

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN



PBL2: ĐỒ ÁN CƠ SỞ LẬP TRÌNH

**ĐỀ TÀI: XÂY DỰNG HỆ THỐNG
QUẢN LÝ THƯ VIỆN**

GIẢNG VIÊN HƯỚNG DẪN: THS. NGUYỄN VĂN NGUYỄN

SINH VIÊN THỰC HIỆN:

NGUYỄN THỊ NGỌC ÁNH

LỚP: 22T_NHAT2 NHÓM: 22.10B

LÝ HOÀNG QUYÊN

LỚP: 22T_DT5 NHÓM: 22.10B

LỜI MỞ ĐẦU

Như chúng ta đã biết thư viện là trung tâm thông tin ở nhiều dạng khác nhau, tạo điều kiện cho người sử dụng tiếp cận nhanh chóng đến tri thức. Cùng với việc phát triển đất nước, thư viện cũng ngày càng được mở rộng với khối lượng sách khổng lồ và lượng thông tin cần quản lý ngày một tăng. Vì lẽ đó mà yêu cầu về việc quản lý thư viện ngày càng được mở rộng.

Đến với đề tài lần này chúng em được hướng dẫn, nghiên cứu để tạo nên một chương trình nhằm mục đích ứng dụng một cách chính xác, nhanh chóng và hiệu quả các thuật toán vào hệ thống quản lý thư viện. Chính vì thế, dự án lần này sẽ giúp chúng em được tiếp cận với thực tế, vận dụng kiến thức, lý thuyết đã học vào thực nghiệm. Đặc biệt là tự rút ra được những bài học kinh nghiệm quý báu cho bản thân. Từ đó, sinh viên có cái nhìn rộng hơn về ngành mà mình đang theo học và định hướng phát triển đúng đắn cho tương lai.

Do trình độ và kinh nghiệm của chúng em vẫn còn nhiều hạn chế nên sẽ không tránh khỏi những sai sót nhất định trong quá trình làm đồ án. Mong quý thầy cô thông cảm và chúng em cũng rất mong nhận được những lời góp ý đến từ thầy cô và các bạn.

Chúng em xin chân thành cảm ơn!

MỤC LỤC

LỜI MỞ ĐẦU	1
MỤC LỤC	2
DANH MỤC HÌNH VẼ	3
1. GIỚI THIỆU ĐỀ TÀI.....	4
1.1. TỔNG QUAN ĐỀ TÀI	5
1.1.1. Giao diện người dùng	5
1.1.2. Các chức năng chính	5
1.1.3. Định dạng và màu sắc	5
1.1.4. Cách sử dụng	5
1.2. THUẬT TOÁN, CẤU TRÚC DỮ LIỆU VÀ HƯỚNG ĐỐI TƯỢNG	6
1.2.1. Thuật toán	6
1.2.2. Ứng dụng cấu trúc dữ liệu	8
1.2.3. Ứng dụng hướng đối tượng	9
2. TRIỂN KHAI VÀ KẾT QUẢ THỰC THI CHƯƠNG TRÌNH.....	12
2.1. TRIỂN KHAI CHƯƠNG TRÌNH.....	12
2.1.1. Chương trình bắt đầu	12
2.1.2. Giao diện ban đầu.....	12
2.1.3. Chức năng thêm sách (Lựa chọn 1).....	12
2.1.4. Chức năng tìm sách theo tên (Lựa chọn 2).....	12
2.1.5. Chức năng in danh sách sách trong thư viện (Lựa chọn 3).....	12
2.1.6. Thoát khỏi chương trình (Lựa chọn 4).....	12
2.2. KẾT QUẢ THỰC THI	13
KẾT LUẬN	18
HƯỚNG PHÁT TRIỂN	20
TÀI LIỆU THAM KHẢO	21
PHỤ LỤC	22

DANH MỤC HÌNH VẼ

Hình 1 . Bảng giới thiệu.	13
Hình 2 . Giao diện ban đầu nếu file danh sách rỗng.	13
Hình 3 . Giao diện ban đầu nếu file có dữ liệu.	14
Hình 4 . Giao diện chính của chương trình.	14
Hình 5 . Nhập danh sách sách.	15
Hình 6 . Nhập danh sách sách.	15
Hình 7 . Thông báo cập nhật thành công.	16
Hình 8 . Tìm sách theo tên.	16
Hình 9 . Liệt kê sách trong thư viện.	17

1. GIỚI THIỆU ĐỀ TÀI

Thư viện đóng vai trò quan trọng trong việc duy trì, tổ chức và cung cấp nguồn tri thức cho cộng đồng. Để tối ưu hóa quá trình quản lý sách, đề tài "Xây dựng hệ thống quản lý thư viện" ra đời nhằm xây dựng một hệ thống linh hoạt và hiệu quả.

Chương trình được viết bằng ngôn ngữ C++, sử dụng hướng đối tượng để xây dựng các lớp và phương thức thực hiện các công việc sau:

- Nhập dữ liệu cho đối tượng từ bàn phím
- Thực hiện thêm sách vào thư viện.
- Thực hiện tìm kiếm thông tin sách theo tên.
- Liệt kê số sách hiện có trong thư viện ra màn hình.
- Đầu vào (Input):
 - Đọc danh sách từ file cho trước.
 - Nhập từ bàn phím một danh sách liên kết chứa thông tin sách trong thư viện.
 - Nhập thông tin sách muốn thêm vào thư viện.
 - Nhập tên cuốn sách muốn tìm kiếm.
- Đầu ra (Output):
 - In ra màn hình danh sách sách có trong thư viện.
 - Hiển thị thông tin quyển sách muốn tìm kiếm.
 - Liệt kê danh sách khi đã được cập nhật.

Yêu cầu: Viết chương trình C++ ứng dụng hướng đối tượng và cấu trúc dữ liệu để xây dựng các lớp và phương thức thực hiện các công việc quản lý thư viện.

1.1. TỔNG QUAN ĐỀ TÀI

1.1.1. *Giao diện người dùng*

Chương trình sử dụng thư viện Console tự định nghĩa để quản lý và hiển thị thông tin trên màn hình.

Giao diện được thiết kế để hiển thị thông tin về thư viện, các tùy chọn cho người dùng, và kết quả của các hoạt động.

1.1.2. *Các chức năng chính*

-*Đọc danh sách từ file:* khi thực hiện chương trình sẽ mặc định chạy một file chứa danh sách có sẵn, nếu file rỗng sẽ thông báo thư viện chưa có sách, ngược lại danh sách được in ra màn hình.

-*Thêm sách vào thư viện:* Người dùng có thể nhập thông tin về sách, bao gồm tên sách, tổng số quyển và số sách đang mượn. Thông tin này được lưu trữ trong danh sách sách trong thư viện (ThuVien).

-*Tìm kiếm sách theo tên:* Người dùng có thể tìm kiếm sách trong thư viện bằng cách nhập tên sách. Chương trình sử dụng phương thức TimSach của lớp ThuVien để thực hiện tìm kiếm.

-*In danh sách sách trong thư viện:* Người dùng có thể in danh sách toàn bộ sách đang còn trong thư viện. Chương trình sử dụng ThuVien để in danh sách này.

-*Thoát khỏi chương trình:* Người dùng có thể thoát khỏi chương trình.

1.1.3. *Định dạng và màu sắc*

Chương trình sử dụng các hàm được định nghĩa trong thư viện Console để định dạng và màu sắc cho văn bản và giao diện người dùng, điều này làm cho giao diện trở nên thân thiện và dễ đọc.

1.1.4. *Cách sử dụng*

- Người dùng có thể sử dụng các phím số để chọn các tùy chọn tương ứng với các chức năng.

- Chương trình hiển thị thông báo và yêu cầu người dùng nhập dữ liệu khi cần thiết.

1.2. THUẬT TOÁN, CẤU TRÚC DỮ LIỆU VÀ HƯỚNG ĐỐI TƯỢNG

1.2.1. Thuật toán

Chúng ta áp dụng thuật toán tìm kiếm tuyến tính cho việc thêm sách và tìm kiếm sách trong thư viện.

- Khi quản trị viên muốn tìm kiếm thông tin về một quyển sách cụ thể trong thư viện, thuật toán tìm kiếm tuyến tính được sử dụng. Các bước thực hiện như sau:
 - Duyệt qua từng quyển sách trong danh sách theo thứ tự.
 - So sánh tên sách cần tìm với tên của quyển sách đang duyệt.
 - Nếu tìm thấy, trả về thông tin về quyển sách; nếu không, tiếp tục duyệt đến khi hết danh sách.

Code:

```
CSach *ThuVien::TimSach(string tenSach)
```

```
{  
    CSach *current = head;  
    while (current != NULL)  
    {  
        if (current->GetTenSach() == tenSach)  
        {  
            return current;  
        }  
        current = current->GetNext();  
    }  
    return NULL;  
}
```

- Khi muốn thêm một quyển sách mới vào thư viện, quản trị viên sử dụng thuật toán tìm kiếm tuyến tính để kiểm tra xem quyển sách đã tồn tại trong danh sách hay chưa. Quy trình này bao gồm:
 - Duyệt qua từng quyển sách trong danh sách.
 - So sánh tên sách mới với tên của quyển sách đang duyệt.
 - Nếu tìm thấy, cập nhật thông tin về số lượng sách; nếu không, thêm sách mới vào danh sách.

Code:

```
void ThuVien::ThemSach(CSach *Sach)
{
    count++;

    if (head == NULL)
    {
        head = Sach;
    }
    else
    {
        CSach *current = head;

        while (current->GetNext() != NULL)
        {
            current = current->GetNext();
        }

        current->SetNext(Sach);
    }
}
```

Thuật toán tìm kiếm tuyến tính này giúp đảm bảo tính nhất quán và dễ dàng trong việc quản lý thông tin sách. Nó cũng thể hiện tính hiệu quả trong việc tìm kiếm và xác định vị trí của sách trong thư viện, đồng thời cung cấp giải pháp đơn giản và hiệu quả cho các tác vụ quản lý sách.

1.2.2. Ứng dụng cấu trúc dữ liệu

Danh sách liên kết đơn là một cấu trúc dữ liệu trong lập trình máy tính, nơi mỗi phần tử (nút) trong danh sách chứa dữ liệu và một liên kết đến phần tử tiếp theo trong danh sách. Danh sách liên kết đơn được sử dụng để tổ chức dữ liệu theo cách linh hoạt và hiệu quả, đặc biệt là khi cần thêm hoặc xóa phần tử một cách thường xuyên và không biết trước kích thước chính xác của dữ liệu.

Chương trình sử dụng danh sách liên kết để lưu và liệt kê thông tin sách có trong thư viện.

Code:

```
class CSach
{
private:
    string TenSach;
    int TongSoQuyenSach;
    int SoSachDangMuon;
    CSach *next;
};
```

1.2.3. Ứng dụng hướng đối tượng

Chương trình xây dựng các lớp đối tượng có các thuộc tính cũng như phương thức tương ứng để quản lý thông tin sách trong thư viện.

1.2.3.1 Lớp CSach

- Là một class đại diện cho thông tin của một cuốn sách trong thư viện.
- Danh sách thuộc tính:
 - TenSach: Lưu trữ tên của quyển sách.
 - TongSoQuyenSach: Lưu trữ tổng số quyển sách có trong thư viện.
 - SoSachDangMuon: Lưu trữ số quyển sách đang được mượn.
 - Con trỏ next: Là con trỏ đến đối tượng sách tiếp theo trong danh sách (nếu có)
- Danh sách phương thức:
 - CSach(string, int, int): Hàm khởi tạo để thiết lập giá trị ban đầu cho các thuộc tính khi một đối tượng sách được tạo.
 - GetTenSach(): Phương thức trả về tên của sách.
 - GetTongSoQuyenSach(): Phương thức trả về tổng số quyển sách.
 - GetSoSachDangMuon(): Phương thức trả về số quyển sách đang được mượn.
 - GetSoSachConLai(): Phương thức tính và trả về số sách còn lại trong thư viện (tổng số - số đang cho mượn).
 - GetNext(): Phương thức trả về con trỏ đến đối tượng sách tiếp theo (nếu có).
 - SetNext(CSach *next): Phương thức để thiết lập con trỏ đến đối tượng sách tiếp theo.
 - SetTongSoQuyenSach(int): Phương thức để thiết lập tổng số quyển sách.

- SetSoSachDangMuon(int): Phương thức để thiết lập số quyển sách đang được mượn.

Độ phức tạp của các hàm trong lớp này đều là $O(1)$, vì chúng đơn giản là truy cập và gán các thuộc tính của lớp.

Code:

```
class CSach
{
private:
    string TenSach;
    int TongSoQuyểnSach;
    int SoSachDangMuon;
    CSach *next;
public:
    CSach(string , int, int);
    string GetTenSach();
    int GetTongSoQuyểnSach();
    int GetSoSachDangMuon();
    int GetSoSachConLai();
    CSach *GetNext();
    void SetNext(CSach *next);
    void SetTongSoQuyểnSach(int);
    void SetSoSachDangMuon(int);
};
```

1.2.3.2 Lớp *ThuVien*

Lớp này quản lý danh sách các sách trong thư viện.

- **Con trỏ head** là con trỏ đến đối tượng sách đầu tiên trong danh sách sách trong thư viện.
- **Biến count** là biến tĩnh (static) được sử dụng để theo dõi tổng số lượng đối tượng CSach được tạo.
- **Hàm ThemSach** có độ phức tạp là $O(n)$, với n là số sách trong danh sách, vì nó phải duyệt qua danh sách để thêm một cuốn sách mới.
- **Hàm NhapDanhSach** có độ phức tạp là $O(n)$ với n là số lượng sách mới cần nhập, vì nó phải thực hiện lặp qua từng cuốn sách để nhập thông tin.
- **Hàm InDanhSach** có độ phức tạp là $O(n)$, với n là số sách trong danh sách, vì nó phải lặp qua danh sách để in thông tin.
- **Hàm TimSach** có độ phức tạp là $O(n)$ trong trường hợp xấu nhất với n là số sách trong danh sách, vì nó phải duyệt qua danh sách để tìm sách.

Code:

```
class ThuVien
{
    public:
        CSach *head;
        static int count;
        ThuVien();
        void ThemSach(CSach*);
        void NhapDanhSach();
        void InDanhSachSach(int,int,int);
        CSach* TimSach(string);
};
```

2. TRIỂN KHAI VÀ KẾT QUẢ THỰC THI CHƯƠNG TRÌNH

2.1. TRIỂN KHAI CHƯƠNG TRÌNH

2.1.1. Chương trình bắt đầu

Chương trình bắt đầu bằng việc hiển thị một màn hình giới thiệu, trong đó nó hiển thị tên tác giả và thông tin về dự án, sau đó đợi người dùng nhấn bất kỳ phím nào để tiếp tục.

2.1.2. Giao diện ban đầu

- Sau khi người dùng nhấn phím, chương trình xóa màn hình và hiển thị giao diện chính.
- Giao diện này gồm một số phần:
 - + Một tiêu đề thông báo về "Quản lý thư viện".
 - +Chương trình sẽ mặc định đọc một file danh sách cho trước, nếu file rỗng chương trình sẽ thông báo thư viện chưa có sách, ngược lại màn hình in ra danh sách sách có sẵn trong file. Chương trình tiếp tục thực hiện.
 - +Một danh sách các lựa chọn dưới dạng số từ 1 đến 4, tương ứng với các chức năng khác nhau.
 - +Một lựa chọn nhập số để chọn một trong các chức năng.

2.1.3. Chức năng thêm sách (Lựa chọn 1)

- Người dùng chọn lựa chọn 1 và nhập thông tin về một cuốn sách mới, bao gồm tên sách, tổng số quyển và số sách đang mượn.
- Sau khi thông tin được nhập, chương trình thêm cuốn sách vào danh sách thư viện hoặc cập nhật thông tin cuốn sách nếu nó đã tồn tại.

2.1.4. Chức năng tìm sách theo tên (Lựa chọn 2)

- Người dùng chọn lựa chọn 2 và nhập tên sách mà họ muốn tìm kiếm.
- Chương trình tìm kiếm trong danh sách thư viện và hiển thị thông tin về cuốn sách nếu nó tồn tại. Nếu không, nó thông báo rằng cuốn sách không tồn tại.

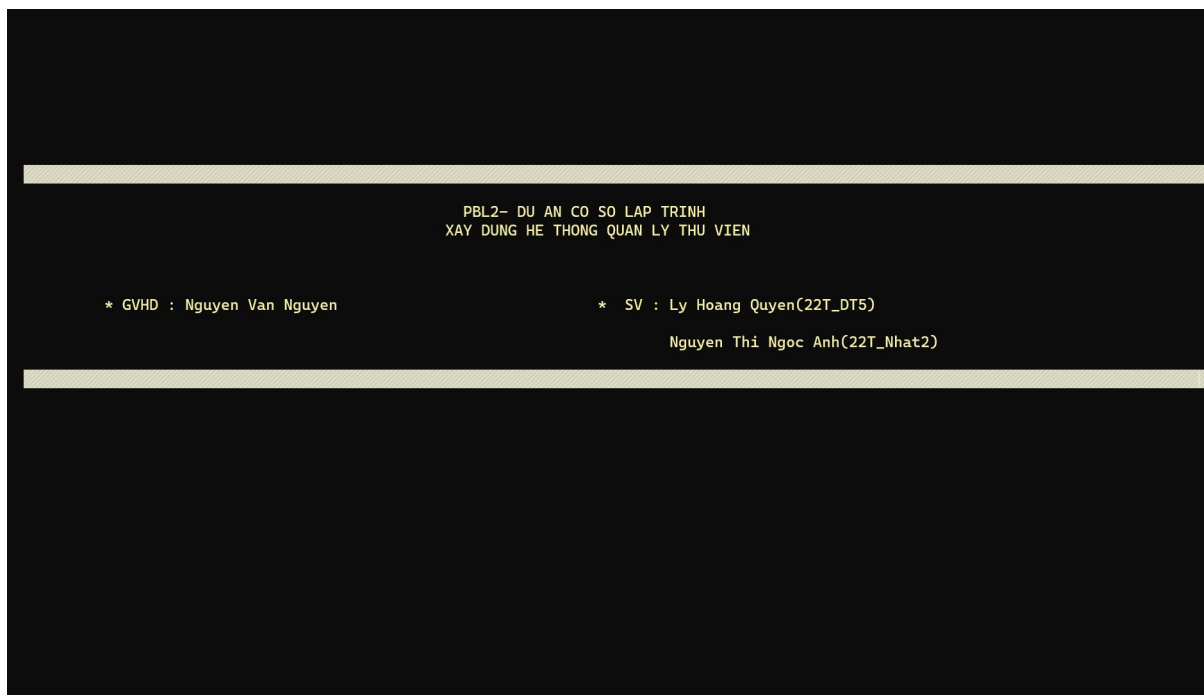
2.1.5. Chức năng in danh sách sách trong thư viện (Lựa chọn 3)

- Người dùng chọn lựa chọn 3 và chương trình hiển thị danh sách toàn bộ sách trong thư viện, bao gồm thông tin về tên sách, tổng số quyển, số sách đang mượn và số sách còn lại.

2.1.6. Thoát khỏi chương trình (Lựa chọn 4)

- Người dùng chọn lựa chọn 4 và chương trình kết thúc.

2.2. KẾT QUẢ THỰC THI



Hình 1. Bảng giới thiệu.



Hình 2. Giao diện ban đầu nếu file danh sách rỗng.

12/02/23

** QUAN LY SACH THU VIEN **

DANH SACH SACH TRONG THU VIEN

Ten sach	Tong so sach	So sach dang muon	So sach con lai
Sach1	150	45	105
Sach2	54	29	25
Sach3	536	149	387
Sach4	84	12	72
Sach5	800	440	360
Sach6	430	319	111
Sach7	778	98	680
Sach8	787	433	354
Sach9	547	519	28
Sach10	706	447	259
Sach11	924	620	304
Sach12	966	47	919
Sach13	252	180	72

Hình 3. Giao diện ban đầu nếu file có dữ liệu.

12/02/23

TRƯỜNG ĐẠI HỌC BẠCH KHOA
** QUAN LY SACH THU VIEN **

BANG LUA CHON

1. Cap nhât them sach
2. Tim sach theo ten
3. Liet ke sach hien co trong thu vien
4. Thoat

Nhan lua chon (1-4) :|

Hình 4. Giao diện chính của chương trình.

12/02/23

TRƯỜNG ĐẠI HỌC BẠCH KHOA
** QUẢN LÝ SÁCH THƯ VIỆN **

BẢNG LỰA CHỌN

1. Cập nhật thêm sách

2. Tìm sách theo tên

3. Liệt kê sách hiện có trong thư viện

4. Thoát

Nhấn lựa chọn (1-4) :1

Nhập số lượng sách: 2

Nhập tên sách thu 1: Sach31

Nhập tổng số quyển của sách thu 1: 50

Nhập số sách đang mượn của sách thu 1: 35

Hình 5. Nhập danh sách sách.

12/02/23

TRƯỜNG ĐẠI HỌC BẠCH KHOA
** QUẢN LÝ SÁCH THƯ VIỆN **

BẢNG LỰA CHỌN

1. Cập nhật thêm sách

2. Tìm sách theo tên

3. Liệt kê sách hiện có trong thư viện

4. Thoát

Nhấn lựa chọn (1-4) :1

Nhập số lượng sách: 2

Nhập tên sách thu 2: Sach32

Nhập tổng số quyển của sách thu 2: 100

Nhập số sách đang mượn của sách thu 2: 32

Hình 6. Nhập danh sách sách.

TRƯỜNG ĐẠI HỌC BẠCH KHOA ** QUẢN LÝ SÁCH THƯ VIỆN **		12/02/23
BẢNG LỰA CHỌN 1. Cập nhật thêm sách 2. Tìm sách theo tên 3. Liệt kê sách hiện có trong thư viện 4. Thoát Nhấn lựa chọn (1-4) :1	CẬP NHẬT SÁCH THÀNH CÔNG	

Hình 7. Thông báo cập nhật thành công.

TRƯỜNG ĐẠI HỌC BẠCH KHOA ** QUẢN LÝ SÁCH THƯ VIỆN **		12/02/23
BẢNG LỰA CHỌN 1. Cập nhật thêm sách 2. Tìm sách theo tên 3. Liệt kê sách hiện có trong thư viện 4. Thoát Nhấn lựa chọn (1-4) :2	Nhập tên sách cần tìm : Sach25 Tên sách :Sach25 Tổng số quyển sách :154 Số sách đang mượn :65	

Hình 8. Tìm sách theo tên.

12/02/23

** QUAN LY SACH THU VIEN **

DANH SACH SACH CON LAI TRONG THU VIEN

Ten sach	Tong so sach	So sach dang muon	So sach con lai
Sach1	150	45	105
Sach2	54	29	25
Sach3	536	149	387
Sach4	84	12	72
Sach5	800	440	360
Sach6	430	319	111
Sach7	778	98	680
Sach8	787	433	354
Sach9	547	519	28
Sach10	706	447	259
Sach11	924	620	304
Sach12	966	47	919
Sach13	252	180	72

Hình 9. Liệt kê sách trong thư viện.

KẾT LUẬN

Qua quá trình nghiên cứu, chúng em nhận thấy chương trình của mình có những ưu điểm, hạn chế cũng như có những phương hướng để cải thiện sau:

- **Ưu điểm**

- **Thêm và cập nhật sách:** Chương trình cho phép người dùng thêm sách mới vào thư viện và cập nhật thông tin về sách, bao gồm tổng số quyển và số sách đang mượn.

- **Tìm kiếm sách:** Người dùng có khả năng tìm kiếm sách theo tên. Điều này giúp họ dễ dàng tìm thông tin về một cuốn sách cụ thể.

- **In danh sách sách:** Chương trình cho phép người dùng in danh sách toàn bộ sách có trong thư viện, bao gồm thông tin về tên sách, tổng số quyển, số sách đang mượn và số sách còn lại.

- **Hạn chế:**

- **Phân quyền và xác thực:** Chương trình không có chức năng phân quyền hoặc xác thực người dùng.

- **Giao diện người dùng hạn chế:** Giao diện của chương trình là dòng lệnh cơ bản và không cung cấp sự tương tác thân thiện với người dùng. Một giao diện đồ họa hoặc một giao diện người dùng đa dạng hơn có thể làm cho ứng dụng dễ sử dụng hơn.

Tổng quan, chương trình trên là một bản thử nghiệm đơn giản về cách quản lý thông tin sách trong thư viện. Để tạo một ứng dụng thực tế và mạnh mẽ hơn, cần cải thiện và mở rộng nhiều khía cạnh, bao gồm tính bảo mật, lưu trữ dữ liệu, và giao diện người dùng.

PBL2: DỰ ÁN CƠ SỞ LẬP TRÌNH

Qua quá trình tìm hiểu và thực hiện đề án chúng em được nghiên cứu và hiểu rõ hơn về lập trình theo hướng đối tượng cũng như về cấu trúc dữ liệu và các phương pháp giải quyết vấn đề quản lý hệ thống thư viện. Chúng em luôn hướng tới sự tối ưu của chương trình để đảm bảo tính chính xác, thời thực hiện nhanh chóng, hiệu năng cao cũng như giao diện thân thiện, đẹp mắt và dễ sử dụng với người dùng. Các cấu trúc dữ liệu và thuật toán được phân tích và lựa chọn kỹ lưỡng để đem lại hiệu quả xử lý là tiết kiệm tài nguyên bộ nhớ.

HƯỚNG PHÁT TRIỂN

Trong quá trình nghiên cứu, tìm hiểu và thực hiện đồ án, chúng em được phát triển hơn về nhiều mặt như: kỹ năng, tư duy giải quyết vấn đề và tìm hiểu thêm các cấu trúc dữ liệu, thuật toán phù hợp với đề tài.

Dự án lần này của chúng em vẫn còn nhiều hạn chế và thiếu sót. Do đó, chúng em đề xuất một số phương thức cải thiện như sau:

- **Phân quyền và xác thực:** Cải thiện tính bảo mật bằng cách thêm tính năng phân quyền và xác thực người dùng. Điều này giúp ngăn chặn truy cập trái phép và chỉ cho phép người dùng có quyền cụ thể thực hiện các chức năng.

- **Giao diện đồ họa:** Cân nhắc phát triển một giao diện đồ họa thân thiện với người dùng để làm cho ứng dụng trở nên dễ sử dụng hơn và trực quan hơn.

Trong tương lai chúng em sẽ cố gắng nghiên cứu, tìm hiểu thêm nhiều hướng giải quyết để phù hợp hơn với đề tài để tạo nên một phần mềm hiệu quả và dễ tiếp cận nhất.

TÀI LIỆU THAM KHẢO

- [1] Lê Thị Mỹ Hạnh, *Bài giảng môn lập trình hướng đối tượng*, Tài liệu lưu hành nội bộ, 2023.
- [2] Phan Chí Tùng, *Bài giảng học phần Cấu trúc dữ liệu*, tài liệu lưu hành nội bộ, Đà Nẵng, 2023.
- [3] Mark Allen Weiss, *Data Structures and Algorithm Analysis in C++*, 2023.
- [4] Wisnu Anggoro, *C++ Data Structures and Algorithms*, 2023.
- [5] Adam Drozdek, *Data Structures and Algorithms*, 2023.
- [6] Trần Đức Minh, *Lập trình hướng đối tượng với C++*, 2023.
- [7] Nguyễn Mạnh Hùng, *Lập trình hướng đối tượng C++*, 2023.
- [8] Lê Minh Hoàng, *Giải thuật và cấu trúc dữ liệu*, 2023.
- [9] Nguyễn Văn Nguyên, *Tài liệu thực hành kỹ thuật lập trình*, 2022.
- [10] Nguyễn Thị Minh Hỷ, *Kỹ thuật lập trình C/C++*, 2022.

PHỤ LỤC

File Thuvien.h:

```
#include "Console.h"
#include "CSach.h"
#include <iostream>
#include <string>
#include <windows.h>
#include<iomanip>
#include<conio.h>
using namespace std;
class ThuVien
{
    public:
    CSach *head;
    static int count;
    ThuVien();
    void ThemSach(CSach*);
    void NhapDanhSach();
    void InDanhSachSach(int,int,int);
    CSach* TimSach(string);
};
```

File ThuVien.cpp:

```
#include "ThuVien.h"

using namespace std;

int ThuVien::count = 0;

CSach *head;

ThuVien::ThuVien()
{
    head = NULL;
}

void ThuVien::ThemSach(CSach *Sach)
{
    count++;

    if (head == NULL)
    {
        head = Sach;
    }
    else
    {
        CSach *current = head;
        while (current->GetNext() != NULL)
        {
            current = current->GetNext();
        }
        current->SetNext(Sach);
    }
}
```



```
void ThuVien::NhapDanhSach()
{
    int n;
    int x = 50, y = 7;
    Console::gotoxy(x, y);
    cout << "Nhap so luong sach: ";
    cin >> n;
    cin.ignore();
    for (int i = 0; i < n; i++)
    {
        string TenSach;
        int TongSoQuyên, SoSachDangMuon;
        Console::gotoxy(x, y + 1);
        cout << "Nhap ten sach thu " << i + 1 << ": ";
        getline(cin, TenSach);

        Console::gotoxy(x, y + 2);
        cout << "Nhap tong so quyên của sach thu " << i + 1 << ": ";
        cin >> TongSoQuyên;

        Console::gotoxy(x, y + 3);
        cout << "Nhap so sach dang muon của sach thu " << i + 1 << ": ";
        cin >> SoSachDangMuon;

        while (SoSachDangMuon > TongSoQuyên)
        {
```

```
Console::gotoxy(x, y + 6);

cout << "So sach dang muon khong hop le! Vui long nhap lai";

getch();

Console::gotoxy(x, y + 6);

for (int i = 0; i < 50; i++)

    cout << " ";

Console::gotoxy(x + 39, y + 3);

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

    cout << " ";

Console::gotoxy(x + 39, y + 3);

cin >> SoSachDangMuon;

}
```

```
Console::gotoxy(x, y + 1);

for (int i = 0; i <= 50; i++)

    cout << " ";

Console::gotoxy(x, y + 2);

for (int i = 0; i <= 50; i++)

    cout << " ";

Console::gotoxy(x, y + 3);

for (int i = 0; i <= 50; i++)

    cout << " ";
```

```
cin.ignore();

CSach *current = head;

while (current != NULL)

{
```

```
        if (current->GetTenSach() == TenSach)
        {
            current->SetTongSoQuyensach(TongSoQuyensach);
            current->SetSoSachDangMuon(SoSachDangMuon);
            break;
        }
        current = current->GetNext();
    }
    if (current == NULL)
    {
        CSach *sach = new CSach(TenSach, TongSoQuyensach, SoSachDangMuon);
        ThemSach(sach);
    }
}

Console::gotoxy(x, y);
cout << "CAP NHAT SACH THANH CONG";
getch();
Console::gotoxy(x, y);
for (int i = 0; i <= 30; i++)
    cout << " ";
}

void ThuVien::InDanhSachSach(int x, int y, int z = -1)
{

    Console::manhinhdanhhsach();
```

```
Console::khungtrong(0);

if (z == 0)
{
    Console::gotoxy(46, 5);

    cout << "DANH SACH SACH CON LAI TRONG THU VIEN";

}
else
{
    Console::gotoxy(49, 5);

    cout << "DANH SACH SACH TRONG THU VIEN";

}

int k = 0;

CSach *current = head;

if (current == NULL)
{
    Console::khungtrong(35);

    Console::gotoxy(x+27, y);

    cout << "Thu vien chua co sach." << endl;

}
else
{
    Console::duongngang(x, ++y, 85);

    Console::gotoxy(x, ++y);

    cout << "| " << setw(20) << left << "Ten sach";

    cout << "| " << setw(17) << left << "Tong so sach";

    cout << "| " << setw(20) << left << "So sach dang muon";

    cout << "| " << setw(17) << left << "So sach con lai"
```

```
<< " |" << endl;

Console::duongngang(x, ++y, 85);

while (current != NULL)
{
    if (current->GetSoSachConLai() > z)
    {
        k++;

        Console::gotoxy(x, min(++y, 35));

        cout << "| " << setw(20) << left << current->GetTenSach();
        cout << "| " << setw(17) << left << current->GetTongSoQuyensach();
        cout << "| " << setw(20) << left << current->GetSoSachDangMuon();
        cout << "| " << setw(17) << left << current->GetSoSachConLai() << " |";

        cout << endl;

        Console::duongngang(x, min(++y, 35), 85);

        current = current->GetNext();

        Console::khungtrong(-1);

        if (k == count)

            Console::khungtrong(35);

    }

}

}
```

```
CSach *ThuVien::TimSach(string tenSach)
{
    CSach *current = head;

    while (current != NULL)
```

```
{  
    if (current->GetTenSach() == tenSach)  
    {  
        return current;  
    }  
    current = current->GetNext();  
}  
return NULL;  
}
```

File main.cpp:

```
#include "ThuVien.h"

#include <fstream>

using namespace std;

void docfile(ThuVien &thuVien)
{
    ifstream fin1("tensach.txt");
    if (fin1.is_open())
    {
        string line;
        while (getline(fin1, line))
        {
            CSach *sach = new CSach(line, 0, 0);
            thuVien.ThemSach(sach);
        }
        fin1.close();
    }

    ifstream fin2("soluong.txt");
    if (fin2.is_open())
    {
        CSach *current = thuVien.head;
        int tongsosach, tongdangmuon;
        for (int i = 1; i <= thuVien.count; i++)
        {
            fin2 >> tongsosach >> tongdangmuon;
            current->SetTongSoQuyenSach(tongsosach);
```

```
        current->SetSoSachDangMuon(tongdangmuon);

        current = current->GetNext();

    }

    fin2.close();

}

}

void luufile(ThuVien &thuVien)
{

    ifstream fin3("theend.txt");

    if (fin3.is_open())
    {

        string line;

        while (getline(fin3, line))
        {

            cout << line << endl;

        }

        fin3.close();

    }

    ofstream fout1("tensach.txt", std::ios::out | std::ios::trunc);

    if (fout1.is_open())
    {

        CSach *current = thuVien.head;

        while (current != NULL)
        {

            fout1 << current->GetTenSach() << endl;
```



```
        current = current->GetNext();

    }

}

fout1.close();

ofstream fout2("soluong.txt", std::ios::out | std::ios::trunc);

if (fout2.is_open())
{
    CSach *current = thuVien.head;
    while (current != NULL)
    {
        fout2 << current->GetTongSoQuyensach() << " " << current->GetSoSachDangMuon() << endl;
        current = current->GetNext();
    }
}

fout2.close();
}
```

```
int main()
{
    system("cls");
    ThuVien thuVien;
    docfile(thuVien);
    Console::BangGioiThieuTacGia();
    Console::manhinhdanhsach();
    thuVien.InDanhSachSach(26,7,-1);
}
```

```
    getch();

    system("cls");

    Console::manhinhchinh();

    int choice;

    while (1)
    {
        Console::gotoxy(24, 12);

        cin >> choice;

        switch (choice)
        {
            case 1:
            {
                thuVien.NhapDanhSach();

                break;
            }

            case 2:
            {
                string s;

                cin.ignore();

                Console::gotoxy(50, 7);

                cout << "Nhap ten sach can tim : ";

                getline(cin, s);

                CSach *current = thuVien.TimSach(s);

                if (current != NULL)
                {
                    Console::gotoxy(50, 8);
```

```
        cout << "Ten sach :" << current->GetTenSach() << endl;

        Console::gotoxy(50, 9);

        cout << "Tong so quyen sach :" << current->GetTongSoQuyenSach() <<
endl;

        Console::gotoxy(50, 10);

        cout << "So sach dang muon :" << current->GetSoSachDangMuon() << endl;

        cout << endl;

    }

    else

    {

        Console::gotoxy(50, 8);

        cout << "Sach khong co trong thu vien\n";

    }

    getch();

    for (int i = 7; i <= 10; i++)

    {

        Console::gotoxy(50, i);

        for (int j = 0; j <= 30; j++)

            cout << " ";

    }

    break;

}

case 3:

{

    thuVien.InDanhSachSach(27,7,0);

    _getch();

    system("cls");
```

```
        Console::manhinhchinh();

        break;

    }

    case 4:

    {

        system("cls");

        luufile(thuVien);

        exit(0);

    }

    default:

    {

        Console::gotoxy(50, 7);

        cout << "Lua chon khong hop le. Hay chon lai!";

        getch();

        Console::gotoxy(50, 7);

        for (int i = 0; i < 38; i++)

            cout << " ";

    }

    }

    Console::gotoxy(24, 12);

    cout << " ";

}

return 0;

}
```

File CSach.h:

```
#include <string>
using namespace std;
class CSach
{
private:
    string TenSach;
    int TongSoQuyensach;
    int SoSachDangMuon;
    CSach *next;
public:
    CSach(string , int, int);
    string GetTenSach();
    int GetTongSoQuyensach();
    int GetSoSachDangMuon();
    int GetSoSachConLai();
    CSach *GetNext();
    void SetNext(CSach *next);
    void SetTongSoQuyensach(int);
    void SetSoSachDangMuon(int);
};
```

File CSach.cpp:

```
#include "CSach.h"

#include<iomanip>

using namespace std;

CSach::CSach(string TenSach, int TongSoQuyênSach, int SoSachDangMuon)
{
    this->TenSach=TenSach;
    this->TongSoQuyênSach=TongSoQuyênSach;
    this->SoSachDangMuon=SoSachDangMuon;
    this->next=NULL;
}

int CSach:: GetSoSachConLai()
{
    return this->TongSoQuyênSach-this->SoSachDangMuon;
}

string CSach::GetTenSach()
{
    return this->TenSach;
}

int CSach::GetTongSoQuyênSach()
{
    return this->TongSoQuyênSach;
}

int CSach::GetSoSachDangMuon()
```

```
{  
    return this->SoSachDangMuon;  
}  
  
CSach* CSach::GetNext()  
{  
    return this->next;  
}  
  
void CSach::SetTongSoQuyenSach(int soluongthem)  
{  
    this->TongSoQuyenSach+=soluongthem;  
}  
  
void CSach::SetSoSachDangMuon(int soluongthem)  
{  
    this->SoSachDangMuon+=soluongthem;  
}  
  
void CSach::SetNext(CSach *next)  
  
{  
    this->next=next;  
}
```

File Console.h:

```
#pragma once
#include <iostream>
#include <windows.h>
#include <iomanip>
#include <conio.h>
class Console
{
public:
    static void gotoxy(short x, short y);
    static void SetBGColor(WORD color);
    static void textcolor(WORD color);
    static void setColor(int maunen, int mauchu);
    static void ThanhNgang(int x, int y, int dodai, int maunen, int mauchu, int makitu);
    static void ThanhTru(int x, int y, int dodai, int maunen, int mauchu, int makitu);
    static void BangGioiThieuTacGia();
    static void khungtrong(int y);
    static void duongngang(int x, int y, int dodai);
    static void manhinhdanh sach();
    static void manhinhhinh();
};
```


File Console.cpp:

```
#include <iostream>

#include <windows.h>

#include <iomanip>

#include <conio.h>

#include "Console.h"

using namespace std;

void Console::gotoxy(short x, short y)
{
    HANDLE hConsoleOutput;

    COORD Cursor_an_Pos = {x, y};

    hConsoleOutput = GetStdHandle(STD_OUTPUT_HANDLE);

    SetConsoleCursorPosition(hConsoleOutput, Cursor_an_Pos);
}

void Console::SetBGColor(WORD color)
{
    HANDLE hConsoleOutput;

    hConsoleOutput = GetStdHandle(STD_OUTPUT_HANDLE);


    CONSOLE_SCREEN_BUFFER_INFO screen_buffer_info;

    GetConsoleScreenBufferInfo(hConsoleOutput, &screen_buffer_info);


    WORD wAttributes = screen_buffer_info.wAttributes;

    color &= 0x000f;

    color <<= 4;

    wAttributes &= 0xff0f;

    wAttributes |= color;
}
```

```
    SetConsoleTextAttribute(hConsoleOutput, wAttributes);
}

void Console::textcolor(WORD color)
{
    HANDLE hConsoleOutput;

    hConsoleOutput = GetStdHandle(STD_OUTPUT_HANDLE);

    CONSOLE_SCREEN_BUFFER_INFO screen_buffer_info;
    GetConsoleScreenBufferInfo(hConsoleOutput, &screen_buffer_info);

    WORD wAttributes = screen_buffer_info.wAttributes;

    color &= 0x000f;
    wAttributes &= 0xffff;
    wAttributes |= color;

    SetConsoleTextAttribute(hConsoleOutput, wAttributes);
}

void Console::setColor(int maunen, int mauchu)
{
    SetBGColor(maunen);
    textcolor(mauchu);
}

void Console::ThanhNgang(int x, int y, int dodai, int maunen, int mauchu, int makitu)
{
    setColor(maunen, mauchu);
    for (int i = x; i < x + dodai; i++)
```

```
{
    gotoxy(i, y);
    cout << char(makitu);
}
}

void Console::ThanhTru(int x, int y, int dodai, int maunen, int mauchu, int makitu)
{
    setColor(maunen, mauchu);
    for (int j = y; j < y + dodai; j++)
    {
        gotoxy(x, j);
        cout << char(makitu) << char(makitu);
    }
}

void Console::BangGioiThieuTacGia()
{
    ThanhNgang(1, 8, 132, 7, 14, 177);
    setColor(0, 14);
    gotoxy(50, 10);
    cout << "PBL2- DU AN CO SO LAP TRINH" << endl;
    gotoxy(48, 11);
    cout << "XAY DUNG HE THONG QUAN LY THU VIEN" << endl;
    gotoxy(10, 15);
    cout << "* GVHD : Nguyen Van Nguyen";
    gotoxy(65, 15);
    cout << "* SV : Ly Hoang Quyen(22T_DT5)";
    gotoxy(65, 17);
```

```
cout << "    Nguyen Thi Ngoc Anh(22T_Nhat2)";

ThanhNgang(1, 19, 132, 7, 14, 177);

_getch();

system("cls");

}

void Console::khungtrong(int y)

{

    if (y > 0)

    {

        ThanhNgang(0, y, 133, 2, 9, 177);

        ThanhTru(0, 0, 36, 2, 9, 177);

        ThanhTru(131, 0, 36, 2, 9, 177);

    }

    else

    {

        ThanhTru(0, 0, 2, 2, 9, 177);

        ThanhTru(131, 0, 2, 2, 9, 177);

    }

    setColor(0, 14);

}

void Console::duongngang(int x, int y, int dodai)

{

    gotoxy(x, y);

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

        cout << "-";

    cout << endl;

}
```

```
void Console::manhinhdanh sach()
{
    setColor(0,14);
    system("cls");
    ThanhNgang(0, 0, 132, 2, 9, 177);
    setColor(0,14);
    char date[10];
    gotoxy(120, 1);
    cout << _strdate(date) << endl;
    gotoxy(50, 3);
    cout << "*** QUAN LY SACH THU VIEN ***";

}

void Console::manhinhchinh()
{
    system("cls");
    ThanhNgang(0, 0, 131, 2, 9, 177);
    ThanhTru(0, 0, 35, 2, 9, 177);
    ThanhTru(131, 0, 35, 2, 9, 177);
    ThanhTru(45, 5, 35, 2, 9, 177);
    ThanhNgang(0, 5, 133, 2, 9, 177);
    ThanhNgang(0, 35, 133, 2, 9, 177);
    setColor(0, 14);
    char date[10];
    gotoxy(120, 1);
    cout << _strdate(date) << endl;
```

```
gotoxy(50, 2);

cout << "TRUONG DAI HOC BACH KHOA";

gotoxy(49, 3);

cout << "*** QUAN LY SACH THU VIEN ***";

gotoxy(3, 7);

cout << "BANG LUA CHON";

gotoxy(3, 8);

cout << "1. Cap nhat them sach";

gotoxy(3, 9);

cout << "2. Tim sach theo ten";

gotoxy(3, 10);

cout << "3. Liet ke sach hien co trong thu vien";

gotoxy(3, 11);

cout << "4. Thoat";

gotoxy(3, 12);

cout << "Nhan lua chon (1-4) : ";

}
```

File soluong.txt:

150 45
54 29
536 149
84 12
800 440
430 319
778 98
787 433
547 519
706 447
924 620
966 47
252 180
202 60
548 393
249 90
249 143
460 53
58 16
443 397
341 335
947 560
630 466
92 6
154 65
9 5
630 380
495 161
815 302
59 41

File tensach.txt:

Sach1
Sach2
Sach3
Sach4
Sach5
Sach6
Sach7
Sach8
Sach9
Sach10
Sach11
Sach12
Sach13
Sach14
Sach15
Sach16
Sach17
Sach18
Sach19
Sach20
Sach21
Sach22
Sach23
Sach24
Sach25
Sach26
Sach27
Sach28
Sach29
Sach30