**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ДВНЗ «УНІВЕРСИТЕТ БАНКІВСЬКОЇ СПРАВИ»**

**ІНСТИТУТ БАНКІВСЬКИХ ТЕХНОЛОГІЙ ТА БІЗНЕСУ**

**КАФЕДРА КІБЕРБЕЗПЕКИ ТА СОЦІАЛЬНИХ НАУК**

**КУРСОВА РОБОТА**

з дисципліни

**«ОБ’ЄКТНО ОРІЄНТОВНЕ ПРОГРАМУВАННЯ»**

(назва дисципліни)

на тему:  **Розробка особистої віртуальної бібліотеки**

Студента 2 курсу 203 групи

спеціальності «Кібербезпека»

Хомічак Микола Остапович

(прізвище та ініціали)

Керівник

Доцент кафедри

(посада, вчене звання, науковий ступінь,

Гордєєв О.О.

прізвище та ініціали)

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_

Оцінка: ECTS \_\_\_\_

Члени комісії

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

Київ - 2019

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ДВНЗ «УНІВЕРСИТЕТ БАНКІВСЬКОЇ СПРАВИ»**

**ІНСТИТУТ БАНКІВСЬКИХ ТЕХНОЛОГІЙ ТА БІЗНЕСУ**

**КАФЕДРА КІБЕРБЕЗПЕКИ ТА СОЦІАЛЬНИХ НАУК**

Спеціальність: «Кібербезпека»

Курс 2 Група 203-Кб Семестр 1

**ЗАВДАННЯ**

**На курсову роботу студента/студентки**

Хомічака Миколи Остаповича

(прізвище, ім’я, по батькові)

1. Тема курсової роботи:

Розробка особистої віртуальної бібліотеки

2. Термін здачі студентом закінченої роботи 19.12.2019

3. Вхідні дані до (роботи) інформаційно-аналітичні інтернет джерела, зразки рішень практичних завдань та задач, нормативно-правова база, щодо оформлення технічної документації програмних продуктів, початкові значення для обробки.

4. Зміст пояснювальної записки (перелік питань, які належить розробити). Вступ. Формулювання вимог до програми. Проектування та практична реалізація розв’язку поставленої задачі. Тестування програми і результати її виконання. Висновки.

5. Перелік графічного матеріалу (з точним зазначенням обов`язкових креслень) Таблиці та схеми представлено в роботі. Презентація додається.

6. Дата видачі завдання «10» жовтня 2019 р.

Зміст

[Вступ 1](#_Toc28122707)

[Розділ 1. Формулювання вимог до програми 2](#_Toc28122708)

[1.1. Аналіз технічного завдання 2](#_Toc28122709)

[1.2. Обґрунтування алгоритму і структури програми 2](#_Toc28122710)

[Розділ 2. Проектування та практична реалізація розв’язку поставленої задачі 4](#_Toc28122711)

[2.1. Оцінка коректності алгоритму та аналіз алгоритму на часову та просторову ефективність 4](#_Toc28122712)

[2.2. Розробка специфікації 4](#_Toc28122713)

[2.3. Опис програми (обґрунтування вибору мови програмування, опис основних процедур та функцій). 4](#_Toc28122714)

[2.4. Опис змінних, та їх ідентифікаторів, що використовувались у програмі 5](#_Toc28122715)

[2.5. Створення об'єктів і розробка головної програми 5](#_Toc28122716)

[2.6. Опис файлів даних та інтерфейсу програми 6](#_Toc28122717)

[Розділ 3. Тестування програми і результати її виконання 7](#_Toc28122718)

[3.1. Інструкція користувача 7](#_Toc28122719)

[Висновки 9](#_Toc28122720)

[Список використаних джерел 10](#_Toc28122721)

[Додатки 11](#_Toc28122722)

[Додаток А. Лістинг програми 11](#_Toc28122723)

Вступ

C++ − мова програмування високого рівня з підтримкою кількох парадигм програмування: об'єктно-орієнтованої, узагальненої та процедурної. Розроблена Б'ярном Страуструпом в AT&T Bell Laboratories 1979 року та початково отримала назву «Сі з класами». Згодом Страуструп перейменував мову на C++ у 1983 р.

Після появи інтернету , з’явилась потреба оцифровки всієї можливої інформації. Найбільше інформації містилось у книжках, саме тому Їх в першу чергу намагались оцифрувати і перенести у світову мережу. Тому з’явилась потреба у створенні програм які б замінили звичайні бібліотеки. Оскільки мова програмування C++ показала свою ефективність і надійність, то на її базі і можна створити програму Домашньої бібліотеки.

Метою цієї курсової роботи є закріплення уже наявних навичок та здобуття нових в області програмування на мові C++ та написання своєї першої програми.

Розділ 1. Формулювання вимог до програми

**1.1. Аналіз технічного завдання**

В основі завдання лежить розробка повноцінної програми Домашньої бібліотеки мовою програмування «С++». Для цього потрібно побудувати алгоритм роботи прогарами, внести список книг, авторів, жанрів та розробити команди, якими буде керувати користувач.

**1.2. Обґрунтування алгоритму і структури програми**

Написання програми полягатиме в створені масиву «library» (бібліотека), масиву «book» та 3 масивів з жанрами книжок. Також буде створено функції для перегляду , зміни та оброблення наявних у програмі даних та показ всієї доступної інформації користувачу. Швидкість виводу рядків буде регулюватись окремою функцією, яка у своїй основі використовує бібліотеку «chrono». Також будуть використанні бібліотеки «string» та “thread” для роботи з рядками і підтримки потоків відповідно. Потоки вказують деяким елементами коли працювати асинхронно , а коли одночасно. Також була розроблена блок-схема алгоритму роботи програми «рис. 1.1.»

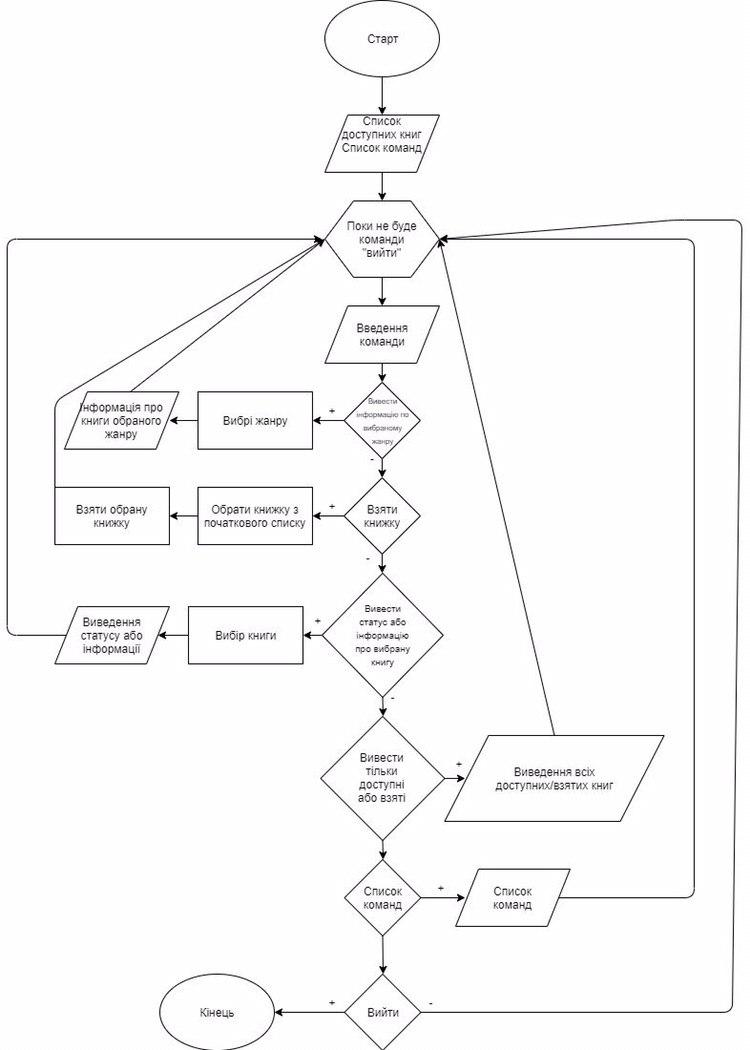


Рисунок 1.1. Блок-схема роботи коду програми

Розділ 2. Проектування та практична реалізація розв’язку поставленої задачі

**2.1. Оцінка коректності алгоритму та аналіз алгоритму на часову та просторову ефективність**

Алгоритм цілком і повністю задовольняє всі вимоги програми «2.2.». В ньому передбачені всі можливості, які є в повноцінній грі «Тетріс», наприклад: можливості повороту фігур, виведення інформації про наступну фігуру та навіть функція паузи в самій грі.

Щодо просторової та часової ефективності, то алгоритм не є ресурсозатратним, оскільки це більш-менш примітивна консольна програма, отже побудова алгоритму компілятором не займає багато часу.

**2.2. Розробка специфікації**

Програма повинна мати певний перелік книг, що містяться в домашній бібліотеці. Щодо функцій самої бібліотеки, то повинні бути реалізовані функції виводу всієї бібліотеки та виводу лише доступних на даний момент часу книг.

**2.3. Опис програми (обґрунтування вибору мови програмування, опис основних процедур та функцій).**

Функція (в програмуванні) – це фрагмент коду або алгоритм, реалізований на якійсь мові програмування, з метою виконання певної послідовності операцій. Отже, функції дозволяють зробити програму модульною, тобто розділити програму на кілька маленьких підпрограм (функцій), які в сукупності виконують поставлене завдання. Ще один величезний плюс функцій в тому, що їх можна багаторазово використовувати. Дана можливість дозволяє багаторазово використовувати один раз написаний код, що в свою чергу, набагато скорочує обсяг коду програми!

Крім того, що в С передбачено оголошення своїх функцій, також можна скористатися функціями визначеними в стандартних заголовних файлах мови програмування С++. Щоб скористатися функцією, визначеної в заголовному файлі, потрібно його підключити.

* print\_genres\_ids() - вивести по жанрам
* print\_chosen\_status()- отримати інформацію про статус обраної книги
* print\_chosen\_info() - отримати інформацію про обрану книгу
* take() - обрати книгу
* check\_availability() - перевірити на наявність

**2.4. Опис змінних, та їх ідентифікаторів, що використовувались у програмі**

Змінна — це абстрактна назва комірки чи декількох комірок пам'яті. Кожна змінна має шість атрибутів: ім'я, адресу, значення, тип, область видимості і час життя. Адреса змінної — це адреса комірки пам'яті, зв'язаної з даною змінною. Якщо змінна займає декілька осередків, її адресою вважається адреса першого осередку. значення змінної — це двійкове число, записане в комірках пам'яті, зв'язаних з даною змінною. Осередок являє собою окрему одиницю пам'яті, що має адресу. Як правило, у більшості сучасних комп'ютерів розмір осередку дорівнює одному байту, що складається з восьми бітів.

Тип змінної визначає діапазон значень, що вона може набувати, і набір операцій, що до неї можна застосовувати. Поняття типу в мові С++ є принципово важливим, тому ми розглянемо його окремо, а поки зупинимося на концепції зв'язування, що також представляє дуже великий інтерес.

Під час написання програми бібліотеки було використано змінну типу “unsigned int”

Використано зміну “id” - відповідає за порядковий номер книжку у списку.

**2.5. Створення об'єктів і розробка головної програми**

Класи і об'єкти в С є основними концепціями об'єктно-орієнтованого програмування – ООП. Об'єктно-орієнтоване програмування – розширення структурного програмування, в якому основними концепціями є поняття класів і об'єктів. Основна відмінність мови програмування С++ від С полягає в тому, що в С немає класів, а отже мова С не підтримує ООП, на відміну від С++ .

Оголошення класу починається з зарезервованого ключового слова клас, після якого пишеться ім'я класу.

У тілі класу оголошуються три мітки специфікації доступу: private, public і protected

Серед використаних класів можна виділити клас бібліотеки, та, оскільки бібліотека не може існувати без книг, то всередині цього класу був розроблений ще клас книги. В самі програмі ведеться робота з єдиним об’єктом класу бібліотеки.

**2.6. Опис файлів даних та інтерфейсу програми**

Весь код до даної програми був написаний на одному файлі формату «cpp», що має назву «Source».

Інтерфейс складається зі стартового меню, на якому вказано весь список доступних команд. Одразу після його виводу програма пропонує користувачу ввести команду, після чого програма залежно від команди виконує певні дії.

Розділ 3. Тестування програми і результати її виконання

**3.1. Інструкція користувача**

Для даної програми була розроблена інструкція для користувача «табл. 3.1.», в якій вказано команди і відповідну функцію, яку команда виконує.

|  |  |
| --- | --- |
| list | Перегляд списку команд |
| рrint\_genres | Вивід доступних жанрів |
| take | Вибрати книгу |
| print\_chosen\_status | Статус обраної книги |
| print\_chosen\_info" | Інформація про обрану книгу |
| print\_only\_taken | Дізнатись які книги обрані |
| print\_only\_availeble | Дізнатися наявні книги |
| quit | Кінець роботи програми |

Таблиця 3.1. Керування

Одразу при запуску програма виводить головне меню «рис. 3.1.», на якому програма пропонує користувачу вибрати команду та виводить список доступних команд.

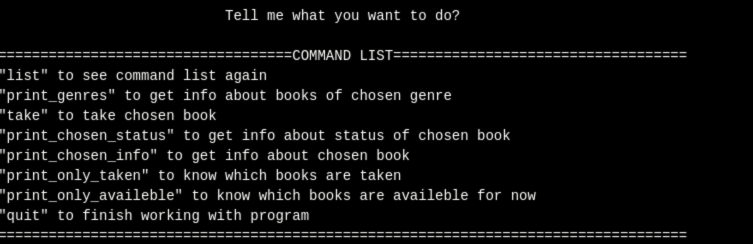


Рисунок 3.1. Головне меню зі списком доступних команд

Одразу при виборі команди програма обробляє введену користувачем команду і перевіряє чи коректно вона введено. Якщо так, то команда виконується. Для перевірки коректності роботи програми був зроблений тестовий запуск «рис. 3.2.», який показав, що команди виконуються відповідно до всіх вимог.

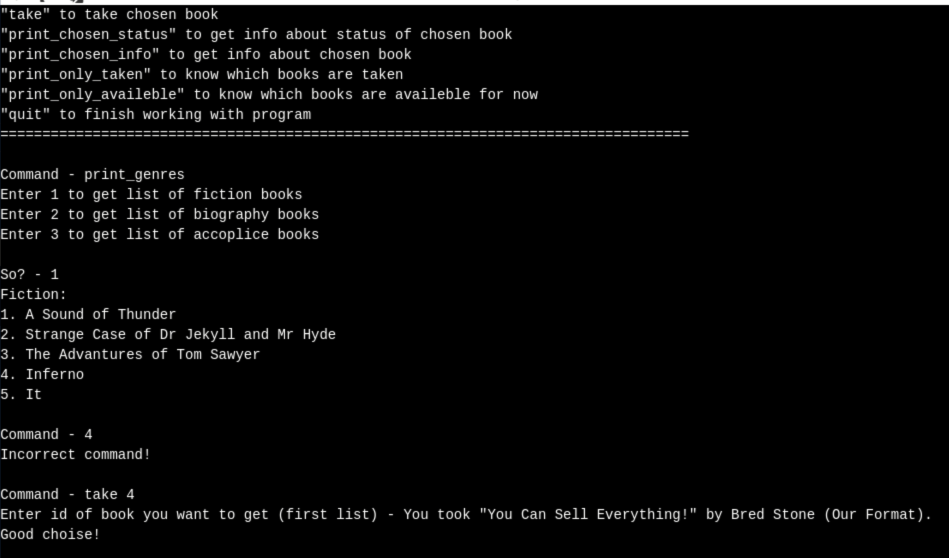


Рисунок 3.2. Тестовий запуск з виконанням всіх команд

Висновки

В даній курсовій роботі було розроблена програма “Особистої бібліотеки та картотеки” . Для виконання поставленого завдання були розкриті такі етапи:

* аналіз та опис предметної області;
* блок-схема програми, яка детально розкриває роботу алгоритму;
* кодування гри;
* аналіз результатів;

До недоліків роботи можна віднести:

* Не реалізована функції видалення книги
* Недостатня кількість жанрів і книжок

Теоретичні відомості були закріплені практичною роботою.

Список використаних джерел

1. Львов М. С. Основи алгоритмізації та програмування / М. С. Львов, О. В. Співаковський. Херсон, 1997. – 256 с.
2. Поляков К. Програмування мовою "С" / К. Поляков. Київ, 2009. – 45 с.
3. Кормен Т. Вступ до алгоритмів / Т. Кормен, Ч. Лейзерсон, Р. Рівест, К. Стайн., 2019. – 1216 с. (3).
4. Уоррен Г. Алгоритмічні трюки для програмістів / Генрі Уоррен. Бостон: Addison-Wesley, 2014. – 509 с. (2).
5. Ковалюк Т. В. Основи програмування / Т. В. Ковалюк. Київ: BHV, 2005. – 385 с.
6. Йен Д. Алгоритм знаходження найкоротшого шляху в загальній мережі / Джин Йен. Маямі: Mathematical Reviews, 1971. – 530 с.
7. Бентлі Д. Перли програмування / Джон Бентлі. Бостон: Addison-Wesley, 2000. (2).

Додатки

**Додаток А. Лістинг програми**

#include <iostream>

#include <string>

#include <chrono>

#include <thread>

using namespace std;

char kov = '"';

class library

{

public:

void print\_status()

{

for (int i = 0, j = 1; i < 10; i++)

{

cout << j << ". ";

lib[i].print\_info();

j++;

}

}

void print\_only\_availeble()

{

for (int i = 0, j = 1; i < 10; i++)

{

lib[i].check\_availability();

if (lib[i].available)

{

cout << j << ". ";

lib[i].print\_info();

j++;

}

}

}

void print\_only\_taken()

{

for (int i = 0, j = 1; i < 10; i++)

if (lib[i].isTaken)

{

cout << j << ". ";

lib[i].print\_info();

j++;

}

}

void print\_chosen\_status(unsigned int id)

{

for (int i = 0; i < 10; i++)

if (i == id - 1)

lib[i].print\_status();

}

void print\_chosen\_info(unsigned int id)

{

for (int i = 0; i < 10; i++)

if (i == id - 1)

lib[i].print\_info();

}

void print\_ids()

{

for (int i = 0, j = 1; i < 10; i++)

{

cout << "Enter " << j << " to get info or status about " << lib[i].name << endl;

j++;

}

}

void print\_accoplices()

{

cout << "Accoplices:" << endl;

for (int i = 0, j = 1; i < 3; i++)

{

cout << j << ". " << accomplice[i].name << endl;

j++;

}

}

void print\_biographies()

{

cout << "Biographies:" << endl;

for (int i = 0, j = 1; i < 2; i++)

{

cout << j << ". "<< biography[i].name << endl;

j++;

}

}

void print\_fiction()

{

cout << "Fiction:" << endl;

for (int i = 0, j = 1; i < 5; i++)

{

cout << j << ". "<< fiction[i].name << endl;

j++;

}

}

void print\_genres\_ids()

{

cout << "Enter 1 to get list of fiction books" << endl;

cout << "Enter 2 to get list of biography books" << endl;

cout << "Enter 3 to get list of accoplice books" << endl;

}

void take(unsigned int id)

{

for (int i = 0; i < 10; i++)

if (i == id - 1)

lib[i].take\_book();

}

private:

class book

{

public:

book(string author, string name, string edition)

{

this->author = author;

this->name = name;

this->edition = edition;

}

void print\_info()

{

if (this->edition == "")

cout << kov << this->name << kov << " by " << this->author << "." << endl;

else

cout << kov << this->name << kov << " by " << this->author << " (" << this->edition << ")." << endl;

}

void print\_status()

{

this->check\_availability();

if (available)

cout << "We have it. You can take it, if you want." << endl;

else

cout << "Oh, sorry, we don't have it. Looks like someone already took it." << endl;

}

void take\_book()

{

this->isTaken = true;

}

void check\_availability()

{

if (isTaken)

this->available = false;

else

this->available = true;

}

private:

string author;

string name;

string edition;

bool isTaken = false;

bool available = true;

friend library;

};

book lib[10] =

{

book("Ray Bradbury", "A Sound of Thunder", "The Golden Apples of the Sun"),

book("Robert Stevenson", "Strange Case of Dr Jekyll and Mr Hyde", ""),

book("Mark Twen", "The Advantures of Tom Sawyer", "American Publishing Company"),

book("Bred Stone", "You Can Sell Everything!", "Our Format"),

book("Daniil Teslenko", "Against All Ods. The Way to Success", ""),

book("Tetiana Kovaliuk", "The Basics of Prgraming", "BHV"),

book("Ivan Saveliev", "General Physics Course", "Moscow: Science"),

book("Dan Brown", "Inferno", ""),

book("Neal Ford", "Building Evolutionary Architectures", ""),

book("Steven King", "It", "")

};

book accomplice[3] = { lib[5], lib[6] ,lib[8] };

book fiction[5] = { lib[0], lib[1], lib[2], lib[7], lib[9] };

book biography[2] = { lib[3], lib[4] };

};

void print\_command\_list()

{

cout << endl;

cout << "===================================COMMAND LIST===================================" << endl;

cout << kov << "list" << kov << " to see command list again" << endl;

cout << kov << "print\_genres" << kov << " to get info about books of chosen genre" << endl;

cout << kov << "take" << kov << " to take chosen book" << endl;

cout << kov << "print\_chosen\_status" << kov << " to get info about status of chosen book" << endl;

cout << kov << "print\_chosen\_info" << kov << " to get info about chosen book" << endl;

cout << kov << "print\_only\_taken" << kov << " to know which books are taken" << endl;

cout << kov << "print\_only\_availeble" << kov << " to know which books are availeble for now" << endl;

cout << kov << "quit" << kov << " to finish working with program" << endl;

cout << "==================================================================================" << endl;

}

int main()

{

unsigned int id;

string command;

library lib;

cout << "Hello! And welcome to your own virtual library!" << endl;

this\_thread::sleep\_for(std::chrono::seconds(2));

cout << "Fist of let me show you our list" << endl << endl;

this\_thread::sleep\_for(std::chrono::seconds(2));

lib.print\_status();

cout << endl;

this\_thread::sleep\_for(std::chrono::seconds(2));

cout << " Tell me what you want to do?" << endl;

print\_command\_list();

while (command != "quit")

{

cout << endl << "Command - ";

cin >> command;

if (command == "print\_genres" || command == "take" || command == "print\_chosen\_status" || command == "print\_chosen\_info" || command == "print\_only\_taken" || command == "print\_only\_availeble" || command == "list")

{

if (command == "print\_genres")

{

lib.print\_genres\_ids();

cout << endl << "So? - ";

cin >> id;

if (id == 1)

lib.print\_fiction();

else

if (id == 2)

lib.print\_biographies();

else

if (id == 3)

lib.print\_accoplices();

else

cout << "Incorrect number" << endl;

}

if (command == "take")

{

cout << "Enter id of book you want to get (first list) - ";

cin >> id;

if (id < 11)

{

lib.take(id);

cout << "You took ";

lib.print\_chosen\_info(id);

cout << "Good choise!" << endl;

}

else

cout << "Incorrect number!" << endl;

}

if (command == "print\_chosen\_status")

{

cout << "Enter id of book you want to know abut (first list) - ";

cin >> id;

lib.print\_chosen\_status(id);

}

if (command == "print\_chosen\_info")

{

cout << "Enter id of book you want get info about (first list) - ";

cin >> id;

lib.print\_chosen\_info(id);

}

if (command == "print\_only\_taken")

lib.print\_only\_taken();

if (command == "print\_only\_availeble")

lib.print\_only\_availeble();

if (command == "list")

print\_command\_list();

}

else

if (command != "quit")

cout << "Incorrect command!" << endl;

}

cout << "Thank you for working with this library! (Created by Khomichak Mykola)" << endl;

return 0;

}