

ویژگیهای جدید کامپایلر

این ویژگیهای جدید که به معرفی آن می پردازیم بیشتر به سهولت در برنامه نویسی کمک کرده و باعث کاهش تعداد خطهای برنامه نویسی می شوند.

:Automatic Properties

اگر تا بحال در برنامه های خود از Property ها استفاده کرده اید بدون شک آن را بصورت زیر نوشته اید.

```
public class Person {  
  
    private string _firstName;  
  
    private string _lastName;  
  
    public string FirstName {  
  
        get {  
  
            return _firstName;  
  
        }  
  
        set {  
  
            _firstName = value;  
  
        }  
  
    }  
  
    public string LastName {  
  
        get {  
  
            return _lastName;  
  
        }  
  
    }  
  
}
```

```
set {  
    _lastName = value;  
}  
}  
}
```

کامپایلر جدید C# که به همراه Orcas می باشد، روش جدید و جالبی را برای تعریف Property هایی شبیه آن چه در مثال بالا نوشته شده است، معرفی می کند. این ویژگی جدید که باعث خلاصه نویسی در تعریف Property ها می شود Properties Automatic نامیده می شود. در روش جدید دیگر نیازی به نوشتن فیلدهای خصوصی و همچنین نوشتن کد برای get/set در Property نیست. در عوض کامپایلر بطور خودکار این کار را انجام خواهد داد. پس با این اوصاف می توان مثال فوق را بصورت زیر نوشت.

```
public class Person {  
    public string FirstName {  
        get; set;  
    }  
    public string LastName {  
        get; set;  
    }  
}
```

و یا حتی ساده تر :

```
public class Person {  
    public property string FirstName;  
    public property string LastName;  
}
```

و یا بصورت زیر :

```
public class Person {  
    [Property]  
    public string FirstName;  
    [Property]  
    public string LastName;  
}
```

هنگامی که کامپایلر C# در Orcas به get/set خالی برای یک Property برخورد می کند بطور خودکار فیلدهای خصوصی مربوطه را تولید کرده و کدهای مرتبط به get/set آن را نیز اجرا می کند.

:Object Initializer

این ویژگی جدید هم برای کامپایلر C# و VB در Studio Orcas Visual گنجانده شده است. اغلب ما برای تعریف یک object از یک کلاس و استفاده از آن همانند مثال زیر عمل می کنیم.

```
Person person = new Person();
```

```
person.FirstName = "name";  
person.LastName = "family";
```

خاصیت جدید **Object Initializer** در کامپایلر **Orcas** به ما اجازه می دهد مثال فوق را بصورت خلاصه شده زیر بنویسیم.

```
Person person = new Person { FirstName = "name ", LastName =  
"family"};
```

همچنین **Property** هایی که دارای تایپ های پیچیده تری نسبت به مثال فوق باشند نیز قابل خلاصه شدن با **Object Initializer** می باشند به مثال زیر توجه کنید.

```
Person person = new Person {  
    FirstName = "name ",  
    LastName = "family ",  
    Address = new Address {  
        Street = "",  
        City = "",  
        State = ""}  
};
```

:Collection Initializer

Object Initializer که در بالا به تعریف آن پرداختیم کمک می کند که به سادگی و بطور خیلی خلاصه **object** هایی را به **collection** ها اضافه کنیم. مثلاً "اگر خواسته باشیم اطلاعات سه نفر را به یک لیست اضافه کنیم همانند زیر عمل می کنیم.

```
List<Person> people = new List<Person>();  
  
people.Add( new Person { FirstName = "name 1", LastName = "family 1"  
} );  
  
people.Add( new Person { FirstName = "name 2", LastName = "family 2"  
} );  
  
people.Add( new Person { FirstName = "name 3", LastName = "family 3"  
} );
```

ملاحظه می کنید که اگر می خواستیم بدون در نظر گرفتن **Object Initializer** مثال فوق را بنویسیم نیاز به کدنویسی بیشتری داشتیم. ویژگی جدید دیگری که در کامپایلر **C#** و **VB** در **Orcas** افزوده شده است **Collection Initializer** نام دارد و به ما اجازه می دهد تا برای **collection** چندین بار تابع **Add** بکار ببریم. به مثال زیر توجه کنید.

```
List<Person> people = new List<Person> {  
  
    new Person { FirstName = "name 1", LastName = "family 1"},  
  
    new Person { FirstName = "name 2", LastName = "family 2"},  
  
    new Person { FirstName = "name 3", LastName = "family 3"}  
  
};
```

هنگامی که کامپایلر به کدی همانند مثال فوق بر می خورد خود بطور خودکار کد مثال قبل را تولید می کند. ملاحظه می کنید که استفاده از این ویژگیهای جدید چطور به کم شدن تعداد خطوط برنامه کمک می کنند.

باید توجه داشته داشت که این قابلیت های جدید هیچ ربطی به C# 3.0 و یا VB 9.0 ندارند بلکه همانطور که گفته شد اینها چند ویژگی جدید از کامپایلر جدید VS Orcas می باشد. یکی از قابلیت های VS Orcas به اصطلاح target multy بودن آن است. بدین معنی که می توان در آن مشخص کرد که با کدام ویرایش از Framework NET می خواهیم کار کنیم. پس با داشتن Orcas می توان برنامه های مبتنی بر NET 2.0 را با استفاده از نکاتی که گفته شد، نوشت.

محمد قلیزادگان

1389/05/06