PROJECT OF FINAL SEMESTER 2024 –
ALGORITHMS AND PROGRAMMING: SYSTEM OF LIBRARY
[ENGLISH VERSION]

Professor: Holger Espinola Rivera

Working in groups of 4 students, create a system of library in C++ which have the next structs:

INDICATIONS

1. Implement a system of library in programming language C++

2. The system need have ordered structure with files .h and .cpp

3. The system need have folder called **data** which will contain the .txt files book.txt, client.txt, order.txt and order_detail.txt

4. The operations and data manipulation released in terminal need to reflect in changes in the data stored in .txt files

5. The system need to publish in Github of coordinator of group. The professor will revise the source code and functionality

6. In the day **17.12.2024** each group will expose the source code and the functionality of the system. Each member will expose 1 of the components of system (Book, Client, Order and OrderDetail). We have 4 components and 4 members, 1 for each one. All students need stay present in the exam.

7. The evaluation have 2 scores: 1 score individual depending of performance of student and 1 score for group depending of quality of system functionality.

8. In GitHub repository you can take as reference:

- link of repo:

https://github.com/HoltechHard/Algorithms-5130203/tree/master

- week08_project: system of orders of products

- week09/program03.cpp: operations of read() and write() using files and struct

- week09_dz: reference of professor Glazunov's files about main components of class BOOK

DATA STRUCTURES

Using variables, vectors, structs and enumerations to represent the next entities of the library system:

1. Book:

Attributes

- code: auto-incremental integer number

- author: vector of char, with maximum 40 letters

- title: vector of char, with maximum 40 letters

- stock: integer number related with total copies of the book

- price: real number between 100 – 50,000 rubles

- year: integer number not lower than 2000

- category: integer number, the code of category

- Category: defined by an enumeration which can be “fantasy”, “fantastic”, “roman” and “history”

2. Client:

Attributes

- code: auto-incremental integer number

- name: vector of char with maximum 40 letters

- age: integer value non-negative

- salary: real number between 20,000 and 500,000 rubles

3. Order:

Attributes

- code: auto-incremental integer number

- Client: instance of data from Client which request the order

- details: vector of Order Details

- number of details: total of order details generated by client in some specific order

- total: real number, which compute the sum of all accumulate money for each order detail

4. Order Detail:

Attributes

- book: instance of book

- quantity: how much copies are bought by client. This value can’t exceed the stock of the book in the library

- subtotal: the accumulated money to pay by the detail of order = price * quantity

OPERATIONS

1. Menu

- Define 1 main menu to able access Book, Client and Order.

- Define 1 submenu for Book, 1 for Client and 1 for Order

2. Insert

- in terminal input by keyboard each attribute of your structure and store the corresponding information in .txt file.

- It is necessary manage 4 .txt files (book.txt, client.txt, order.txt and details.txt) inside of folder data which will contain all the information in persistence layer.

- Book, Client, Order and Order Detail need have insert operation

- To store the information in .txt file, use the function **fprintf()** from library **<cstdio>**

- Use dynamic array of structs to manage the insertion of new instances of entities. You can define a maximum number of elements in this structure, but implement some functionality to if the memory capacity surpasses, the system will allocate additional memory space defined by the maximum number of elements to continue the process of insertion

3. Read

- the system need read the information from .txt file and show in terminal organized in format of table, all attributes with their respective data.

- Book, Client and Order with their details need have read operation

- to read the information from .txt file, use the function **fscanf()** from library **<cstdio>**

4. Update

- Book and Client need have the update operation

- the system need requite input the name to search, and after execute the operation of update. The changes in update need to see in the .txt file

5. Delete

- Book, Client and Order Details need have the delete operation

- the system can drop all the information contained in the .txt file corresponded by the entity

6. Sort

- Implement just for books

- Provide to user the ability to display a list of books in ascendent order depending of the criteria (defined by the enumeration Sort Criteria)

- Criteria: defined by an enumeration of 5 possible criteria of ordering: by “author”, “title”, “price”, “year” and “category”

7. Search

- Implement just for books

- Search books by criteria (the same enumeration defined in point 6-Sort). The user will choose the criteria and depending of this, will search the book. Depending of criteria, the system will search and show a list of books which coincide with this criterion.

8. Exit

- If you stay in submenu, need to return to main menu

- If you stay in the main menu, close the program

- The system need ask between each transition of selected operations of the menu if the user desire continue or not