**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TPHCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

---o0o---



**BÀI TẬP CUỐI KỲ: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

**ĐỀ TÀI 15**

**Tìm hiểu List, Array List trong C#: Cấu trúc dữ liệu, diễn giải các thao tác cơ bản của List và Array List tương ứng trong C#.**

**Cài đặt các ví dụ minh họa cho các thao tác đó trên C#.**

**Viết chương trình trên ngôn ngữ C/C++ để thực hiện**

**Bài toán Quản lý Sách trong thư viện.**

***TP. Hồ Chí Minh, tháng 5 năm 2023***

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TPHCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

A picture containing symbol, logo, trademark, text

Description automatically generated---o0o---

**BÀI TẬP CUỐI KỲ: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

**ĐỀ TÀI 15:**

**Tìm hiểu List, Array List trong C#: Cấu trúc dữ liệu, diễn giải các thao tác cơ bản của List và Array List tương**

**ứng trong C#. Cài đặt các ví dụ minh họa cho các thao tác đó trên C#. Viết chương trình trên ngôn ngữ C/C++ để thực hiện**

**Bài toán Quản lý Sách trong thư viện.**

**Nhóm SV thực hiện:** 06

**GVHD:** Nguyễn Thị Hồng Tú

**Thành viên:**

1. Nguyễn Thị Ngọc Thanh
2. Đỗ Lê Phương Quỳnh
3. Trương Mỹ Hoa

***TP. Hồ Chí Minh, tháng 5 năm 2023***

**LỜI CẢM ƠN**

Đầu tiên, nhóm chúng em xin chân thành cảm ơn các thầy cô trong khoa Công Nghệ Thông Tin trường Đại Học Công Nghiệp Thực Phẩm đã truyền đạt những kiến thức chuyên ngành để giúp chúng em có thể tìm hiểu rõ hơn về đề tài mà mình ngiên cứu cũng như các kiến thức nền tảng để phát triển vấn đề sâu hơn. Trong quá trình thực hiện đề tài “*Tìm hiểu List, Array List trong C#: Cấu trúc dữ liệu, diễn giải các thao tác cơ bản của List và Array List tương ứng trong C#. Cài đặt các ví dụ minh họa cho các thao tác đó trên C#. Viết chương trình trên ngôn ngữ C/C++ để thực hiện Bài toán Quản lý sách tại thư viện HUFI*”, không thể tránh khỏi những thiếu sót, hạn chế về thời gian và công cụ, chúng em mong các bạn và thầy cô góp ý để nhóm phát triển thêm.

Cuối cùng, chúng em đặc biệt xin bày tỏ lòng biết ơn sâu sắc tới cô Nguyễn Thị Hồng Tú – người đã tận tình hướng dẫn và chỉ bảo cho chúng em về kiến thức về môn “Cấu trúc dữ liệu và giải thuật” để hoàn thành báo cáo này!

**Chúng em chân thành cảm ơn!**

MỤC LỤC

[*1.LIST TRONG C# LÀ GÌ ?* 5](#_Toc134341810)

[1.1 Các hàm của list. 5](#_Toc134341811)

[1.2 Các phương thức của List: 6](#_Toc134341812)

[*2.Cơ chế hoạt động của List* 7](#_Toc134341813)

[*3. Các thao tác trên List* 7](#_Toc134341814)

[3.1 .Khởi tạo list 7](#_Toc134341815)

[3.2 Thêm phần tử vào list 8](#_Toc134341816)

[3.3 Xóa phần tử ra khỏi list 8](#_Toc134341817)

[3.4 Tìm kiếm thông tin trong List 9](#_Toc134341818)

[3.5 Duyệt theo List 10](#_Toc134341819)

[CHƯƠNG 2: ARRAY LIST 11](#_Toc134341820)

[*1.Array List trong C# là gì ?* 11](#_Toc134341821)

[1.1. Hàm Array List 11](#_Toc134341822)

[1.2.Các phương thức trong Array List 12](#_Toc134341823)

[*2.Cơ chế hoạt động của Array List* 13](#_Toc134341824)

[*3.Các thao tác của Array List trong c#* 13](#_Toc134341825)

[3.1.Khởi Tạo Array List 13](#_Toc134341826)

[3.2 Thêm các phần tử trong Array List 14](#_Toc134341827)

[3.3 Xóa phần tử khỏi ArrayList 15](#_Toc134341828)

[3.4 Thêm phần tử vào ArrayList 16](#_Toc134341829)

[3.5 Duyệt theo ArrayList 16](#_Toc134341830)

[CHƯƠNG 3: CHƯƠNG TRÌNH QUẢN LÝ SÁCH 17](#_Toc134341831)

[*1.Lí do chọn đề tài.* 17](#_Toc134341832)

[*2.Mục tiêu* 17](#_Toc134341833)

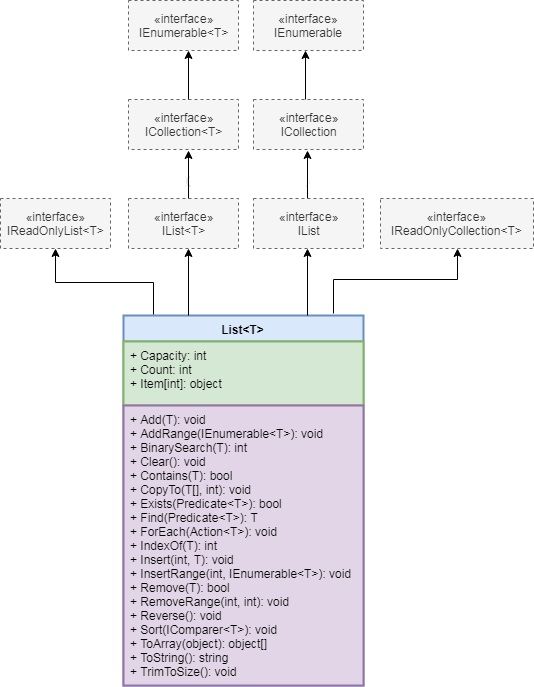
[*3. Phương thức thực hiện* 17](#_Toc134341834)

[4. *Một số chức năng cơ bản trong quản lí sách trong thư viện.* 18](#_Toc134341835)

CHƯƠNG 1: LIST

## *1.LIST TRONG C# LÀ GÌ ?*

* List là một cấu trúc dữ liệu trong C# được sử dụng để lưu trữ và quản lí một tập hợp các phần tử.
* List là một kiểu generic, tức là bạn có thể tạo một List chứa các đối tượng của bất kì kiểu dữ liệu nào.
* List được sử dụng rộng rãi trong lập trình để lưu trữ và xử lý một tập hợp các đối tượng, ví dụ như lưu trữ danh sách các mục trong một ứng dụng, hoặc lưu trữ danh sách các mục trong một ứng dụng, hoặc lưu trữ các bản ghi trong cơ sỡ dữ liệu.



### 1.1 Các hàm của list.

#### 1.1.1 Cấu tạo của List:

* List là một lớp được định nghĩa trong hệ thống namespace System.Collections.Generic trong C#.
* List cũng là một đối tượng được sữ dụng để lưu trữ danh sách các phần tử.

#### 1.1.2 Các thuộc tính của List:

| **Thuộc tính** | **Mô tả** |
| --- | --- |
| Items | Nhận hoặc đặt phần tử ở chỉ mục được chỉ định |
| Count | Trả về tổng số phần tử trong List |

### 1.2 Các phương thức của List:

| **Phương pháp** | **Mô tả** | |
| --- | --- | --- |
| Add() | Thêm một phần tử vào cuối Danh sách <T>. | |
| AddRange() | | Thêm các phần tử của tập hợp được chỉ định vào cuối Danh sách <T>. |
| BinarySearch() | | Tìm kiếm phần tử và trả về một chỉ mục của phần tử. |
| Clear() | | Loại bỏ tất cả các phần tử khỏi Danh sách <T>. |
| Contains() | | Kiểm tra xem phần tử được chỉ định có tồn tại hay không trong Danh sách <T>. |
| Find() | | Tìm phần tử đầu tiên được chỉ định. |
| Foreach() | | Lặp lại qua một danh sách <T>. |
| Insert() | | Chèn một phần tử tại chỉ mục được chỉ định trong Danh sách <T>. |
| InsertRange() | | Chèn các phần tử của một tập hợp khác vào chỉ mục được chỉ định. |
| Remove() | | Loại bỏ sự xuất hiện đầu tiên của phần tử được chỉ định. |
| RemoveAt() | | Loại bỏ phần tử tại chỉ mục được chỉ định. |
| RemoveRange() | | Loại bỏ tất cả các phần tử. |
| Sort() | | Sắp xếp tất cả các phần tử. |
| TrimExcess() | | Đặt dung lượng thành số phần tử thực tế. |
| TrueForAll | | Xác định xem mọi phần tử trong Danh sách <T> có khớp với các điều kiện được xác định bởi vị từ được chỉ định hay không. |

## *2.Cơ chế hoạt động của List*

* List không được triễn khai bằng mảng tĩnh mà được triễn khai bằng danh sách liên kết đôi(doubly linked list) động (Dynamic List).
* Khi thêm một phần tử mới vào danh sách , List sẽ thêm phần tử đó vào cuối danh sách chứ không phải cuối mảng .
* Khi loại bỏ một phần tử khỏi List, các phần tử nằm sau phần tử lợi bỏ sẽ được dịch chuyển về phía trước để điền vào khoảng trống do phần tử đã tự loại bỏ.

## *3. Các thao tác trên List*

### 3.1 .Khởi tạo list

* List chỉ chứa các phần tử cùng kiểu dữ liệu, do đó khi khai báo kiểu dữ liệu của các phần tử được khai báo trước.
* Các phần tử của List được lưu trữ liên tiếp trong bộ nhớ, do đó việc truy cập phần tử là nhanh và hiệu quả.
* List cũng có thể mở rộng kích thước của nó tự động khi bạn thêm phần tử mới vào danh sách.
* Khởi tạo biến List<T> trong C#
* Để sử dụng được List<T> đầu tiên các bạn cần import thư viện

using System.Collections.Generic;

**Ví dụ:**



### 3.2 Thêm phần tử vào list

* Để thêm tất cả các phần tử từ một Array hoặc một List khác ta sử dụng phương thức AddRange().

**Ví dụ:**

**A picture containing text, screenshot, font

Description automatically generated**

* Nếu muốn thêm phần tử vào vị trí cụ thể của List, ta sữ dụng phương thức Insert:



* Index là vị trí mà bạn muốn chèn phần tử vào.
* Item là phần tử mà bạn muốn thêm vào List.



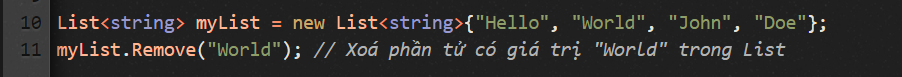
### 3.3 Xóa phần tử ra khỏi list

- Sử dụng phương thức Remove() để xóa phần tử xuất hiện đầu tiên trong List

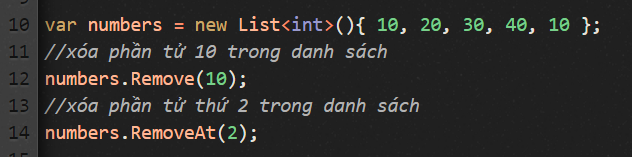
- Sử dụng phương thức RemoveAt() để xóa phần tử tại một vị trí index được chỉ định.

* Remove()

**Cú pháp:**

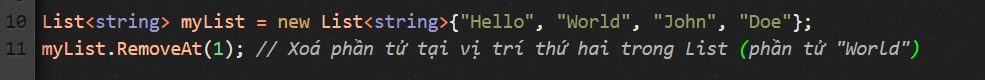
****

**Ví dụ**

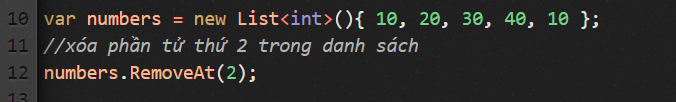
****

* RemoveAt()

**Cú pháp:**



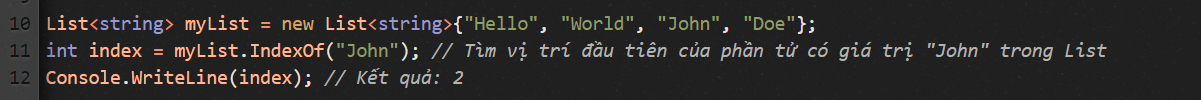
**Ví dụ**



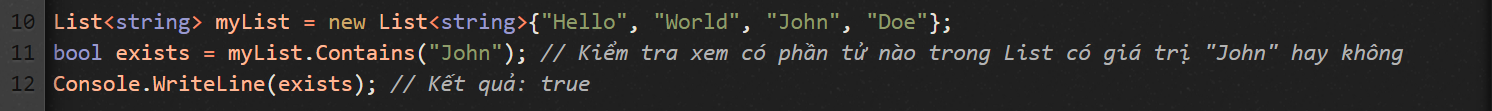
### 3.4 Tìm kiếm thông tin trong List

- Để tìm kiếm phần tử trong List, ta sữ dụng phương thức *IndexOf* hoặc *Contains.*

* Phương thức *IndexOf* sẽ trả về chỉ số đầu tiên của phần tử được tìm thấy trong List có giá trị bằng đối số truyền vào.



* Phương thức *Contains* sẽ trả về giá trị Boolean xác định có tồn tại phần tử trong List có giá trị bằng đối số truyền vào hay không.



### 3.5 Duyệt theo List

- Để duyệt các phần tử trong List, sữ dụng vòng lặp for, do-while

* Cấu trúc vòng lặp for

A picture containing text, font, multimedia software, screenshot

Description automatically generated

* Duyệt theo do-while

A picture containing text, font, screenshot

Description automatically generated

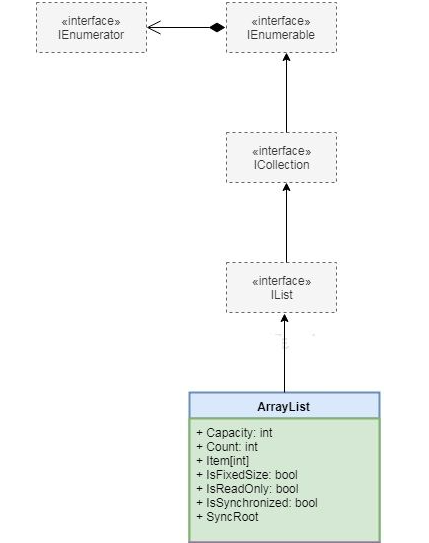
# CHƯƠNG 2: ARRAY LIST

## *1.Array List trong C# là gì ?*

* Arrylist là một kiểu dữ liệu trong C# được sữ dụng để lưu trữ các đối tượng có thể thay đổi kích thước trong quá trình thực thi của chương trình.
* ArrayList giống như một mảng động, các vị trí trong mảng có thể tự động được thay đổi tuỳ thuộc vào số lượng phần tử được thêm vào hoặc xoá ra khỏi ArrayList.
* ArrayList cũng là một kiểu dữ liệu non-generic, tức là có thể thêm bất kì loại đối tượng nào.

=>Tuy nhiên do ArrayList là kiểu dữ liệu non-generic, do đó khi sử dụng ArrayList với các đối tượng khác nhau, phải thêm các xử lý kiểm tra kiểu dữ liệu để tránh lỗi.

Sau đây là sơ đồ minh họa cho class Array List trong C#:



### 1.1. Hàm Array List

#### 1.1.1 Cấu tạo của Array List

* + ArrayList một lớp được định nghĩa trong hệ thống namespace System.Collections trong C#.

#### 1.1.2 Các thuộc tính cơ bản của Array List

| **Thuộc tính** | **Mô tả** |
| --- | --- |
| Capaty | Lấy hoặc đặt số phần tử mà Array List có thể chứa |
| Count | Đếm số phần tử thực sự có trong Array List |
| IsFixedSize | Cho biết ArrayList có kích thước cố định hay không |
| IsReadOnly | Cho biết ArrayList có ở chế độ ReadOnly (chỉ cho đọc) hay không |
| Item | Nhận hoặc đặt phần tử ở vị trí index được chỉ định |

### 1.2.Các phương thức trong Array List

Trong ArrayList có các phương thức như sau:

| **Phương thức** | **Mô tả** |
| --- | --- |
| Add() / AddRange() | Phương thức Add () thêm các phần tử đơn lẻ vào cuối ArrayList.  Phương thức AddRange () thêm tất cả các phần tử từ bộ sưu tập được chỉ định vào ArrayList. |
| Insert() / InsertRange() | Phương thức Insert () chèn một phần tử duy nhất tại vị trí được chỉ định trong ArrayList.  Phương thức InsertRange () chèn tất cả các phần tử của tập hợp bắt đầu từ vị trí được chỉ định trong ArrayList. |
| Remove() / RemoveRange() | Phương thức Remove () loại bỏ phần tử được chỉ định khỏi ArrayList.  Phương thức RemoveRange () loại bỏ một loạt các phần tử khỏi ArrayList. |
| RemoveAt() | Loại bỏ phần tử tại vị trí được chỉ định khỏi ArrayList. |
| Sort() | Sắp xếp toàn bộ các phần tử của ArrayList. |
| Reverse() | Đảo ngược thứ tự của các phần tử trong toàn bộ ArrayList. |
| Contains() | Kiểm tra xem phần tử được chỉ định có tồn tại trong ArrayList hay không. Trả về true nếu tồn tại, ngược lại là false. |
| Clear() | Loại bỏ tất cả các phần tử trong ArrayList. |
| CopyTo() | Sao chép tất cả các phần tử hoặc phạm vi phần tử vào Array. |
| GetRange() | Trả về số phần tử được chỉ định từ vị trí được chỉ định từ ArrayList. |
| IndexOf() | Tìm kiếm phần tử được chỉ định và trả về vị trí của nó trong ArrayList. Trả về 0 nếu tìm thấy và trả về -1 nếu không tìm thấy phần tử. |
| ToArray() | Chuyển đổi ArrayList thành Array. |

## *2.Cơ chế hoạt động của Array List*

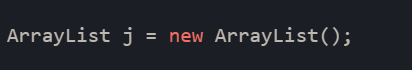
* ArrayList được triễn khai bằng cách sữ dụng một mảng động để lưu trữ các phần tử của nó.
* Khi thêm một phần tử mới vào danh sách, ArrayList sẽ tìm kiếm trong mảng để kiểm tra xem mảng có đủ chỗ trống để chứa phần tử mới hay không.Nếu mảng không còn đủ chỗ trống để chứa phần tử mới, ArrayList sẽ tạo một mảng mới với kích thước lớn hơn và sao chép các phần tử từ mảng cũ sang mảng mới .Sau đó ArrayList thêm phần tử mới vào mảng mới này và sử dụng mảng mới để thay thế cho mảng cũ.
* Khi bạn loại bỏ một phần tử khỏi ArrayList, các phần tử nằm sau phần tử loại bỏ sẽ được dịch chuyển về phía trước để điền vào khoảng trống do phần tử đã bị loại bỏ.

## *3.Các thao tác của Array List trong c#*

### 3.1.Khởi Tạo Array List

* ArrayList có thể chứa các đối tượng bất kỳ và các dữ liệu nguyên thuỷ khác nhau.
* Các phần tử của Arrylist được lưu trữ kiểu bất kì, vì vậy không cần phải khai báo trước kiểu dữ liệu.
* ArrayList có thể mở rộng kích thước của nó tự động nếu số lượng phần tử tăng lên.

Để tạo một đối tượng của ArrayList ta sử dụng:



**Ví dụ:**

A picture containing text, font, screenshot, line

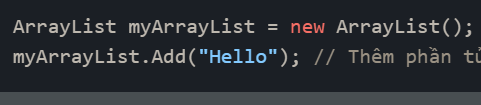
Description automatically generated

### 3.2 Thêm các phần tử trong Array List

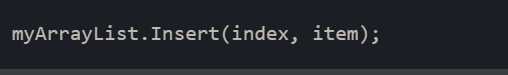
- Có thể sử dụng phương thức Add() hoặc cú pháp trình khởi tạo đối tượng để thêm một phần tử vào ArrayList.

Trong một ArrayList có thể chứa nhiều giá trị null trùng lặp nhau.

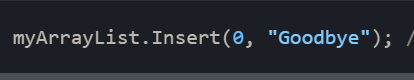
* Nếu muốn thêm phần tử ArrayList, ta sữ dụng phương thức Add:



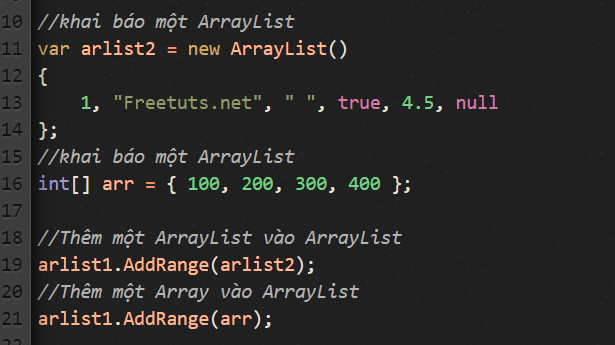
* Nếu muốn thêm phần tử vào vị trí cụ thể của List, ta sữ dụng phương thức Insert:



**Ví dụ**

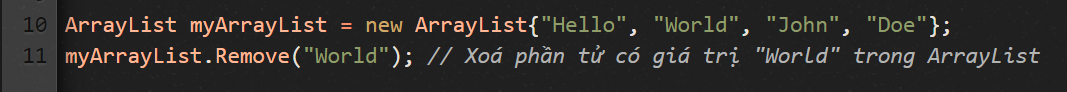


* **Thêm một Array / ArrayList vào ArrayList:**

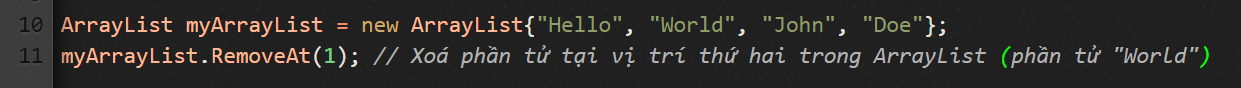


### 3.3 Xóa phần tử khỏi ArrayList

* Xoá trong ArrayList sữ dụng phương phức Remove và RemoveAt(xoá phần tử đầu tiên trong List)
* Xoá Remode.



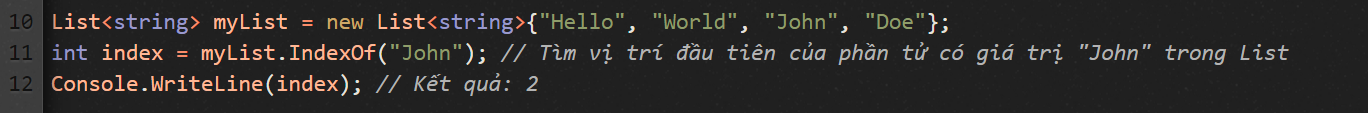
* Xoá RemoveAt là sẽ xoá phần tử trong ArrayList tại vị trí được chỉ định.



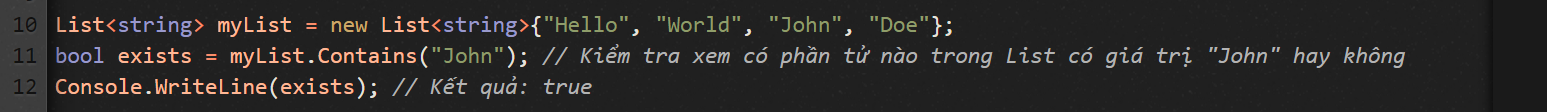
### 3.4 Thêm phần tử vào ArrayList

- Để tìm kiếm phần tử trong ArrayList, ta sữ dụng phương thức IndexOf hoặc Contains.

* Phương thức IndexOf sẽ trả về chỉ số đầu tiên của phần tử được tìm thấy trong List có giá trị bằng đối số truyền vào.



* Phương thúc Contains sẽ trả về giá trị Boolean xác định có tồn tại phần tử trong List có giá trị bằng đối số truyền vào hay không.



### 3.5 Duyệt theo ArrayList

* Để duyệt các phần tử trong ArrayList, sữ dụng vòng lặp for, do-while.
* Cấu trúc vòng lặp for

A picture containing text, screenshot, font, line

Description automatically generated

* Duyệt theo do-while

A picture containing text, font, screenshot

Description automatically generated

# CHƯƠNG 3: CHƯƠNG TRÌNH QUẢN LÝ SÁCH

## *1.Lí do chọn đề tài.*

* Quản lí sách là một lĩnh vực quan trọng trong thư viện và tổ chức giáo dục,nó liên quan đến việc lưu trữ, sắp xếp và truy xuất thông tin.Vì vậy chọn chủ đề quản lí sách là cách hiểu rõ hơn về tài liệu, cũng như quy trình, tiêu chuẩn và phương pháp.
* Quản lí sách liên quan đến việc sắp xếp thông tin và phân loại các chuỗi. Điều này giúp cải thiện khả năng tổ chức thông tin và tạo ra một cơ sỡ dữ liệu tốt hơncho các dự án lớn, nghiên cứu, phát triễn sau này.
* Chủ đề quản lí sách còn mang tính ứng dụng cao, vì liên quan đến các kĩ năng và công cụ để xay dựng và duy trì các hệ thống quản lí tài liệu.
* Cuối cùng , quản lí sách còn đóng vcai trod quan trọng trong việc giảng dạy và học tập, dặc biệt các chương trình về đào tạo thư viện giúp hiểu rỏ hơn veef quản lí thông tin.

## *2.Mục tiêu*

* *Lưu trữ và duy trì thông tin các sách* : Mục tiêu chính của quản lý dữ liệu sách là lưu trữ các thông tin về sách như tên sách , tên tác giả, năm xuất bản,….. Và duy trì thông tin để có thể truy xuất, sửa đổi và tìm kiếm thông tin tại 1 thời điểm.
* *Tăng cường khả năng tìm kiếm thông tin về s*ách thông qua tìm kiến thông tin về sách thông qua việc tìm kiếm và sắp xếp thông tin, giúp cho ta tìm kiếm và truy cập thông tin 1 cách nhanh chóng và dễ dàng.
* *Tạo ra hệ thống quản lí sách hiệu quả*: Đáng tin cậy, quản lý sách nhanh chóng và hiệu quả , đáng tin cậy giúp người dung tiết kiệm thời gian và công sức qurn lý sách.

## *3. Phương thức thực hiện*

* Linh động: Danh sách liên kết đơn là một cấu trúc linh động, cho phép thêm, xoá, nhập và sửa các thông tin liên quan đến sách một cách dễ dàng và nhanh chóng.
* Tiết kiệm bộ nhớ: Đối với những cơ sở dữ liệu lớn, việc sử dụng danh sách liên kết đơn giúp tiết kiệm bộ nhớ hơn so với mảng động, chỉ sử dụng bộ nhớ cho số lượng phần tử thực sự cần thiết.
* Dễ dàng trong việc sắp xếp thông tin: Với danh sách liên kết đơn, việc sắp xếp thông tin sách để tìm kiếm thông tin trở nên dễ dàng.
* Tính an toàn và bảo mật cao: Do danh sách liên kết đơn được thiết kế để chỉ cho phép các node/đối tượng kế tiếp tham chiếu đến node/đối tượng hiện tại, do đó danh sách liên kết đơn đảm bảo tính bảo mật tốt hơn so với mảng động.
* Khả năng truy xuất phần tử 1 cách dễ dàng: Mỗi node trong một danh sách liên kết đơn được liên kết nhau thông qua con trỏ, vì vậy việc truy xuất rất dễ dàng và hiệu quả.

## 4. *Một số chức năng cơ bản trong quản lí sách trong thư viện.*

|  |  |  |
| --- | --- | --- |
| * Thêm đầu   void insertAtHead(List\* list)  {  Sach data;  printf("\nNhap ma sach: ");  rewind(stdin);  gets\_s(data.maSach);  printf("\nNhap ten sach: ");  rewind(stdin);  gets\_s(data.tenSach);  printf("\nNhap ten tac gia: ");  rewind(stdin);  gets\_s(data.tenTacGia);  printf("\nNhap nha xuat ban: ");  rewind(stdin);  gets\_s(data.nhaXuatBan);  printf("\nNhap nam xuat ban: ");  scanf("%d", &data.namXuatBan);  printf("\nNhap the loai: ");  rewind(stdin);  gets\_s(data.theLoai);  Node\* p = createNode(data);  if (isEmpty(\*list))  {  list->head = list->tail = p;  }  else  {  p->next = list->head;  list->head = p;  }  } | * Thêm cuối   void insertAtTail(List\* list)  {  Sach data;  printf("\nNhap ma sach: ");  rewind(stdin);  gets\_s(data.maSach);  printf("\nNhap ten sach: ");  rewind(stdin);  gets\_s(data.tenSach);  printf("\nNhap ten tac gia: ");  rewind(stdin);  gets\_s(data.tenTacGia);  printf("\nNhap nha xuat ban: ");  rewind(stdin);  gets\_s(data.nhaXuatBan);  printf("\nNhap nam xuat ban: ");  scanf("%d", &data.namXuatBan);  printf("\nNhap the loai: ");  rewind(stdin);  gets\_s(data.theLoai);  Node\* p = createNode(data);  if (isEmpty(\*list))  {  list->head = list->tail = p;  }  else  {  list->tail->next = p;  list->tail = p;  }  } | * Thêm sau   void insertAfter(List\* list, char x[])  {  Sach data;  rewind(stdin);  printf("\nNhap ma sach: ");  rewind(stdin);  gets\_s(data.maSach);  printf("\nNhap ten sach: ");  rewind(stdin);  gets\_s(data.tenSach);  printf("\nNhap ten tac gia: ");  rewind(stdin);  gets\_s(data.tenTacGia);  printf("\nNhap nha xuat ban: ");  rewind(stdin);  gets\_s(data.nhaXuatBan);  printf("\nNhap nam xuat ban: ");  scanf("%d", &data.namXuatBan);  printf("\nNhap the loai: ");  rewind(stdin);  gets\_s(data.theLoai);  Node\* p = createNode(data);  if (isEmpty(\*list))  {  list->head = list->tail = p;  }  else  {  Node\* q = list->head;  while (q != NULL && strcmp(q->data.maSach, x) != 0)  {  q = q->next;  }  if (q == NULL)  {  printf("Khong tim thay sach co ma %s", x);  return;  }  else  {  p->next = q->next;  q->next = p;  }  }  } |

|  |  |  |
| --- | --- | --- |
| * Xoá đầu   void deleteAtHead(List\* list)  {  if (isEmpty(\*list))  {  printf("Danh sach rong");  return;  }  Node\* p = list->head;  list->head = list->head->next;  free(p);  printf("Xoa thanh cong\n");  } | * Xoá cuối   void deleteAtTail(List\* list)  {  if (isEmpty(\*list))  {  printf("Danh sach rong");  return;  }  Node\* p = list->head;  while (p->next != list->tail)  {  p = p->next;  }  p->next = NULL;  free(list->tail);  list->tail = p;  printf("Xoa thanh cong\n");  } | * Xoá sau   void deleteAfter(List\* list, char x[])  {  if (isEmpty(\*list))  {  printf("Danh sach rong");  return;  }  Node\* p = list->head;  while (p != NULL && strcmp(p->data.maSach, x) != 0)  {  p = p->next;  }  if (p == NULL)  {  printf("Khong tim thay sach co ma %s", x);  return;  }  else  {  Node\* q = list->head;  while (q->next != p)  {  q = q->next;  }  q->next = p->next;  free(p);  printf("Xoa thanh cong\n");  }  } |

|  |  |  |
| --- | --- | --- |
| * Sắp xếp   void sortByTheLoai(List& list)  {  if (isEmpty(list))  {  printf("Danh sach rong");  return;  }  Node\* p = list.head;  while (p != NULL)  {  Node\* q = p->next;  while (q != NULL)  {  if (strcmp(p->data.theLoai, q->data.theLoai) > 0)  {  Sach temp = p->data;  p->data = q->data;  q->data = temp;  }  if (strcmp(p->data.theLoai, q->data.theLoai) == 0)  {  if (strcmp(p->data.maSach, q->data.maSach) > 0)  {  Sach t = p->data;  p->data = q->data;  q->data = t;  }  }  q = q->next;  }  p = p->next;  }  } | * Thống kê   void thongKe(List list)  {  if (isEmpty(list))  {  printf("Danh sach rong");  return;  }  Node\* p = list.head;  while (p != NULL)  {  int count = 0;  Node\* q = list.head;  while (q != NULL)  {  if (strcmp(p->data.tenTacGia, q->data.tenTacGia) == 0 && strcmp(p->data.nhaXuatBan, q->data.nhaXuatBan) == 0)  {  count++;  }  q = q->next;  }  printf("\n%s co %d quyen sach o %s \n", p->data.tenTacGia, count, p->data.nhaXuatBan);  p = p->next;  }  } | * Tìm sách   void findNXB(List list)  {  if (isEmpty(list))  {  printf("Danh sach rong");  return;  }  Node\* p = list.head;  int min = 1000;  while (p != NULL)  {  int count = 0;  Node\* q = list.head;  while (q != NULL)  {  if (strcmp(p->data.nhaXuatBan, q->data.nhaXuatBan) == 0)  {  count++;  }  q = q->next;  }  if (count < min)  {  min = count;  }  p = p->next;  }  printf("\nNha xuat ban co it sach nhat la: \n");  p = list.head;  while (p != NULL)  {  int count = 0;  Node\* q = list.head;  while (q != NULL)  {  if (strcmp(p->data.nhaXuatBan, q->data.nhaXuatBan) == 0)  {  count++;  }  q = q->next;  }  if (count == min)  {  printf("%s\n ", p->data.nhaXuatBan);  }  p = p->next;  }  } |

**Tài liệu tham khảo**

# 1.Nguyễn Thảo**,** [C#] Hướng ẫn sử dụng List trong lập trình csharp, Học lập trình VB.NET, [https://laptrinhvb.net/bai-viet/chuyen-de-csharp/---Csharp----Huong-dan-su-dung-List%3CT%3E--trong-lap-trinh-csharp/73340d5d2cb7f55a.html[Truy](https://laptrinhvb.net/bai-viet/chuyen-de-csharp/---Csharp----Huong-dan-su-dung-List%3CT%3E--trong-lap-trinh-csharp/73340d5d2cb7f55a.html%5bTruy) [Truy cập 13/4/2023]

# 2. Cách dùng List trong C#, freetuts.net,<https://freetuts.net/list-trong-c-sharp-3557.html> [Truy cập 11/4/2023]

# 3. Lớp List - ArrayList trong C#, vietJack.com,

<https://vietjack.com/csharp/lop_arraylist_trong_csharp.jsp> [Truy cập 15/4/2023]

# 4.Danh sách (list) trong C#: ArrayList, List, SortedList, tuhocict,

/<https://tuhocict.com/danh-sach-list-trong-c-arraylist-list-sortedlist/>[Truy cập 15/4/2023]