

CÁC THAO TÁC TRÊN LIST

- Cho class Sinhvien có các thuộc tính
- { ID,Name,Toan,Ly,Hoa,Tb,Xl};



CÁC THAO TÁC TRÊN LIST

- Khai báo:

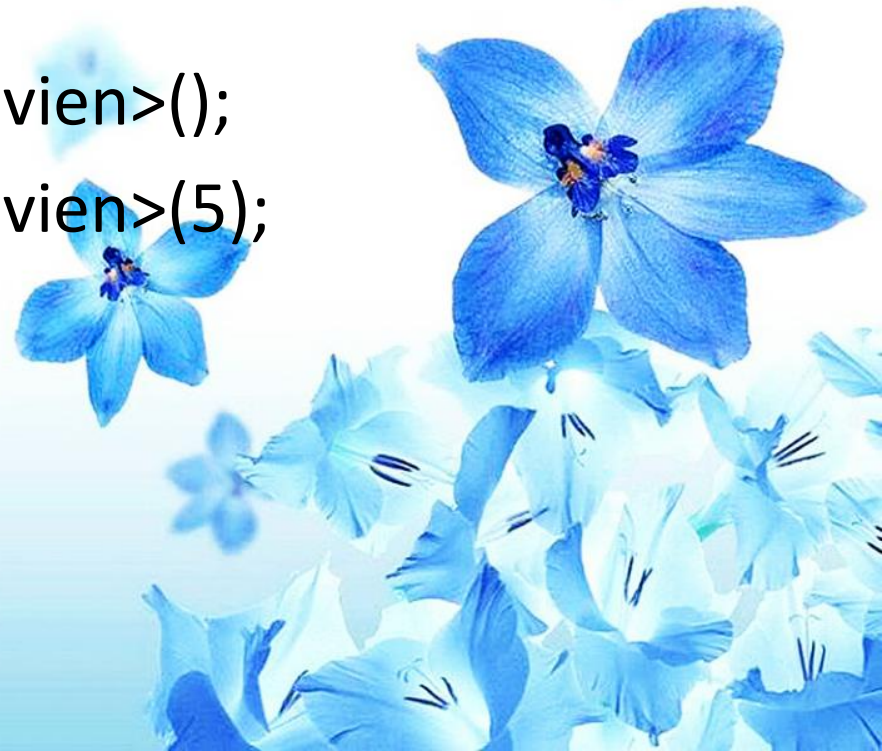
List<Kiểu dữ liệu> tên list=new List<kiểu dữ liệu>();

List<Kiểu dữ liệu> tên list=new List<kiểu dữ liệu>(spt);

Ví dụ:

List <Sinhvien>sv=new List<Sinhvien>();

List <Sinhvien>sv=new List<Sinhvien>(5);



CÁC THAO TÁC TRÊN LIST

- Thêm phần tử:

List.add(item)

Ví dụ:

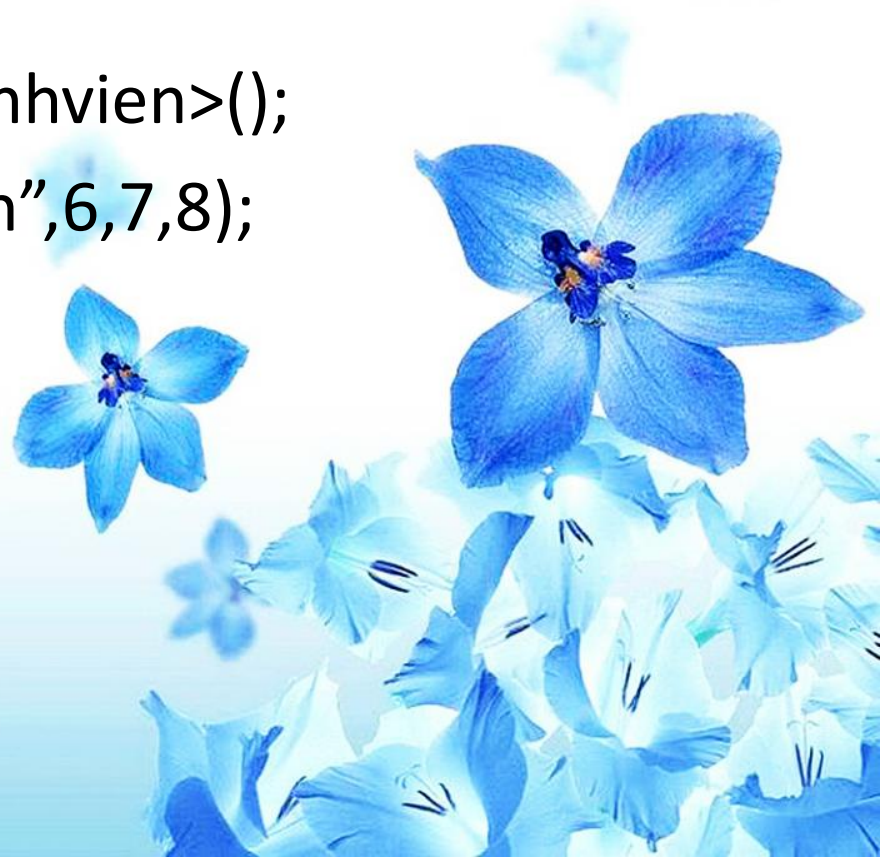
```
List<sinhvien>listsv=new List<sinhvien>();
```

```
Sinhvien sv=new sinhvien(1,"Lan",6,7,8);
```

```
Sv.tb=sv.trungbinh();
```

```
Sv.xl=sv.Xeploai();
```

```
Listsv.add(sv);
```



CÁC THAO TÁC TRÊN LIST

- Xóa phần tử:

List.remove(item)

Ví dụ:

Tìm kiếm sinh viên sv, sau đó xóa sinh viên sv

Listsv.remove(sv)



CÁC THAO TÁC TRÊN LIST

- Sắp xếp:

→ sắp tăng : `.OrderBy(tên thuộc tính A)`

nếu bằng thì theo thuộc tính B

`OrderBy (Thuộc tính A).ThenBy(thuộc tính B)`

→ Sắp giảm:

`OrderByDescending(tên thuộc tính A)`

nếu bằng thì theo thuộc tính B

`OrderBy Descending (Thuộc tính A).ThenByDescending (thuộc tính B)`



CÁC THAO TÁC TRÊN LIST

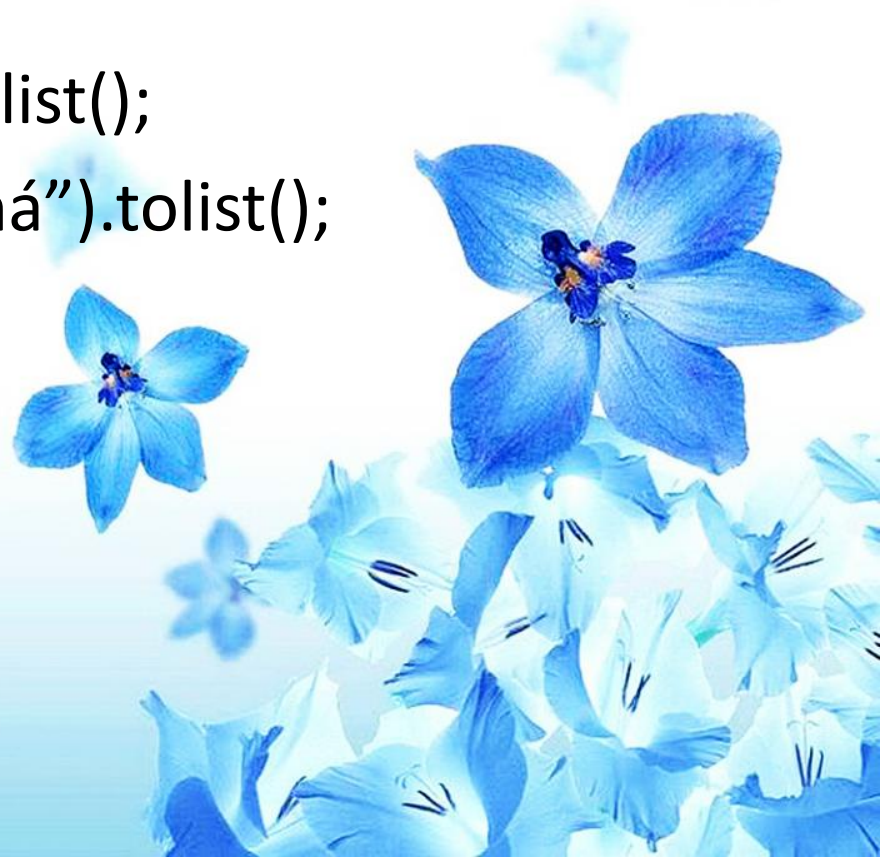
- Trích lọc theo thuộc tính A:

tên list.Where($t \Rightarrow t.A$ điều kiện).tolist();

Ví dụ:

listsv.Where($t \Rightarrow t.tb > 7$).tolist();

listsv.Where($t \Rightarrow t.xl == \text{"Khá"}$).tolist();



CÁC THAO TÁC TRÊN LIST

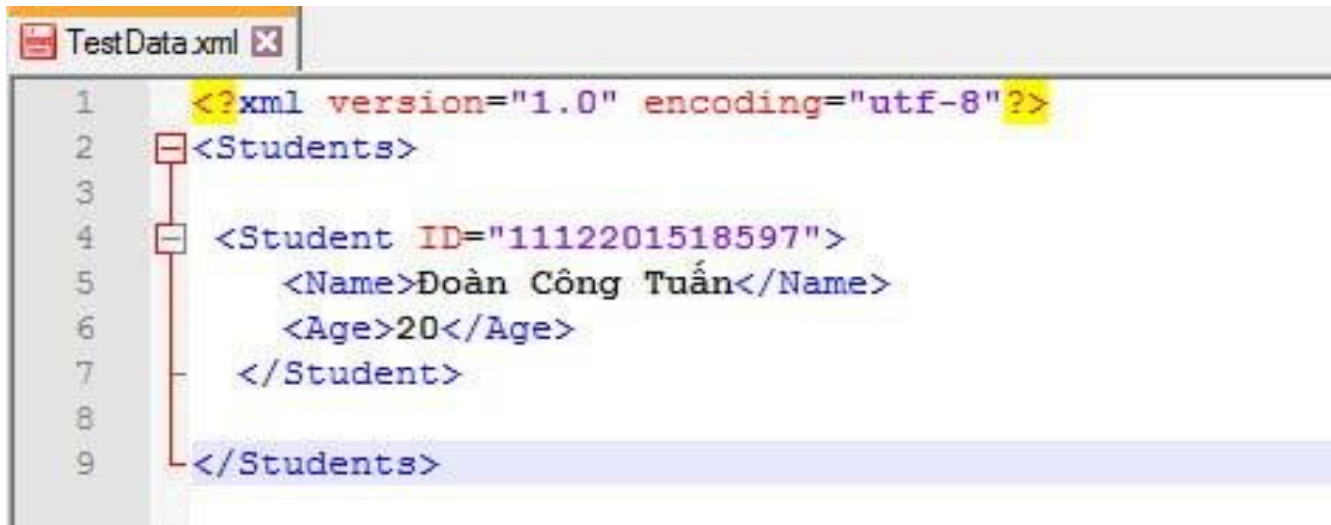
Phương pháp	Mô tả
Add()	Thêm một phần tử vào cuối Danh sách <T>.
AddRange()	Thêm các phần tử của tập hợp được chỉ định vào cuối Danh sách <T>.
BinarySearch()	Tìm kiếm phần tử và trả về một chỉ mục của phần tử.
Clear()	Loại bỏ tất cả các phần tử khỏi Danh sách <T>.
Contains()	Kiểm tra xem phần tử được 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>.

CÁC THAO TÁC TRÊN LIST

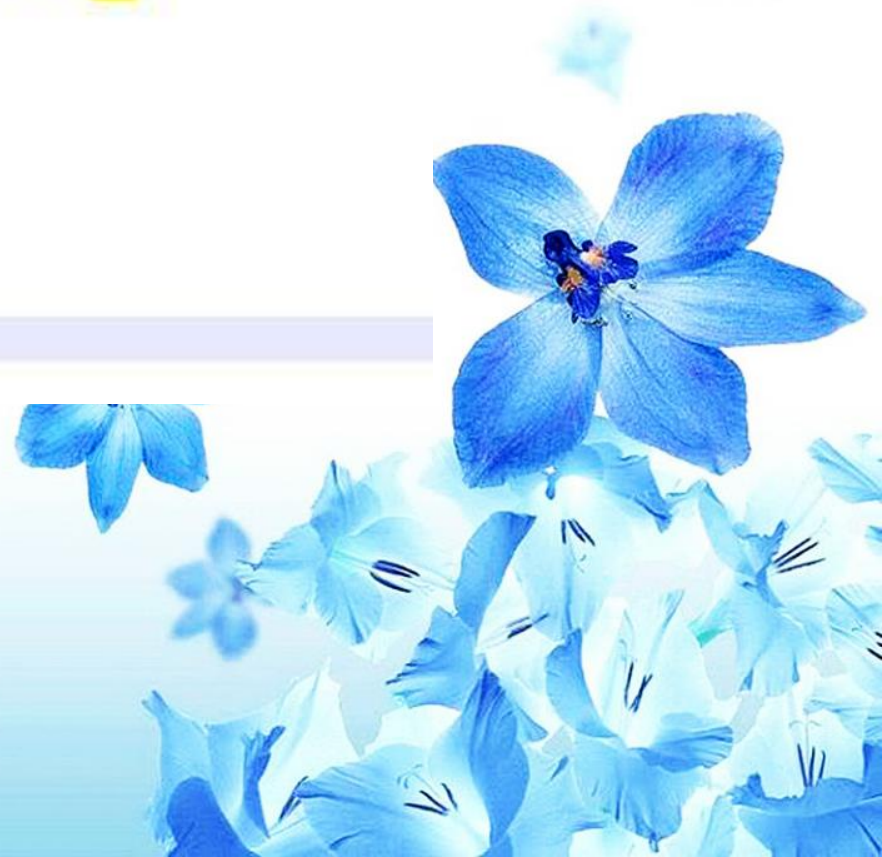
Phương pháp	Mô tả
Insert()	Chèn một phần tử tại chỉ mục được chỉ định trong Danh sách <T>.
InsertRange()	Chèn các phần tử của một tập hợp khác vào chỉ mục được chỉ định.
Remove()	Loại bỏ sự xuất hiện đầu tiên của phần tử được chỉ định.
RemoveAt()	Loại bỏ phần tử tại chỉ mục được chỉ định.
RemoveRange()	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.

ĐỌC FILE XML

Một cấu trúc dữ liệu XML mẫu



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <Students>
3
4 <Student ID="1112201518597">
5   <Name>Đoàn Công Tuấn</Name>
6   <Age>20</Age>
7 </Student>
8
9 </Students>
```



ĐỌC FILE XML

Trong cấu trúc trên thấy <students> là tag thể hiện bảng <student> chứa các record ID, Name, Age...

```
XmlDocument docXML = new XmlDocument();  
docXML.Load("D:\\sinhvien.xml");
```

SelectNodes(): Chọn ra một tập các Node thỏa điều kiện Xpath.

SelectSingleNode(): Chọn ra Node đầu tiên thỏa điều kiện.

InnerText: Lấy giá trị của node hiện tại.



```
XmlDocument read = new XmlDocument();  
read.Load("danhsach.xml"); //load file  
XmlNodeList nodelist = read.SelectNodes("/DANHSACH/SV");  
foreach(XmlNode node in nodelist)  
{  
    sinhvien sv = new sinhvien();  
    sv.Id = int.Parse(node["ID"].InnerText);  
    sv.Name = node["NAME"].InnerText;  
    // sv.Name = node.ChildNodes[1].InnerText;  
    sv.Toan = double.Parse(node["TOAN"].InnerText);  
    sv.Ly = double.Parse(node["LY"].InnerText);  
    sv.Hoa = double.Parse(node["HOA"].InnerText);  
    sv.Tb = sv.trungbinh();  
    sv.Xl = sv.Xeploai();  
    listSV.Add(sv);  
}
```



Ghi file XML:

Sử dụng **XmlTextWriter** để ghi dữ liệu

```
private void createNode(string id, string pName, .....  
XmlTextWriter writer)
```

```
{  
    writer.WriteStartElement("SV");  
    writer.WriteStartElement("ID");  
    writer.WriteString(id);  
    writer.WriteEndElement();  
    writer.WriteStartElement("NAME");  
    writer.WriteString(pName);  
    writer.WriteEndElement();  
}
```



<https://www.techieclues.com/blogs/how-to-read-and-write-xml-documents-in-csharp>

<https://zetcode.com/csharp/xmldocument/>

