

لیست

در زبان برنامه‌نویسی سی‌شارپ، لیست‌ها یکی از ساختارهای داده‌ای بسیار مفید و کاربردی هستند که برای ذخیره‌سازی مجموعه‌ای از اشیاء به کار می‌روند. لیست‌ها به برنامه‌نویسان این امکان را می‌دهند که مجموعه‌ای از اشیاء هم‌نوع را به صورت دینامیک و قابل تغییر ذخیره کنند. این به این معناست که می‌توان به آسانی به عناصر اضافه کرد، آن‌ها را حذف کرد و یا به عناصر خاصی دسترسی داشت.

خصوصیات کلاس

کلاس لیست در سی‌شارپ دارای دو خصوصیت کلیدی به نام‌های **Count** و **Capacity** است. **Capacity** پروپرتی **Count** تعداد عناصر موجود در لیست را برمی‌گرداند، در حالی که **Capacity** حداکثر تعداد عناصری را که لیست می‌تواند بدون افزایش اندازه ذخیره کند.

متدهای کلاس

کلاس لیست در سی‌شارپ دارای مجموعه‌ای از متدهای کاربردی است که مدیریت داده‌ها را تسهیل می‌کنند. **Add(T item)** برای اضافه کردن یک عنصر جدید به انتهای لیست استفاده می‌شود، در حالی که **Remove(T item)** برای حذف اولین وقوع یک عنصر مشخص به کار می‌رود. **Insert** به برنامه‌نویس این امکان را می‌دهد که یک عنصر را در یک موقعیت خاص در لیست وارد کند. **Sort** برای مرتب‌سازی عناصر لیست به کار می‌رود و **Contains(T item)** وجود یک عنصر خاص را در لیست بررسی می‌کند. همچنین، **Clear** تمامی عناصر موجود در لیست را حذف می‌کند.

مثال : با فرض موجود بودن آرایه ای از عدد های صحیح برنامه ای بنویسید که داده های آن را در لیستی قرار دهد.

```
C# Program.cs
int[] nums={1,2,3};
List<int> ls=new List<int>();
ls.AddRange(nums);
Console.WriteLine(ls[0]);
```

مثال : با فرض موجود بودن لیستی از عدد های صحیح بنام **ls** برنامه ای بنویسید که عددی از کاربر گرفته و به انتهای لیست اضافه نماید.

```
C# Program.cs
List<int> ls=new List<int>();
int num=Convert.ToInt32(Console.ReadLine());
ls.Add(num);
```

مثال : با فرض موجود بودن لیستی از عددهای صحیح بنام **ls** برنامه ای بنویسید که داده های موجود در لیست را در خروجی نمایش دهد.

```
C# Program.cs
List<int> ls=new List<int>();
int[] nums={22,55,77};
ls.AddRange(nums);
foreach (int n in nums){
    Console.WriteLine(n);
}
```

مثال: بافرض موجود بودن لیستی از عددهای صحیح بنام **ls** برنامه ای بنویسید که عدد ۱۰۰ را به ابتدای این لیست اضافه نماید.

```
C# Program.cs
List<int> ls=new List<int>()
{11,22,33};
ls.Insert(0,100);
foreach(int n in ls){
    Console.WriteLine(n);
}
```

مثال: بافرض موجود بودن لیستی از عددهای صحیح بنام **ls** برنامه ای بنویسید که عدد ۱۲ را از آن حذف نماید.

```
C# Program.cs
List<int> ls=new List<int>()
{11,22,33,12};
ls.Remove(12);
foreach(int n in ls){
    Console.WriteLine(n);
}
```

مثال: بافرض موجود بودن لیستی از عددهای صحیح بنام **ls** برنامه ای بنویسید که داده های آن را مرتب و آنها را در خروجی از بزرگترین عدد به کوچکترین عدد نمایش دهد.

```
C# Program.cs
List<int> ls=new List<int>()
{7,2,80,12};
ls.Sort();
ls.Reverse();
foreach(int n in ls){
    Console.WriteLine(n);
}
```

مثال: بافرض موجود بودن لیستی از عددهای صحیح بنام **ls** برنامه ای بنویسید که عدد ۲ را به همه داده های لیست اضافه نماید.

```
C# Program.cs
List<int> ls=new List<int>()
{7,2,80,12};
for(int i=0;i<ls.Count;i++){
    ls[i]=ls[i]+2;
}
foreach(int n in ls){
    Console.WriteLine(n);
}
```

مثال: بافرض موجود بودن لیستی از عددهای صحیح بنام **ls** برنامه ای بنویسید که بزرگترین عدد این لیست را در خروجی نمایش دهد

```
C# Program.cs
List<int> ls=new List<int>()
{7,2,80,12};
Console.WriteLine(ls.Max());
```

مثال: بافرض موجود بودن لیستی از عددهای صحیح بنام **ls** برنامه ای بنویسید که میانگین داده های موجود در لیست را در خروجی نمایش دهد.

```
C# Program.cs
List<int> ls=new List<int>()
{7,2,80,12,10};
Console.WriteLine(ls.Sum()/ls.Count);
```