**Ministerul Educaţiei și Cercetării al Republicii Moldova**

**Colegiul Universitatii Tehnice a Moldovei**

**RAPORT**

Lucrare de laborator

Bazele OOP

A efectuat: Dabija Cristian

A verificat:                                               Cătălin Coșeru

Chișinău 2024

***Lucrare de laborator***

***Condiția lucrării:***

***Tema lucrarii*:** Bazele OOP

A simple laboratory work focused on the basics of Object-Oriented Programming (OOP). In this lab, students will create a basic program to manage a library of books using Python. This lab covers the fundamental concepts of classes, objects, and basic inheritance.

***Lab Title:*** Simple Library Management System

***Objective:*** To create a simple Library Management System using Object-Oriented Programming concepts.

***Sarcina lucrării:***

1. Implement two classes: Book and Library.
2. The Book class should have attributes like title, author, and ISBN.
3. The Library class should have a list to store books and methods to add a book, remove a book, and display all books in the library.
4. The code should be uploaded to your github account, if not then make sure you’ll create one.
5. Your repository should be public.
6. You can use gitUI, fork or other programs instead of git command line.

***Textul programului în limbajul C++:***

***#include <iostream>***

***#include <vector>***

***#include <algorithm>***

***#include <string>***

***using namespace std;***

***class Book {***

***public:***

***Book(string title, string author, string isbn):title(title), author(author), isbn(isbn) {}***

***string getTitle() const { return title; }***

***string getAuthor() const { return author; }***

***string getISBN() const { return isbn; }***

***private:***

***string title;***

***string author;***

***string isbn;***

***};***

***class Library {***

***public:***

***void addBook(const Book& book) {***

***books.push\_back(book);***

***}***

***void removeBook(const string& isbn) {***

***auto it =remove\_if(books.begin(), books.end(),***

***[&isbn](const Book& book) { return book.getISBN() == isbn; });***

***if (it != books.end()) {***

***books.erase(it, books.end());***

***cout << "Book with ISBN " << isbn << " removed from the library.\n";***

***} else {***

***cout << "Book with ISBN " << isbn << " not found in the library.\n";***

***}***

***}***

***void displayBooks() const {***

***if (books.empty()) {***

***cout << "Library is empty.\n";***

***} else {***

***cout << "Books in the library:\n";***

***for (const auto& book : books) {***

***cout << "Title: " << book.getTitle() << ", Author: " << book.getAuthor() << ", ISBN: " << book.getISBN() << "\n";***

***}***

***}***

***}***

***private:***

***vector<Book> books;***

***};***

***int main() {***

***Library library;***

***library.addBook(Book("The Catcher in the Rye", "J.D. Salinger", "978-0-316-76948-0"));***

***library.addBook(Book("To Kill a Mockingbird", "Harper Lee", "978-0-06-112008-4"));***

***library.addBook(Book("1984", "George Orwell", "978-0-452-28423-4"));***

***library.displayBooks();***

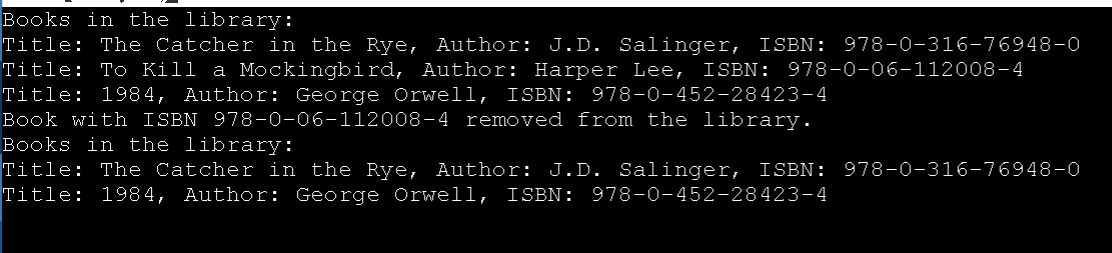
***library.removeBook("978-0-06-112008-4");***

***library.displayBooks();***

***return 0;***

***}***

***Rezultatele obţinute în urma execuţiei:***

******

**Concluzie:**

*In conclusion, the provided C++ code demonstrates a basic implementation of a Library system using two classes: Book and Library. The Book class encapsulates information about a book, including its title, author, and ISBN. The Library class manages a collection of books and provides functionality to add, remove, and display books.*

*The main function initializes a Library object, adds several books to it, displays the initial list of books, removes one book based on its ISBN, and displays the updated list of books.*

*However, it's important to note that this code is a basic example, and in a real-world scenario, additional features, error handling, and user interactions would likely be necessary. Moreover, the use of “using namespace std;” is present, but in larger projects, it is generally recommended to avoid this using directive and specify the namespace where needed to prevent naming conflicts.*