

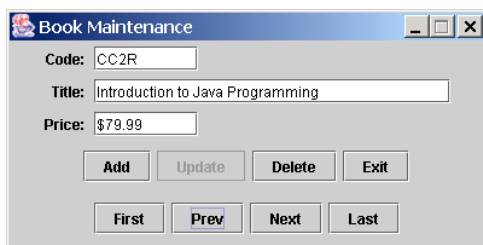
## Lab 4

### **Description of the Problem**

Write a GUI program that lets user maintain records in a file or database that contains the data for Book objects.

### **Specifications**

- When the program is started, the data are automatically loaded from a file or database. The first record is initially displayed. The Update button is disabled until new text is entered in any of the text fields, as shown below:



- To navigate through the book records, the user can click on the one of the buttons in the bottom row: the First button displays the first record in the file or database; the Prev button displays the previous record; the Next button displays the next record; and the Last button displays the last record. Whenever a new record is displayed, the Update button is disabled.
- To modify the data for a book, user makes a change to the title or price fields. At that point, the Update button is enabled and the Add, Delete, and navigation buttons are disabled. Then, to save the changes to the file or database, the user clicks on the Update button.

Hint: to enable or disable buttons, you can write a helper function, like this:

```
private void enableButtons(boolean flag1) {  
    boolean flag2 = false;  
    if (!flag1)  
        flag2 = true;  
    updateButton.setEnabled(flag2);  
    addButton.setEnabled(flag1);  
    deleteButton.setEnabled(flag1);  
    firstButton.setEnabled(flag1);  
    nextButton.setEnabled(flag1);  
    prevButton.setEnabled(flag1);  
    lastButton.setEnabled(flag1);  
}
```

- To add a record to the file or database, the user clicks on the Add button. This clears the text fields, enables the Update button, and disables the Add, Delete, and navigation buttons. Then, the user can enter the code, title, and price (with or without

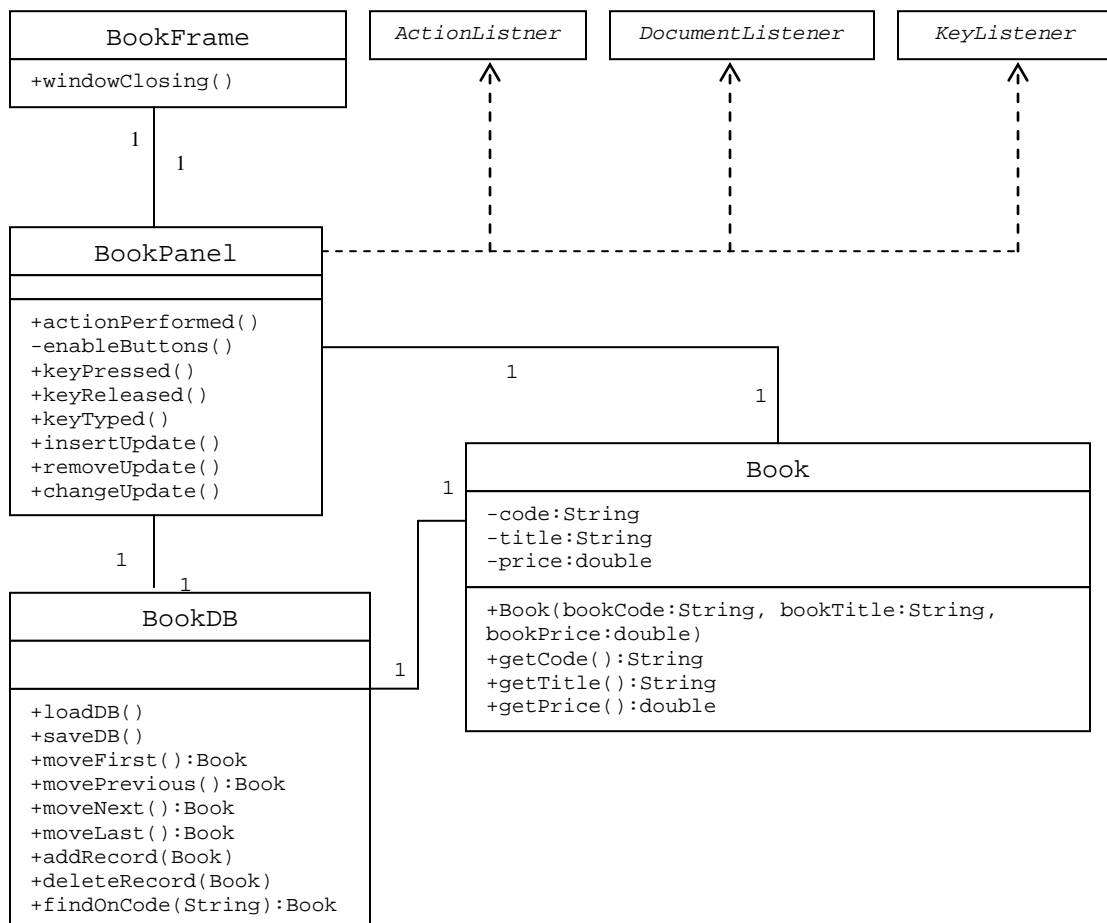
dollar sign) for a new book and click on the Update button to save the new record to the file or database. This record will be added at the end of the file or database.

- To delete a record, the user navigates to that record and clicks on the Delete button. After it deletes the record from the file or database, the program displays the data for the next record (or the new last record if the deleted record was the last record).
- To cancel any add or update operation, the user can press Esc key when the focus is in one of the text fields

Hint: implement KeyListener:

```
public void keyPressed(KeyEvent e) {
    int keyCode = e.getKeyCode();
    if (keyCode == KeyEvent.VK_ESCAPE) {
        // codeField.requestFocus();
        // performBookDisplay();
        // enableButtons(true);
    }
}
```

- To exit from the program at any time, the user can click on the Exit button. This will cancel any change, addition, or deletion that's in progress.
- Design your application based on Object-oriented programming principle. Your GUI interface should be separated from the operational logic. Changing the operational logic will not affect your GUI interface. Here is a sample design in UML:



## ***Requirements***

- Your program should follow the coding guideline.
- Turn in the following
  1. Source code
  2. Sample interactions that demonstrate all of the features.

## ***Grading criteria***

- Design (is OO design principle followed?)
- Correctness (does your program work?)
- Clarity (simple, no redundant classes/methods/variables, etc.)
- Completeness (all required parts turned in?)
- Documentation (follow javadoc and coding guideline)
- Timeliness (on-time?)