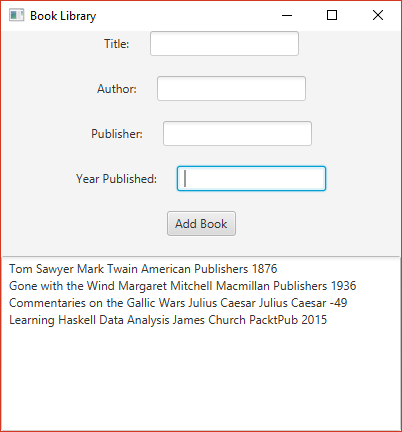
# Assignment 13: Write a Book Library Application

## Objectives

* Create a Book class.
* Use JavaFX to complete a program which maintains a list of book.
* Use an ArrayList to store a list of Book objects.
* Use StringBuilder to build a String based on all Book objects in an ArrayList.

## Motivation

In this program, you will be creating a Book Library application. Use the image provided for reference.



Book Library gui image

## Instructions

Name your project FirstnameLastnameAssignmentNumber

## The Book Class

The Book class will maintain four private final fields:

* String title
* String author
* String publisher
* int year

The Book class will require two methods, both of which require JavaDoc documentation.

### public Book Constructor

public Book(String title, String author, String publisher, int year)

This method will associate each of the four parameters with their respective fields. There are no preconditions. The post condition is that the object is created. If you make the constructor before the four fields, NetBean's Code Fix feature will create the four private final fields with a few mouse clicks.

### public String getDescription()

public String getDescription()

This method returns a String of the four fields by order of title, author, publisher, and year, with a space character separating each. There are no preconditions or postconditions or parameters. You do have to document the return value.

## The GUI

Write a program that will display four Label and TextField pairs, a Button, and a TextArea.

This will require the following:

* There will be a single Vertical Box which contains the following:
  + 5 Horizontal Boxes, all of which are centered using Pos.CENTER alignment.
    - A Title Box containing a Title Label and a Title TextField.
    - A Author Box containing an Author Label and an Author TextField.
    - A Publisher Box containing a Publisher Label and a Publisher TextField.
    - A Year Box containing a Year Label and a Year TextField.
    - A button which will perform all events named "Add Book".
  + A TextArea field. TextArea objects behave the same as TextField objects. Since you've already used TextField, I feel that's enough of a description.
* A title bar which says "Your Name's Book Library". If this does not include your name, you will lose points.
* In your start method, make sure that you include an ArrayList of type Book.

## Event Handler

The EventHandler for this assignment is rather involved. It should perform these steps, all of which were done inside of a large try block.

1. Create a Book object based on the four TextField fields. This will require that you parse the integer of the field that you used for the year a book was published.
2. Add the Book object to the end of your ArrayList. The method to add an element on to the end of an ArrayList is add.
3. Create a StringBuilder object initialized to an empty string.
4. Iterate over all of the Book objects in your ArrayList. For each object, append a call getDescription() to your StringBuilder object. Also, append a newline character.
5. Set the text of the TextArea object to the String being stored in your StringBuilder object. You'll need to call your object's toString() method to obtain the String object.
6. Set the text of each of the four TextField objects to an empty string. This clears the boxes.

All of this should be wrapped in a large try block. You should catch a NumberFormatException. When this happens, show a JOptionPane error message, which is identical to what you did in the last homework assignment.

## Notes

* The Scene size is 400 pixels wide and 400 pixels tall.
* In the image provided, every horizontal pane used the "Pos.CENTER" alignment. I did not apply an alignment to the vertical pane.
* All panes used 20 pixel spacing between children.
* My Book.java file was less than 35 lines of code.
* My Assignment13.java file was less than 130 lines of code.

## Documentation

Your source code must include the following documentation:

* Your name
* The class (CS 2070) and the section number (on ground is 08, online is W1).
* The date on which you turned in the assignment.
* A short description of the software. Usually a sentence or two is sufficient.

## Turning it in.

To turn in your application, find the folder containing your entire project (not the folder with the "java" file), zip it up, and turn it in.