

عنوان: ایجاد لیستی از کلاسی جنریک

نویسنده: وحید نصیری

تاریخ: ۱۳۹۰/۰۵/۲۹ ۲۱:۲۰:۰۰

آدرس: [www.dotnettips.info](http://www.dotnettips.info)

برچسب‌ها: #C

کلاس جنریک زیر را در نظر بگیرید:

```
public class Column<T>
{
    public string Name { set; get; }
    public T Data { set; get; }
}
```

مشکلی که با این نوع کلاس‌ها وجود دارد این است که نمی‌توان مثلا لیست زیر را در مورد آن‌ها تعریف کرد:

```
ICollection<Column<T>> myList = new List<Column<T>>>();
```

به عبارتی می‌خواهیم یک لیست از کلاسی جنریک داشته باشیم. راه حل انجام آن به صورت زیر است:

```
using System.Collections;

namespace Tests
{
    public interface IColumn
    {
        string Name { set; get; }
        object Data { set; get; }
    }

    public class Column<T> : IColumn
    {
        public string Name { set; get; }

        public T Data { set; get; }

        object IColumn.Data
        {
            get { return this.Data; }
        }
    }
}
```

```

        set { this.Data = (T)value; }
    }
}
}

```

ابتدا یک اینترفیس عمومی را همانند اعضای کلاس Column تعریف می‌کنیم که در آن بجای T از object استفاده شده است. سپس یک پیاده سازی جنریک از این اینترفیس را ارائه خواهیم داد؛ با این تفاوت که اینبار خاصیت Data مربوط به اینترفیس، به صورت خصوصی و صریح با استفاده از IColumn.Data تعریف می‌شود و نمونه‌ی جنریک هم نام آن، عمومی خواهد بود. اکنون می‌توان نوشت:

```
var myList = new List<IColumn>();
```

برای مثال در این حالت تعریف لیست زیر که از تعدادی وهله‌ی کلاسی جنریک ایجاد شده، کاملاً مجاز می‌باشد:

```

var myList = new List<IColumn>
{
    new Column<int> { Data = 1, Name = "Col1"},
    new Column<double> { Data = 1.2, Name = "Col2"}
};

```

خوب، تا اینجا یک مرحله پیشرفت است. اکنون اگر بخواهیم در این لیست، Data مثلاً عنصری را که نامش Col1 است، دریافت کنیم چه باید کرد؟ آن هم نه به شکل object بلکه از نوع T مشخص:

```

static T GetColumnData<T>(IList<IColumn> list, string name)
{
    var column = (Column<T>)Convert.ChangeType(list.Single(s => s.Name.Equals(name)),
    typeof(Column<T>), null);
    return column.Data;
}

```

و نمونه‌ای از استفاده آن:

```
int data = GetColumnData<int>(myList, "Col1");
```

## نظرات خوانندگان

نویسنده: رضا عرب  
تاریخ: ۱۳۹۰/۰۵/۳۱ ۰۳:۰۳:۱۴

ممنون از مطلب مفیدتون، به نظر شما این کد خواناتر نیست؟

```
(static T GetColumnData(IList list, string name  
    }  
    return (T)list.Single(p => p.Name == name).Data  
    }
```

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۰/۰۵/۳۱ ۰۸:۳۱:۲۷

بله. این روش هم خوبه.

بیشتر هدفم طرح نکته وجود Convert.ChangeType در مورد تبدیل نوعهای جنریک بود.