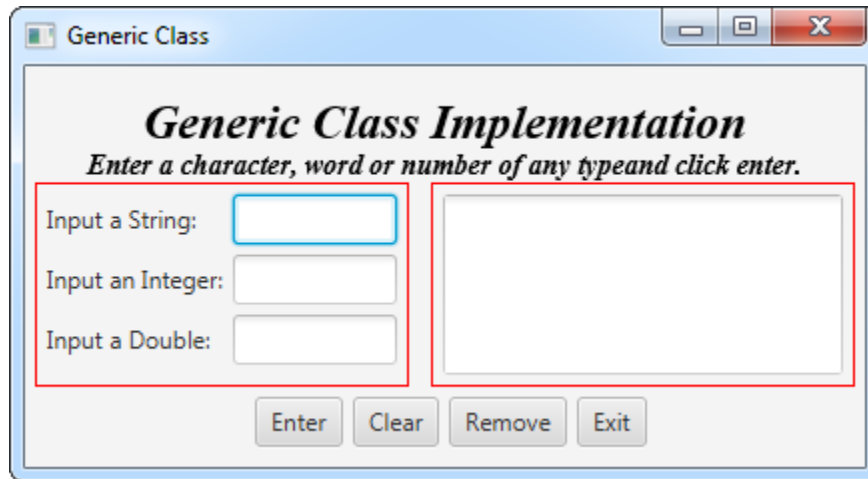


JavaFX Generics

Assignment

This week, we will create a simple program that implements a Generic class for adding and removing objects to an ArrayList.



Requirements

- Your GUI may not appear exactly as the sample above. However, it should have all of the components/nodes shown
- Implement a JavaFX main class and an external Generic (sub)class
- The results area (on right) is a TextArea control
- The Enter button sends whatever is located in the three TextFields (Integer, String or Double) to the Generic class for entry into an ArrayList
- Any combination of entries can be made
- Do not allow zeros for the Integer or Double entries
- At least one box must have a value, but two or all three may have values
- All values will be sent for entry into a generic ArrayList defined in a Generic external subclass. Example: `private ArrayList<E> list = new ArrayList<>();`
- Validate the Integer and Double entries as either inter of double respectively
- The Generic class must accept any object type (Integers, Strings and/or Doubles) without explicitly defining them by type. Example, the push method should accept object similar to the following: `public void push(E o) {`
- The TextArea should update each time the Enter button is clicked
- Separate entries with a single space
- The Clear button clears the TextField and the TextArea controls and requests focus back (places cursor in) to the first TextField. It should also empty the ArrayList
- The Remove button removes the last item entered into the ArrayList (see screen shots)
- The Exit button closes the program
- Use entry verification and alert popup messages to alert the user of:
 - No entry in the TextFields
 - An empty ArrayList
 - Incorrect data types
 - Zeros (0) entered into the Integer and/or Double fields

- Consider implementing an external Validation subclass for the alerts
- You are not required to incorporate an icon/picture in your GUI
- This assignment covers subjects from Chapter 1 -19

Hints

Your Generic class will have at least 5 methods:

- **Push or Add:** To add an element to the ArrayList
- **getSize:** To return the size of the ArrayList
- **getObj:** To return the current element value. This can be helpful when iterating through the ArrayList, typically when outputting elements to the TextArea from either entering or removing. The first data point to be entered will setText() while the others may appendText()
- **Pop or Remove:** Used to remove the last element from the ArrayList
- **Clear:** Used to empty or clear the contents of the ArrayList

Listing 19.1 in your text may be of **GREAT** assistance for this assignment

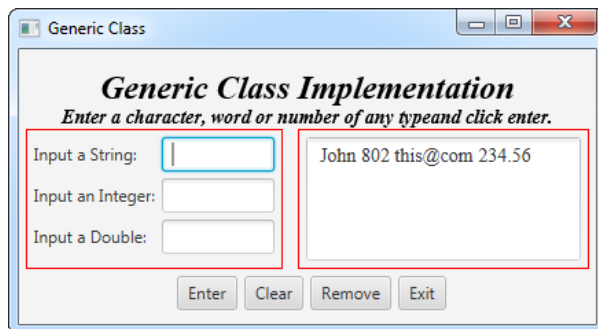
Expected Output

You have creative freedom with this assignment. Your result is not required to appear exactly as the sample above or screen shots below. However, it must contain all of the same components and include the functionality described above and below in the screen shot examples. Your GUI must also be visually appealing with consistent spacing/gaping.

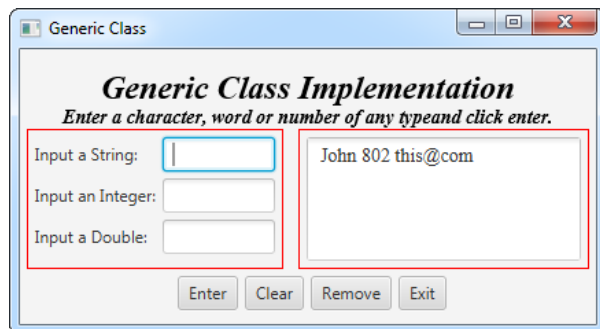
Deliverables

Please zip your program and submit the zip file by the due date listed in the requirements.

Screen Shots



Result after 4 entries (John, 802, this@com, 234.56)



Result after one click of Remove button (234.56 removed)