Code/Design Before & After Refactoring:

- Long classes and methods →
    - addressed by breaking big classes up to improve and maintiant Single Responsibility DP
    - extracting long methods into components in the app
    - prevent repeated code by utilizing the new extracted method
- Duplicate code and methods across different classes →
    - addressed by extracting code found in UI in a common component
- Long parameters being passed from one class to another (ex. Found in BookModel) →
    - addressed and reafactored utilize the MVP architecture correctly, introducing parameter object, or being able to extract the class depending on the use of the method and the values being passed.
    - For example for book information, instead the object clicked on is passed instead of each value of the book clicked.
- Some variables were not declared final (detected by automatic tool) →
    - addressed by reviewing the use of the instance variable and if the recommendation could be applied
- Other refactoring tasks:
    - Initital App Load-Time - from 10s to <2s using parallel streams
    - refactor application to be single page application and how to make components reusable
    - Conversion from multiple JFrames to JPanels
    - include regex for input validation