**Trường Đại Học Công Nghiệp Thực Phẩm TP.HCM**

**Khoa Công Nghệ Thông Tin**



**BÁO CÁO MÔN HỌC**

**CẤU TRÚC DỮ LIỆU**

**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 trong thư viện.**

Giảng viên hướng dẫn: **Huỳnh Thị Châu lan**

Sinh viên thực hiện: **Nguyễn Hoàng Kha** MSSV: **2001190115**

**Nguyễn Minh Triết** MSSV: **2001190306**

**Nguyễn Trọng Liêm** MSSV: 2**001191021**

**Lê Hoàng Nhựt** MSSV: **2001190723**

**Nguyễn Quốc Hoàng Sơn** MSSV: **2001190787**

TP. Hồ Chí Minh, ngày 10 tháng 6 năm 2016

***Lời mở đầu***

**MỤC LỤC**

[**I. List, ArrayList** 3](#_Toc91015216)

[**1.** **Cấu tạo của List, ArrayList** 3](#_Toc91015217)

[**2.** **Cơ chế hoạt động của List, ArrayList** 4](#_Toc91015218)

[**3.** **Khởi tạo List, ArrayList** 8](#_Toc91015219)

[**4.** **Thêm, xóa phần tử trong List, ArrayList**  9](#_Toc91015220)

[**5.** **Tìm kiếm thông tin trong List ArrayList** 12](#_Toc91015221)

[**6.** **Duyệt các phần tử trong List, ArrayList và các thao tác khác** 15](#_Toc91015222)

[**II. Chương trình minh họa bằng C#** 17](#_Toc91015223)

[**III. Chương trình quản lý dữ liệu sách bằng ngôn ngữ C/C++** 24](#_Toc91015229)

Ngày nay, cùng với sự vươn lên của đất nước, hòa vào đó Công nghệ thông tin là một ngành đã có những bước phát triển mạnh mẽ không ngừng và nó đã trở thành chiếc chìa khóa dẫn đến thành công cho nhiều cá nhân, tổ chức. Với những ứng dụng của mình, công nghệ thông tin đã góp phần mang lại nhiều lợi ích mà không ai có thể phủ nhận được. Đặc biệt là trong lĩnh vực quản lý kinh tế, công nghệ thông tin đã góp phần tạo ra sự thay đổi nhanh chóng cho bộ mặt xã hội.

Trong quá trình thực hiện đề tài nhóm đã gặp phải không ít khó khăn và trở ngại, tuy nhiên, với sự giúp đỡ của bạn bè, cùng sự hướng dẫn tận tình, chỉ bảo cặn kẽ của giảng viên **Huỳnh Thị Châu Lan**, cuối cùng chúng em cũng đã cố gắng hoàn thành đề tài này với tất cả sự cố gắng và nỗ lực. Chúng em xin được gửi lời cảm ơn tới những người bạn đã góp ý và giúp đỡ nhóm, và đặc biệt là cảm ơn cô, người đã giúp chúng tôi xác định được những yêu cầu và mục tiêu đề tài, truyền dạy các kĩ năng nghiệp vụ cũng như hướng dẫn cho chúng tôi cách thức thực hiện đề tài..

Do kiến thức thực tế và hiểu biết còn hạn chế nên trong quá trình hoàn thành báo cáo không tránh khỏi sai sót, nhóm chúng em rất mong sự góp ý đóng góp của thầy và các bạn để nhóm có thể hoàn thiện bài tập lớn được tốt hơn.

Chúng em xin chân thành cảm ơn!

1. **List, ArrayList**
2. **Cấu tạo của List, ArrayList**
3. **List**

List trong C# là một Generic Collections giúp lưu trữ và quản lý một danh sách các đối tượng theo kiểu mảng (truy cập các phần tử bên trong thông qua chỉ số **index**).

Để sử dụng các Collections trong .NET ta cần thêm thư viện System.Collections.**Generic** bằng câu lệnh:

using System.Collections.**Generic**;

Vì List là một lớp nên trước khi sử dụng ta cần khởi tạo vùng nhớ bằng toán tử new:

/ khởi tạo 1 List các số nguyên rỗng

List<**int**> MyList = **new** List<**int**>();

Bạn cũng có chỉ định sức chứa (Capacity) ngay lúc khởi tạo bằng cách thông qua constructor được hỗ trợ sẵn:

// khởi tạo 1 List các số nguyên và chỉ định Capacity ban đầu là 5

List<**int**> MyList2 = **new** List<**int**>(5);

Ngoài ra bạn cũng có thể khởi tạo 1 List chứa các phần tử được sao chép từ một Generic Collections khác (lưu ý là có cùng kiểu dữ liệu truyền vào):

/\*

\* Khởi tạo 1 List số nguyên có kích thước bằng với MyList2.

\* Sao chép toàn độ phần tử trong MyList2 vào MyList3.

\*/

List<**int**> MyList3 = **new** List<**int**>(MyList2);

1. **ArrayList**

**ArrayList** trong C#:

* Là một Collections giúp lưu trữ và quản lý một danh sách các đối tượng theo kiểu mảng (truy cập các phần tử bên trong thông qua chỉ số *index*).
* Rất giống mảng các object nhưng có thể thêm hoặc xoá các phần tử một cách linh hoạt và có thể tự điều chỉnh kích cỡ một cách tự động.

Để sử dụng các Collections trong **.NET** ta cần thêm thư viện System.Collections bằng câu lệnh:

using **System.Collections**;

Vì ArrayList là một lớp nên trước khi sử dụng ta cần khởi tạo vùng nhớ bằng toán tử new:

// khởi tạo 1 ArrayList rỗng

ArrayList MyArray = **new** ArrayList();

Bạn cũng có chỉ định sức chứa (Capacity) ngay lúc khởi tạo bằng cách thông qua **constructor**được hỗ trợ sẵn:

// khởi tạo 1 ArrayList và chỉ định Capacity ban đầu là 5

ArrayList MyArray2 = **new** ArrayList(5);

Ngoài ra bạn cũng có thể khởi tạo 1 ArrayList chứa các phần tử được sao chép từ một Collections khác:

/\*

\* Khởi tạo 1 ArrayList có kích thước bằng với MyArray2.

\* Sao chép toàn độ phần tử trong MyArray2 vào MyArray3.

\*/

ArrayList MyArray3 = **new** ArrayList(MyArray2);

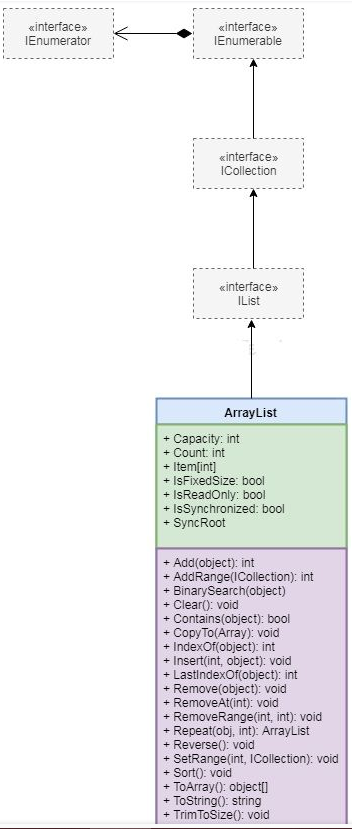
1. **Cơ chế hoạt động của List, ArrayList**
2. **List**

Diagram

Description automatically generated

* List tương đương với ArrayList.
* Thuộc System.Collections.Generic namespace.
* Có thể chứa các dữ liệu thuộc kiểu dữ liệu chỉ định.
* Các phần tử có thể thêm bằng phương thức Add(), AddRange() hoặc bằng phương thức khởi tạo.
* Các phần tử được truy cập bằng chỉ mục (ví dụ myList[0], truy cập phần tử thứ 0).
* List hoạt động nhanh hơn và ít lỗi hơn ArrayList.

1. **ArrayList**



Như bạn có thể nhìn thấy từ biểu đồ trên, lớp ArrayList triển khai thực hiện các interface IEnumerable, ICollectionvà IList. Vì vậy, bạn có thể tạo một đối tượng của lớp ArrayList và gán nó cho bất kỳ biến nào có kiểu dữ liệu là những interface trên.

Tuy nhiên, nếu bạn gán nó cho biến có kiểu dữ liệu là interface IEnumerable hoặc ICollection thì bạn sẽ không thể thêm các phần tử vào ArrayList và truy cập ArrayList theo chỉ mục.

1. **Tổng kết chung về cách hoạt động của List, ArrayList**

Bên trong, List và ArrayList hoạt động bằng cách duy trì một mảng đối tượng bên trong, được thay thế bằng một mảng lớn hơn khi đạt đến dung lượng. Việc thêm các phần tử là hiệu quả (vì thường có một chỗ trống ở cuối), nhưng việc chèn các phần tử có thể chậm (vì tất cả các phần tử sau điểm chèn phải được dịch chuyển để tạo ra một chỗ trống). Cũng như với mảng, tìm kiếm hiệu quả nếu phương pháp BinarySearch được sử dụng trên list đã được sắp xếp, nhưng nếu không thì không hiệu quả vì mỗi mục phải được kiểm tra riêng lẻ.

1. **Khởi tạo List, ArrayList**
2. **List**

Để tạo một List ta sử dụng từ khóa new cùng với đó là kiểu dữ liệu cho tham số cần lưu trữ.

Text

Description automatically generated

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

Text

Description automatically generated

1. **ArrayList**

Arayy list cung cấp 3 overload để khởi tạo object

Graphical user interface

Description automatically generated with medium confidence

Cách đơn giản nhất là dùng overload không tham số.

Overload thứ hai nhận một tập hợp giá trị bất kỳ làm tham số. Khi khởi tạo, các phần tử của tập hợp này sẽ được sao chép sang và trở thành phần tử của arraylist.

Overload thứ ba nhận một số nguyên làm tham số. Số nguyên này thể hiện dung lượng (capacity) tạm thời của arraylist, nghĩa là số phần tử có thể chứa được. Lưu ý rằng, không giống với mảng, dung lượng của arraylist có thể tiếp tục tăng lên theo nhu cầu. Số phần tử thực chứa trong arraylist luôn nhỏ hơn hoặc bằng với dung lượng.

1. **Thêm, xóa phần tử trong List, ArrayList.**
2. **List**

**Thêm:**

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

Text

Description automatically generated

**Xóa:**

Sử dụng phương thức Remove() để xóa phần tử xuất hiện đầu tiên trong List. Hoặc sử dụng phương thức RemoveAt() để xóa phần tử tại một vị rí index được chỉ định.

Text

Description automatically generated

**Cú pháp:**

Text

Description automatically generated

1. **ArrayList**

**Thêm:**

Để thêm phần tử vào cuối danh sách có thể dùng hai phương thức: Add để thêm từng phần tử đơn; AddRange để thêm danh sách phần tử. AddRange nhận tham số là một mảng các object.

Graphical user interface, text

Description automatically generated

ArrayList chấp nhận kiểu của phần tử là object . Điều này có nghĩa là nó chấp nhận phần tử thuộc tất cả các kiểu dữ liệu của C#.

Trong C#, lớp object (hoặc Object ) là lớp tổ tổng của tất cả các lớp khác, kể cả lớp do người dùng xây dựng. Theo cơ chế đa hình và kế thừa, một đối tượng tạo ra từ lớp con cũng được xem là đối tượng của lớp cha (biến thuộc kiểu con đồng thời cũng là biển thuộc kiểu cha). Do vậy, bất kỳ đối tượng nào trong C# cũng đồng thời là đối tượng của lớp object. Nếu phương thức chấp nhận tham số thuộc kiểu object, nó chấp nhận tham số thuộc bất kỳ kiểu nào của C#.

Chúng ta cũng có thể chèn phần tử vào vị trí bất kỳ trong danh sách với phương thức Insert hoặc InsertRange

Text

Description automatically generated with low confidence

Khi chèn vào một vị trí, tất cả các phần tử đứng sau sẽ bị dồn về cuối danh sách.

**Xóa:**

Để xóa phần tử khỏi danh sách có thể sử dụng một trong các phương thức: Remove,

RemoveAt, RemoveRange.

Graphical user interface, text

Description automatically generated with medium confidence

• Remove sẽ xóa phần tử đầu tiên trong danh sách có giá trị trùng với giá trị cung cấp.

• RemoveAt xóa bỏ phần tử ở một vị trí xác định.

• RemoveRange xóa bỏ một nhóm phần tử.

Ngoài ra, để xóa bỏ tất cả các phân tử, có thể dùng phương thức Clear.

1. **Tìm kiếm thông tin trong List, ArrayList.**
2. **List**

Một số phương thức cho phép tìm kiếm, tra cứu vị trị trí các phần tử trong List

|  |  |
| --- | --- |
| **Phương thức** | **Mô tả** |
| **IndexOf (obj)** | **Tìm Index của đối tượng trong List** |
| **LastIndexOf (obj)** | **Tìm Index của phần tử cuối cùng có giá trị bằng obj trong List** |
| **FindIndex** | **Tìm kiếm trả về Index** |
| **FindLastIndex** | **Tìm kiếm trả về Index cuối** |
| **Find (Predicate)** | **Tìm kiếm trả về phần tử** |
| **FindAll (Predicate)** | **Tìm kiếm trả về danh sách phần tử** |
| **FindLast** | **Tìm kiếm trả về phần tử cuối tìm thấy** |

Trong các phương thức trên, có các phương thức ví dụ Find, chứa tham số là delegate bool Predicate<in T>(T obj); nó là hàm callback, trả về true là phần tử phù hợp trả về .

Ví dụ sau đây là một Delegate phù hợp gán cho tham số Predicate

// Delegate trả về true khi tên bằng "Galaxy Note 10"

(Product ob) => {

return (ob.Name == "Galaxy Note 10");

}

Đoạn mã này có thể làm tham số cho Find, FindAll ...

Product foundpr1 = products.Find(

**(Product ob) => { return (ob.Name == "Galaxy Note 10");}**

);

if (foundpr1 != null)

Console.WriteLine("(found) " + foundpr1.ToString("O"));

// (found) Xuất xứ: Việt Nam - Tên: Galaxy Note 10 - Giá: 700 - ID: 5

Các delegate cũng có thể viết gọn lại

// tìm index của đối tượng có xuất xứ là "Trung Quốc"

var ifound = products.FindIndex(x => x.Origin == "Trung Quốc");

// tìm các sản phẩm có giá trên 100

List<Product> p\_100 = products.FindAll(product => product.Price > 100);

Nếu muốn tùy biến cao hơn Delegate, để tìm kiếm theo tham số tùy chọn, bạn có thể để Delegate trên vào lớp chức năng, ví dụ xây dựng lớp SearchNameProduct

public class SearchNameProduct {

string namesearch;

public SearchNameProduct(string name) {

namesearch = name;

}

// Hàm gán cho delegage

public bool search(Product p) {

return p.Name == namesearch;

}

}

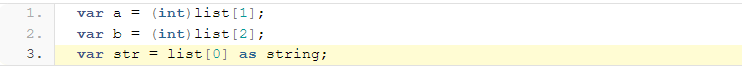
Thực hiện tìm kiếm, ví dụ

Product pr1 = products.Find( (new SearchNameProduct("Galaxy Note 10")).search); // Tìm sản phẩm có tên Galaxy Note 10

Product pr2 = products.Find( (new SearchNameProduct("IPhone 12 pro max")).search); // Tìm sản phẩm có tên Iphone 12 ProMax

1. **ArrayList**

Để tìm kiếm từng phân tử, chúng ta dùng pháp toàn chỉ số (Index operator) tương tự mảng:



Sự khác biệt ở chỗ, khi đọc phần tử của danh sách, chúng ta đồng thời phải thực hiện ép kiểu (type casting) từ kiểu object sang kiểu cụ thể khác. Trong ví dụ trên, chúng ta ép từ object sang int và string.

**Ví dụ:**

ArrayList myArryList = new ArrayList();

myArryList.Add(1);

myArryList.Add("Two");

myArryList.Add(3);

myArryList.Add(4.5f);

//Chuyển sang dữ liệu thích hợp

int firstElement = (int) myArryList[0]; //returns 1

string secondElement = (string) myArryList[1]; //returns "Two"

int thirdElement = (int) myArryList[2]; //returns 3

float fourthElement = (float) myArryList[3]; //returns 4.5

//sử dụng var

var firstElement = myArryList[0]; //returns 1

Sử dụng vòng lặp foreach hoặc vòng lặp for để duyệt ArrayList.

IList myArryList = new ArrayList() { 1, "Two", 3, 4.5F };

foreach (var item in myArryList)

{

Console.WriteLine(item);

}

// Hoặc

// for (int i = 0 ; i< myArryList.Count; i++)

// {

// Console.WriteLine(myArryList[i]);

// }

Kết quả khi biên dịch và chạy chương trình:

1

Two

3

4.5

1. **Duyệt các phần tử trong List, ArrayList và các thao tác khác**
2. **List**

Trong List bạn dùng indexer với chỉ số (chỉ số bắt đầu từ 0). Ví dụ lấy phần tử ở Index = 1;

var pro = products[2]; // đọc phần tử có index = 2

Console.WriteLine(pro.ToString());

Để duyệt qua các phần tử bạn có thể dùng lệnh for hoặc foreach

// Duyệt qua tất cả các phần tử bằng for

// products.Count = lấy tổng phần tử trong List

for (int i = 1; i < products.Count; i++)

{

var pi = products[i - 1];

Console.WriteLine(pi.ToString());

}

// Duyệt qua các phần tử bằng foreach

foreach (var pi in products)

{

Console.WriteLine(pi.ToString());

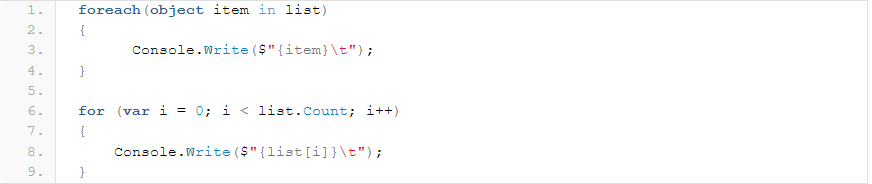
}

**Một số tính năng và phương thức khác**

* + Chèn phần tử vào List
  + Truy cập List bằng LinQ
  + Kiểm tra phần tử trong List
  + Contains(obj) kiểm tra có chứa phần tử obj hay ko
  + Reverse() đảo thứ tự danh sách
  + ToArray() copy các phần tử ra mảng

1. **ArrayList**

Để duyệt danh sách, chúng ta sử dụng vòng lặp tương tự như mảng. Một cách khác thường gặp là sử dụng vòng lặp foreach:



**Một số tính năng khác**

* Lấy thông tin về số lượng phần tử đang chứa trong danh sách: thuộc tỉnh Count
* Lấy thông tin về dung lượng hiện tại: thuộc tỉnh Capacity
* Xác định xem một giá trị có năm trong danh sách; phương thức Contains
* Sap xếp danh sách: phương thức Sort
* Xác định chỉ số của phần tử theo giá trị: phương thức IndexOf
* Chuyển đổi danh sách thành mảng: phương thức ToArray

1. **Chương trình minh họa bằng C#**

**ARRAYLIST**

Tạo 1 arraylist mới:

Graphical user interface, text, application

Description automatically generated

Chèn các phần tử vào Arraylist

Text

Description automatically generated

Thiết lập số phần tử mà Arraylist có thể chứa

Text

Description automatically generated with low confidence

Số lượng phần tử thực sự được chứa trong Arraylist:

Text

Description automatically generated

Insert phần tử:

Text, letter

Description automatically generated

Đảo ngược tất cả phần tử:

Text, letter

Description automatically generated

Sắp xếp theo thứ tự tăng dần:

Text, letter

Description automatically generated

Tìm kiếm nhị phân:

Chart, scatter chart

Description automatically generated

Xóa phần tử chỉ định:

Text

Description automatically generated

Kiểm tra phần tử:

Graphical user interface, text, application, email

Description automatically generated

Tạo bản sao từ Array hiện tại:

Text, letter

Description automatically generated

Trả về 1 mảng:

Text, letter

Description automatically generated

**LIST**

Khai báo và khởi tạo list

Text

Description automatically generated

Trả về tất cả các phần tử trong danh sách

Text

Description automatically generated

Duyệt các phần tử của mảng

Text

Description automatically generated

Sắp xếp theo thứ tự tăng dần

Text

Description automatically generated

Tạo 1 list các kiểu string và thêm 2 phần tử vào list

Text, letter

Description automatically generated

In giá trị các phần tử trong List

Graphical user interface, text, application

Description automatically generated

Chèn 1 phần tử vào đầu list

Text

Description automatically generated with low confidence

In lại giá trị các phần tử trong List để xem đã chèn được hay chưa

Text

Description automatically generated

Kiểm tra 1 phần tử có tồn tại trong List hay không

Graphical user interface, text, application, email

Description automatically generated

Xóa phần tử

Text

Description automatically generated

Đảo ngược các phần tử trong list

Text

Description automatically generated

1. **Chương trình quản lý dữ liệu sách bằng ngôn ngữ C/C++**

Vì phần code này khá dài nên nhóm tụi em đã tạo một thư mục chứa code riêng ở trong thư mục nộp bài

**TÀI LIỆU THAM KHẢO**

**[1]** https://tuhocict.com/danh-sach-list-trong-c-arraylist-list-sortedlist/

**[2]** <https://www.howkteam.vn/course/khoa-hoc-lap-trinh-c-nang-cao/arraylist-trong-c-1559>

**[3]** [**https://xuanthulab.net/collection-va-list-trong-c-c-sharp.html**](https://xuanthulab.net/collection-va-list-trong-c-c-sharp.html)

**[4]** [**https://comdy.vn/c-sharp/list-t-trong-c-sharp/**](https://comdy.vn/c-sharp/list-t-trong-c-sharp/)

**PHỤ LỤC**

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Độc lập – Tự do – Hạnh phúc**

**BIÊN BẢN HỌP NHÓM**

**1. Thời gian, địa điểm, thành phần tham dự.**

**1.1. Thời gian: 12h00p**

**1.2. Địa điểm: Zalo**

**1.3. Thành phần tham dự:**

+ Chủ trì: Nguyễn Hoàng Kha

+ Tham dự: 5

+ Vắng: 0

**2. Nội dung cuộc họp**

**2.1. Công việc các thành viên như sau\* (Bắt buộc không được để trống)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **MSSV** | **HỌ TÊN** | **TỶ LỆ ĐÓNG GÓP** | **NHÓM/**  **ĐỀ TÀI** | **Nhiệm vụ được phân công** | **Nhóm đánh giá mức độ hoàn thành công việc được phân công** |
| 1 | 2001190115 | Nguyễn Hoàng Kha | 100 | Nhóm 6/ đề tài 15 | Code 4 câu cuối phần sách c++ | Hoàn thành tốt |
| 2 | 2001190723 | Lê Hoàng Nhựt | 100 | Nhóm 6/ đề tài 15 | Code phần list của ví dụ minh hoạ c# và code 3 câu giữa c++ | Hoàn thành tốt |
| 3 | 2001190646 | Nguyễn Trọng Liêm | 100 | Nhóm 6/ đề tài 15 | Làm Word, chỉnh sửa lỗi sai, thêm bớt nội dung. Code phần arraylist của ví dụ minh họa c# | Hoàn thành tốt |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 4 | 2001190306 | Nguyễn Minh Triết | 100 | Nhóm 6/ đề tài 15 | Làm powerpoint và phần lý thuyết | Hoàn thành tốt |
| 5 | 2001190787 | Nguyễn Quốc Hoàng Sơn | 100 | Nhóm 6/ đề tài 15 | Tạo file dữ liệu và code 3 câu đầu phần sách c++ | Hoàn thành tốt |

**2.2. Ý kiến của các thành viên:** Tất cả thành viên đồng ý với ý kiến của nhóm trưởng

**2.3. Kết luận cuộc họp**

Phân công nhiệm vụ và hoàn thành trước ngày 20/12/2021

Cuộc họp đi đến thống nhất và kết thúc lúc ..12..giờ..0.. phút cùng ngày.

|  |  |
| --- | --- |
| **Thư ký**  **Nguyễn Trọng Liêm** | **Chủ trì**  **Nguyễn Hoàng Kha** |