River Crane

Professor Conrad

CIS 18B

5 June 2020

GUI Calculator Final Project Documentation

For the final project of this course, I have created a standard calculator making use of JavaFX and Java DB. The standard calculator created contains unique features to improve user experience. Following the requirements given, the project presents advanced string usage, files and streams, object serialization, databases, and concurrency. Starting with advanced string usage, the project takes advantage of StringBuilder to display any number values appearing on the calculator’s leftmost label. Functioning as a screen, a label is used from JavaFX’s scene builder to present numbers and calculated values based on user input. StringBuilder is initialized as eleven different strings containing numbers zero to nine, and an empty string for clearing the view. When any of the number buttons are pressed, the label named “display” will make use of setText and adding getText with the StringBuilder variables to present new values. Making use of fileIO in Java, the requirement of files and streams is successfully fulfilled. On this calculator, a button labeled help is displayed at the top right. This button fills a textArea containing information on how to use the application’s special features. When this button is pressed, a file named “Instruction.txt” will be read using scanner and making use of a while loop to read each line of text within the file. The textArea named instructionDisplay will then feature the text from the file read to guide users on the application. Once the help button is pressed it will be disabled to prevent unnecessary repetition of the instructional text. For object serialization, this project stores the most recent calculated answer as a byte stream. From the processOperands method, each calculation is decided by if else statements to determine which operand was used. After the operand is determined the variable named “operation”, it is assigned a new value which will tell the remainder of the method what to do for a switch case statement. Each case makes use of FileOutputStream and ObjectOutputStream to write the latest answer into the file within the project named “answers.txt”. This can be used as a possible backup feature in case the user has not utilized the save to history button, and wants to retrieve their last calculated answer from a previous session as a byte stream. The “Save to History” button introduces the part of the project which utilizes java DB. The database named “history” is used to store the latest value displayed on the main calculator label. The java class named “Queries” contains the code to connect to the database, and the code for the prepared statements. When adding to the database, the public void addHistory has a parameterized query, ensuring faulty information cannot be entered to crash the program or manipulate the data in unintended ways. The public void deleteHistory will clear the saved values within the database, although the identification values will still continue from where they left off as they are auto generated. Both addHistory and deleteHistory work in unison with the java class named “FXMLController” to properly update displays based on the buttons’ methods. For concurrency, the project makes use of a background thread to generate new numbers. When the “Lucky Number” button displayed above the database table view is pressed, a random value between 1 and 100000 is generated as an integer. The main application thread which updates the GUI is unaffected by the background thread, which demonstrates the efficiency of multithreading. These generated values will appear on the textArea below the help button, and can be used for inspiration as possible numbers to be used in the calculator. The textArea will have a scroll bar when the list becomes too large, so all generated vales alongside the help instructions can still be displayed. The only java class in this documentation not mentioned yet is “Display.java”, which holds the other programmer documentation kept in comments, the code for presenting the GUI application, and naming the window it opens in “Calculator”. Using all of these programming skills, a standard calculator with additional features for users’ quality of life has been created.