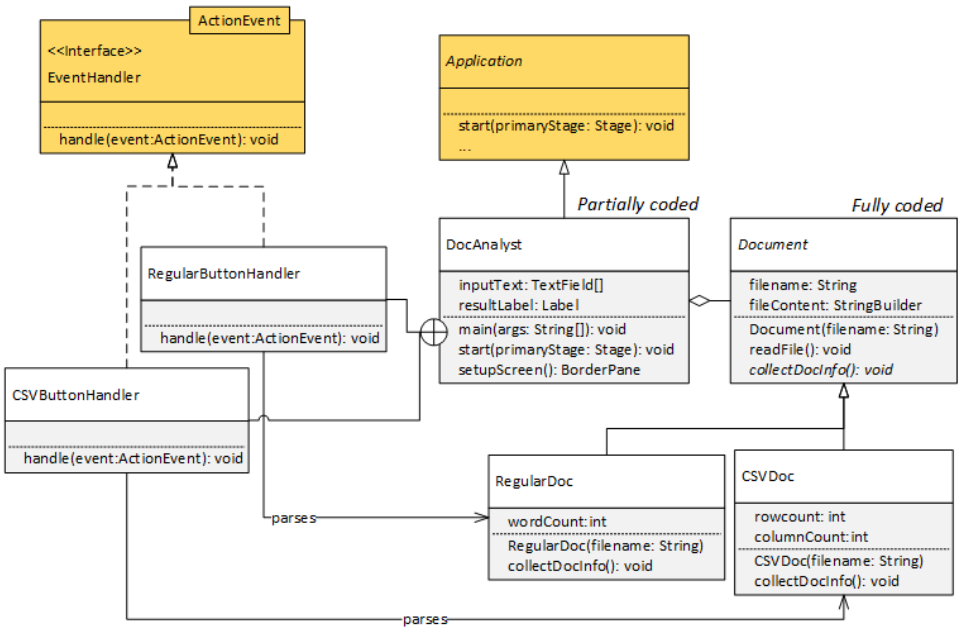


Figure 5: Enter regular file name and press CSV button

Solution Design: The program is launched from **DocAnalyst.java**. The abstract **Document** class has been provided to you that is fully coded. You need to do the following

1. Create **CSVDoc**, and **RegularDoc** classes that extend Document and override collectDocInfo(). In CSVDoc, the collectDocInfo () method sets the value of rowCount and columnCount. In RegularDoc, it sets wordCount. Most likely you should be able to use these files from Lab3.
2. Create handlers for two buttons in DocAnalyst.java: CSVButtonHandler and RegularButtonHandler. These two handlers do the following:
   i. create appropriate doc object using inputText name (Fig.1).
   ii. Invoke doc's read() method
   iii. Invoke doc's collectDocInfo()
   iv. Update resultLabel with doc's appropriate information.



Instructions:

1. Download DocAnalyst.java, Document.java, and TestDocs.java. You may already have books.csv, and sample.txt from Lab3. If not, download them as well.
2. Create a package named lab5 and import java files into it. Import the two text files in Labs project if you don't have them from Lab3
3. Create your classes as required and complete DocAnalyst.java. Write your name and Andrew id in all files
4. Test your code using TestDoc.java
5. Create a zip-folder named as your **Andrew id-lab5** that should have DocAnalyst.java, RegularDoc.java, and CSVDoc.java. Submit the zip folder on Canvas.

Rubric: GUI outputs: 6 points for four scenarios (1.5 points each); Test-cases: 3 points for 3 test-cases; 1 point for submission