

# فصل پنجم-قسمت دوم

کار کردن با ساختار داده ای

## مفهوم آرایه:

یکی از عمومی ترین نیازها در برنامه نویسی قابلیت نگهداری لیستی از اطلاعات مشابه و یا مرتبط به هم است. برای این مورد بایستی از آرایه ها استفاده کنید. آرایه ها، لیستی از متغیر ها می باشند که همه از یک نوع هستند. برای مثال ممکن است بخواهید سن تمامی دوستان خود را در یک آرایه از اعداد صحیح و یا نام تمامی آنها را در یک آرایه از رشته ها قرار دهید. در این بخش با نحوه ایجاد، پر کردن و استفاده از آرایه ها در برنامه آشنا خواهید شد.

## تعریف و استفاده از آرایه ها:

هنگامی که در برنامه یک آرایه تعریف می کنید، در حقیقت متغیری ایجاد می کنید که بیش از یک عنصر را بتواند در خود نگهداری کند. برای مثال اگر یک متغیر رشته ای را به صورت زیر تعریف کنید فقط یک رشته را میتوانید در آن نگهداری کنید:

```
string strName;
```

اما آرایه ها باعث ایجاد نوعی حالت افزایشی در متغیر می شوند، به این ترتیب بیش از یک مقدار را می توانید در یک متغیر ذخیره کنید. اگر بخواهید یک متغیر را به صورت آرایه تعریف کنید باید در مقابل نوع متغیر از علامت [ ] استفاده کنید برای مثال کد زیر متغیری ایجاد می کند که بتواند ۱۰ عنصر را در خود نگهداری کند:

```
string[] strName = new string[10];
```

هنگامی که یک آرایه را ایجاد کردید، می توانید به تک تک عناصر آن با استفاده از اندیس آن عنصر دسترسی پیدا کنید. اندیس عناصر یک آرایه همواره عددی بین صفر و شماره ی آخرین عنصر آرایه است. شماره ی آخرین عنصر هر آرایه هم برابر با یک واحد کمتر از طول آرایه است. بنابراین اندیس هر آرایه عددی بین صفر تا یک واحد کمتر از آخرین عنصر آرایه است. برای مثال اگر بخواهید مقدار عنصر سوم آرایه قبلی را برابر با رشته ی "Katie" قرار دهید باید از کد زیر استفاده کنید:

```
strName[2] = "Katie";
```

برای دسترسی به مقدار یک عنصر از آرایه هم میتوانید از همین روش استفاده کنید:

```
MessageBox.Show(strName[2]);
```

نکته ی مهم در اینجا این است که اگر یکی از عناصر آرایه را تغییر دهید، مقدار بقیه عناصر تغییری نخواهد کرد. برای مثال اگر کد زیر را در برنامه به کار ببرید:

```
strName[3] = "Betty";
```

مقدار `strName[2]` همچنان برابر با "Katie" خواهد ماند. احتمالاً بهترین روش برای فهمیدن این که آرایه ها چیستند و چگونه کار می کنند نوشتن برنامه ای با استفاده از آنها است.

## امتحان کنید: تعریف و استفاده از یک آرایه ساده

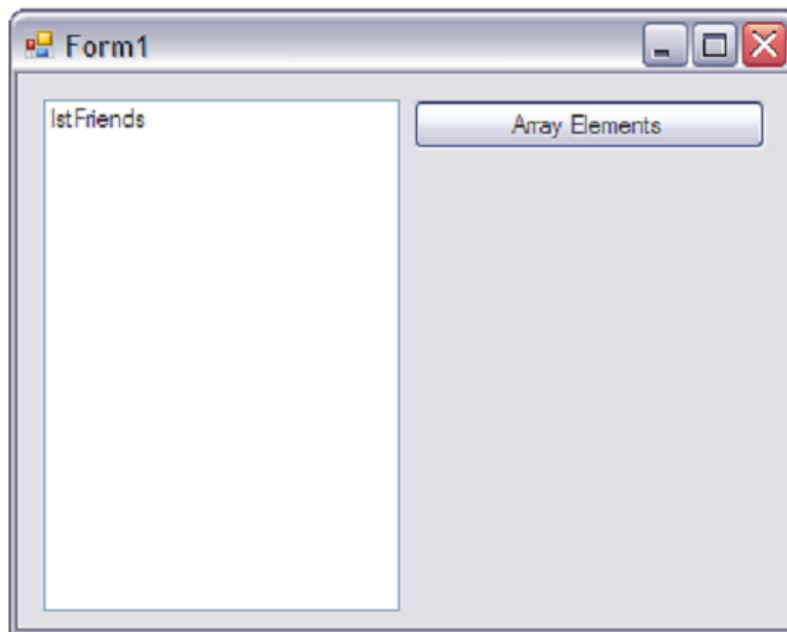
- (۱) با استفاده از ویژوال استودیو روی منوی File کلیک کرده و سپس Project → New را انتخاب کنید. در پنجره New Project یک پروژه ویندوزی به نام Array Demo ایجاد کنید.
- (۲) هنگامی که قسمت طراحی Form1 به نمایش داده شد، یک کنترل ListBox به فرم اضافه کنید. خاصیت Name این کنترل را برابر با lstFriends و خاصیت IntegralHeight آن را برابر با False قرار دهید.
- (۳) اکنون یک کنترل Button به فرم اضافه کرده، خاصیت Name آن را برابر با btnArrayElements و خاصیت Text آن را برابر با Array Elements قرار دهید. فرم شما باید مشابه شکل ۱-۵ شده باشد.
- (۴) روی کنترل Button دو بار کلیک کنید و کد زیر را در متد مربوط به رویداد Click آن وارد کنید:

```
private void btnArrayElements_Click(object sender,
                                   EventArgs e)
{
    // Declare an array
    string[] strFriends = new string[5];

    // Populate the array
    strFriends[0] = "Robbin";
    strFriends[1] = "Bryan";
    strFriends[2] = "Stephanie";
    strFriends[3] = "Sydney";

    strFriends[4] = "Katie";

    // Add the first array item to the list
    lstFriends.Items.Add(strFriends[0]);
}
```



شکل ۱-۵

- (۵) برنامه را اجرا کنید و روی دکمه ی Array Elements کلیک کنید. کنترل ListBox روی فرم با نام Robbin پر می شود.

## استفاده از foreach:

یکی از عمومی ترین روشهای استفاده از آرایه ها، به وسیله حلقه های foreach است. این نوع حلقه ها در فصل ۴، هنگامی که با آرایه رشته ای برگردانده شده توسط تابع GetDirectories کار می کردید، معرفی شدند. در بخش امتحان کنید بعد، ملاحظه خواهید کرد که چگونه می توانیم از این حلقه ها در آرایه ها استفاده کنیم.

### امتحان کنید: استفاده از حلقه های foreach با آرایه ها

(۱) اگر برنامه ی Arrays Demo در حال اجرا است آن را ببندید. قسمت ویرایشگر کد را برای Form1 باز کنید و کد زیر را در بالاترین قسمت در بدنه کلاس خود وارد کنید:

```
public partial class Form1 : Form
{
    // Declare a form level array
    private string[] strFriends = new string[5];
```

(۲) به قسمت طراحی فرم مربوط به Form1 برگردید و بر روی قسمت خالی فرم دو بار کلیک کنید. متد مربوط به رویداد Load فرم به صورت اتوماتیک ایجاد می شود. حال کد زیر را در این متد وارد کنید:

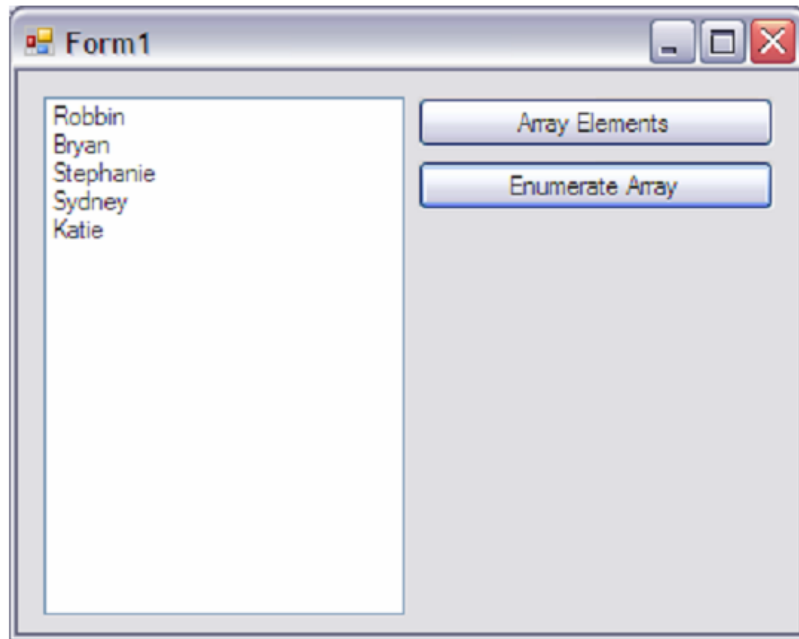
```
private void Form1_Load(object sender, EventArgs e)
{
    // Populate the array
    strFriends[0] = "Robbin";
    strFriends[1] = "Bryan";
    strFriends[2] = "Stephanie";
    strFriends[3] = "Sydney";
    strFriends[4] = "Katie";
}
```

(۳) مجدداً به قسمت طراحی فرم برگردید و یک کنترل Button به فرم اضافه کنید. خاصیت Name این کنترل را برابر با btnEnumerateArray و خاصیت Text آن را برابر با Enumerate Array قرار دهید.

(۴) روی این کنترل دو بار کلیک کنید و کد زیر را در متد مربوط به رویداد Click آن وارد کنید:

```
private void btnEnumerateArray_Click(object sender,
    EventArgs e)
{
    // Enumerate the array
    foreach (string strName in strFriends)
    {
        // Add the array item to the list
        lstFriends.Items.Add(strName);
    }
}
```

(۵) برنامه را اجرا کرده و بر روی دکمه ی Enumerate Array کلیک کنید نتیجه ای را مشابه شکل ۵-۲ مشاهده خواهید کرد.



شکل ۲-۵