Да се напше клас ***Person*** със следните полета :

- име – низ с произволна дължина;

- ЕГН – цяло число;

- пол – може да бъде един символ, символен низ или стойност от изброим тип.

За този клас да се напишат подходящи конструктори, деструктор, гетъри и сетъри за полетата, както и да се предефинира оператор = .

Да се напише клас ***LibraryBook*** със селдните полета:

- име – низ с произволна дължина;

- автор – обект от тип *Person*;

- [ISBN](https://bg.wikipedia.org/wiki/Международен_стандартен_номер_на_книга) – масив от символи;

- жанр - може да бъде символен низ с произволна дължина или стойност от изброим тип;

- препоръчителна цена – рационално число;

- инндекс на текущо наелия я потребител – цяло число;

За този клас да се напишат подходящи конструктори, деструктор, гетъри и сетъри за полетата, както и да се предефинира оператор = .

Да се напише клас ***Library*** със следните полета:

- книги – масив от *Book* с произволна дължина;

- текущ брой книги – положително цяло число;

- текущ капацитет на масива с книги – положително цяло число;

- потребители – масив от *Person* с произволна дължина;

- текущ брой потребители – положително цяло число;

- текущ капацитет на масива с потребители – положително цяло число;

- потребителска такса – рационално число;

-текущ финасов баланс - рационално число.

За този клас да се напишат подходящи конструктори, деструктор, гетъри и сетъри за полетата, както и да се предефинира оператор = .

Да се реализират и следните методи:

- *void addBook(const LibraryBook& book)* – добавя нова книга към масива от книги, като от текущия баланс на библиотеката се изважда цената на книгата. Ако масиват с книги е достигнал своя капацитет, то той се преоразмерява – заделя се нов масив с по-голям капацитет, там се копират обектите от стария масив, след което се добавя новия обект.

- *void addUser(cosnt Person& person)* – добавя нов потребител към масива от потребители, като към текущия баланс на библиотеката прибавя стойността на потребителската такса. Ако масивът с потребители е достигнал своя капацитет, то той се преоразмерява.

-*void rentBook(const Person& user, const LibraryBook& book)* – репрезентира наемане на книгата book от потребителя user. За целта първо трябва да се “намери” book в масива от книги – търсим книгата с ISBN, еднакъв с този на book (това означава, че двата обекта представляват една и също книга), а след това като неин текущ наемател отбелязваме индекса от масива с потребители, на който се намира обект с EГН, еднакъв в този на user. Ако в масивите с книги или потребители не съществуват дадените книга или потребител, не се извършва никаква операция.

- *void returnBook(const LibraryBook& book)* – връщане на книга. Книгата се “намира” в масива с книги и в полето *инндекс на текущо наелия я потребител* се записва стойност -1. Ако в масива с книги не съществува такава книга(книга с ISBN, еднакъв с този на book), не се извършва никаква операция.

- *void registerLostBook(const LibaryBook &book)* – регистрира се загубена книга. Тово означава, че тя трябва да се намери и премахне от масива с книги, като към баланса на библиотеката се прибави цената на книгата, тъй като считаме, че загубилият я потребител е длъжен да възстанови стойността й на библиотеката. Ако в библиотеката няма такава книга, не се извършва никаква операция.

- *void printInfoForAllBooksRentedByUser(const Person& user)* – отпечатва информация за всички книги, текущо наети от дадения потребител, ако има такива.