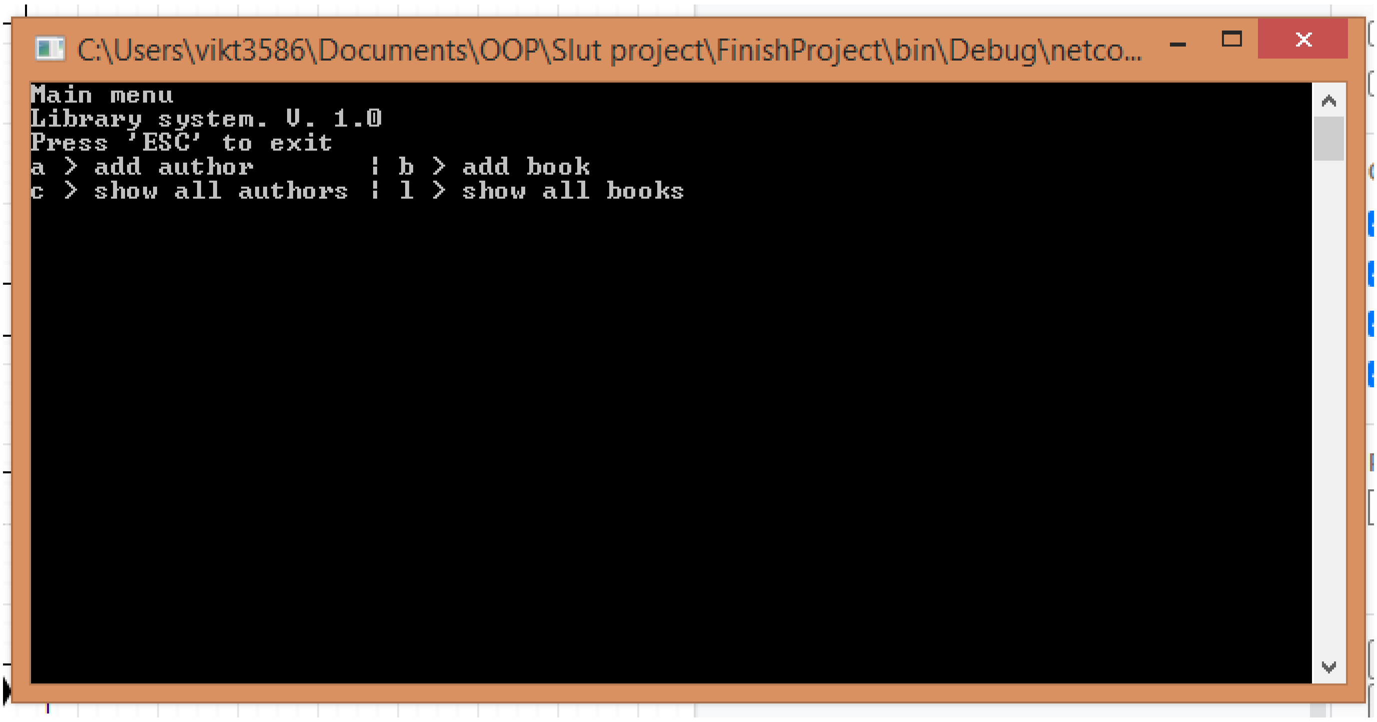
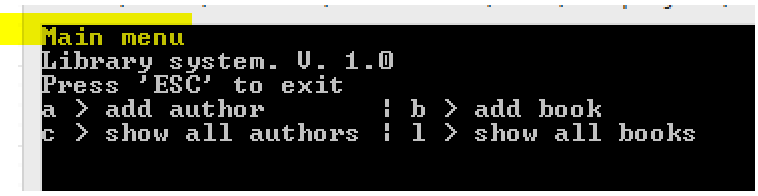
How to use

This is the main stage of the program


C:\Users\vikt3586\Documents\OOP\Slut project\FinishProject\bin\Debug\netco...

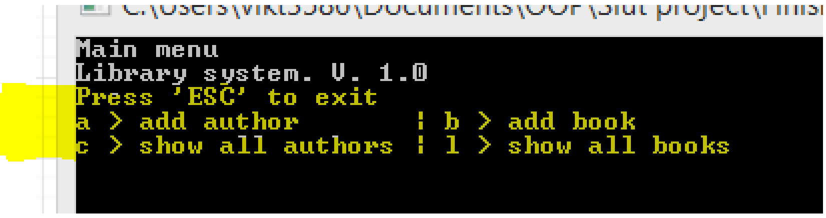
```
Main menu
Library system. V. 1.0
Press 'ESC' to exit
a > add author       ¦ b > add book
c > show all authors ¦ l > show all books
```

The first line tells you which stage in the program you are on



```
Main menu
Library system. V. 1.0
Press 'ESC' to exit
a > add author       ¦ b > add book
c > show all authors ¦ l > show all books
```

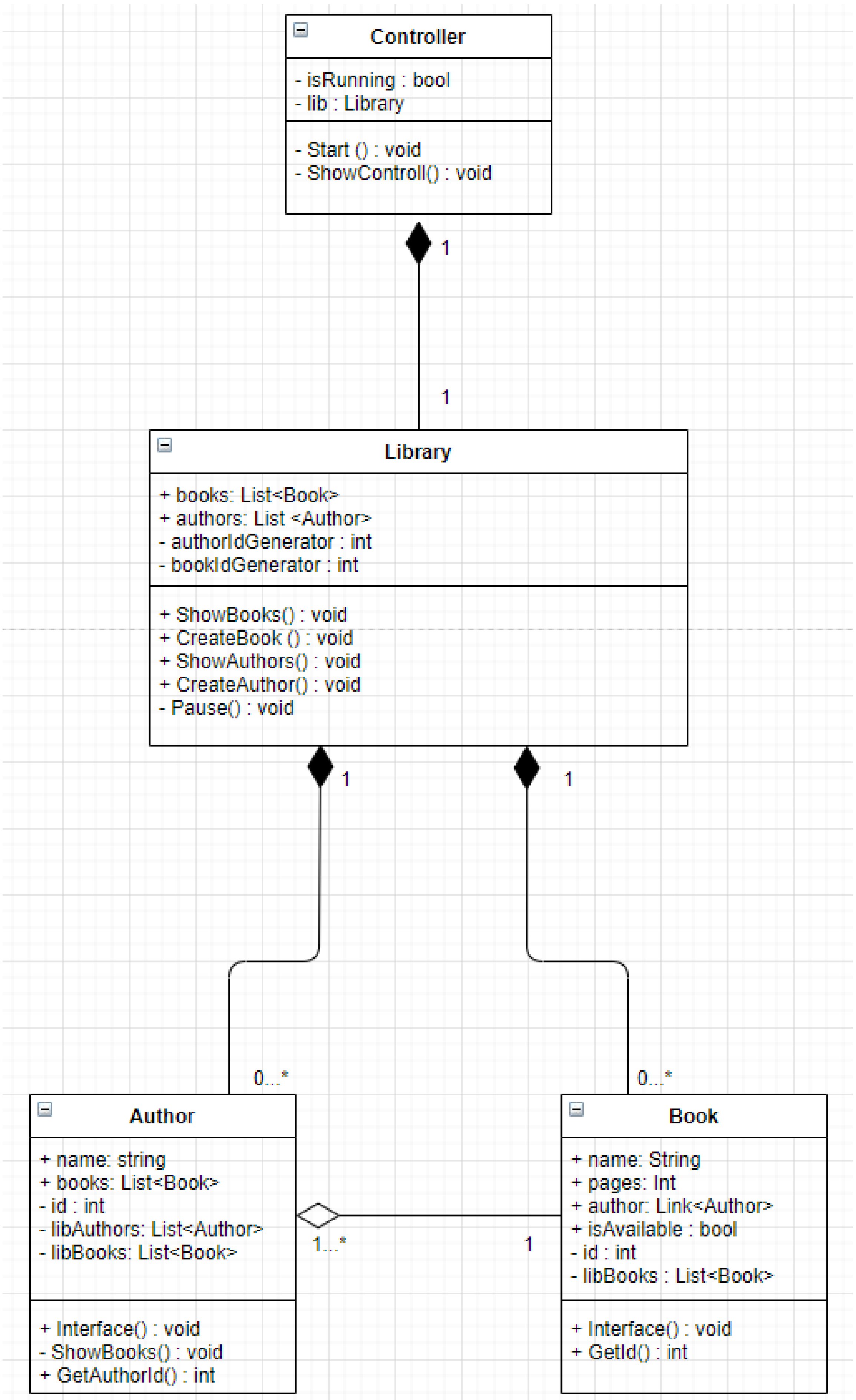
This part is responsible for controlling the program in a certain scene



```
Main menu
Library system. V. 1.0
Press 'ESC' to exit
a > add author       ¦ b > add book
c > show all authors ¦ l > show all books
```

The program is very easy to use and very easy to get used to. The whole program is built on the principle of levels. At each level, the program has certain buttons to control and also some functionality

Application UML diagram



Description

Program for library management.

In this program you can register a writer and also the books he wrote and then you can mark whether this book is currently available or not.

In the program I use 4 classes. The most important class is the controller. The program loop begins with this class. When an instance of the controller class is created, the main program loop (start method) is started in the constructor.

Library this class performs 2 functions. It is used as a repository for created instances of the author and book classes and contains the logic of the program itself (methods to show books / authors and also create a book / author)

When creating an author, the program checks whether an instance of the class with that name has already been created, and if created, the program writes an error. When creating a book there, the program also checks whether the created author is specified when creating the book and if it is missing, the program creates an instance of the class and adds a pointer to the book to the author. If it is created the program uses the existing instance and adds a pointer.

2 classes (author, book) to describe and save as a certain unit of the author and the book. These are almost identical classes and their main purpose is to represent a certain object as a unit. Also these classes have an interface method that allows the user to interact with the object. For example, in a book it is possible to take it and return it to the library and delete it from the library. The author will open the interface to a specific book or delete the author. When you delete an author, all his books in the library are also deleted

Each book and author has an incremental mechanism for id. Each new book or author will have an id greater by 1.

The program was written as a project for an educational institution.

Viktor Viskov