Library Records.
Pavel Arefyev, Metropolia University of Applied Sciences, Smart IoT Systems major.
User Manual.

Start of the program

The user is asked to upload the file. Type **yes** to choose the file on your system.

Main menu

The menu is printed for the user. Type an option number in order to choose the action:

1) Initialize the storage with new records, clear the current record.
2) Add a new book to the current record.
3) Borrow a book from the current record.
4) Return a book to the current record.
5) Delete a book from the current record.
6) Print the current record.
7) Change sort parameter for option number 7.
8) Save the current record to a file specified by user.
9) Upload the record from the file, clear the current record.
0) Exit the program.

Sort menu

Sort option menu provides three choices. Type an option number in order to choose an action:

1) Sort by availability (default)
2) Sort by ID (lowest ID to highest)
3) Sort by title (alphabetic order)

File handling

User can save and open previously saved files through the program's interface. However, if necessary, user may create a file for further reading himself. Each line of the file should be in the following format:

ID;Title;FLAG;owner;Day Month

FLAG indicates book's availability: A - available, B - borrowed. Examples of correct files:

2131;The Lord of The Rings;A;
22;Harry Potter and the Goblet of Fire;B;John Brown;31 January

Note, that the record will take the highest ID counter mentioned in the file, that will be further used for assignments.

End of the program

User is asked to save the file. Type **yes** to choose the file name to be saved on your system.

Technical design

Class Library

Class Library has functions to operate with the storage, as well as save and upload he file. Has a subcomponent vector of Book class object.

vector<Book> storage: vector of books of the library
int idCounter: counter that is used to dynamically assign an ID to the new book. idCounter is incremented each time book added to the record. When uploading a file, idCounter takes the highest id of the book in the file.
int sortOption: used to define the sorting algorithm in printBookRecord function. By default, equals to 1.

sortOption 1 = Sort by availability
sortOption 2 = Sort by ID
sortOption 3 = Sort by title

Class Book

Class Book has functions to operate with individual books in the storage, such as inputting/getting/setting values. Operator>> and operator<< are overloaded for convenient file handling. Has a subcomponent object of class Date.

string title: title of the book
int id: id of the book, assigned automatically
string owner: owner of the borrowed book
Date date: return deadline
bool borrowed: state of the book (borrowed/available) -> (true/false)

String inputs (title, owner) are limited by the number of characters using a parameter maxLength (by default, equals to 20) in functions **userInput**, **ownerInput**. In the main.cpp, value passed is defined as constant and equals to 35.

Class Date

Class Date makes the interface more intuitive. Operator>> and operator<<, are overloaded for user input/output and more convenient file handling.

There are plans to add functionality and make more use of Date class.