

بسم الله الرحمن الرحيم

نام و نام خانوادگی : امیر حسین تقی زاده

رشته : فناوری اطلاعات و ارتباطات

موضوع : دیتا تایپ Reference – مرجع

استاد : میثاق یاریان

Value Type

به داده نوعی **Value Type** گفته می شود که یک مقدار را در فضای حافظه ی خود ذخیره کند. و این به این معناست که متغیر هایی که از نوع این داده نوع تعریف میشوند به طور مستقیم دارای مقداری در خود هستند.

نکته : تمام Value Type ها از فضای نام `System.ValueType` مشتق میشوند که آن فضای نامی هم در فضای نام `System.Object` قرار دارد.

ارسال با مقدار

را به عنوان آرگومان برای یک متد ارسال می کنید ، سیستم **Value Type** وقتی یک متغیر از نوع یک کپی جداگانه از آن متغیر را ایجاد کرده و آن را برای متد ارسال میکند. بنابراین اگر تغییری در مقدار آن متغیر در متد مربوطه ایجاد شود تاثیری بر مقدار اصلی آن ندارد.

نمونه مثال زیر نحوه ی عملکرد روش ارسال با مقدار را نشان میدهد :

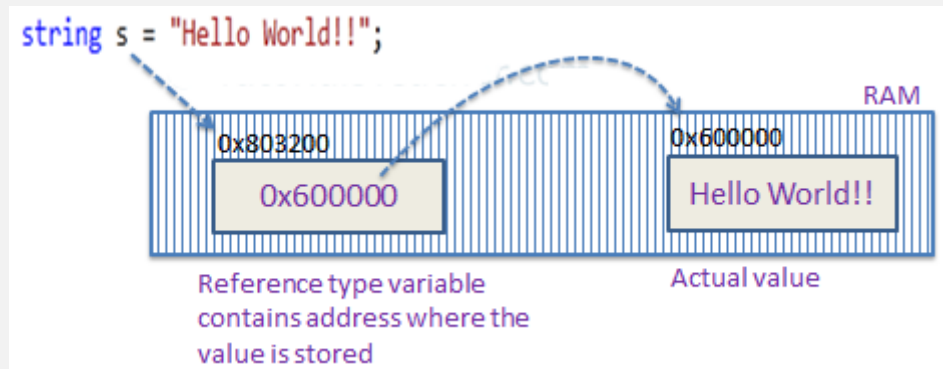
```
static void ChangeValue(int x)
{
    x = 200;
    Console.WriteLine(x);
}

static void Main(string[] args)
{
    int i = 100;
    Console.WriteLine(i);
    ChangeValue(i);
    Console.WriteLine(i);
}
```

Reference Type

برخلاف Value Type ها ، Reference Type ها مقادیرشان را به صورت مستقیم در خود ذخیره نمی کنند. در عوض آنها آدرس مکانی از حافظه را که مقدار در آن قرار گرفته است، در خود ذخیره میکنند. به عبارت دیگر Reference Type ها شامل یک اشاره گر هستند که به مکانی دیگر از حافظه اشاره میکند که داده یا مقدار در آن ذخیره شده است.

تصویر زیر چگونگی تخصیص حافظه را برای متغیر رشته ای بالا نشان میدهد:



همانطور که در تصویر بالا مشاهده می کنید سیستم یک مکان تصادفی در حافظه (0x803200) را برای متغیر "S" انتخاب کرده است. مقداری که در متغیر "S" قرار میگیرد 0x600000 است که آدرس خانه ای از حافظه است که مقدار اصلی یعنی Hello Worlds در آن قرار گرفته است.

داده نوع های زیر همگی Reference Type هستند:

- String
- تمام آرایه ، حتی اگر مقادیر آنها از نوع value type باشد
- Class
- Delegates

ارسال با ارجاع

وقتی یک متغیر Reference Type را به عنوان آرگومان از یک متد به متد دیگری میفرستید دیگر کپی ایی از آن ساخته نمیشود. در عوض آدرس آن متغیر به متد مربوطه ارسال میشود. نمونه مثال زیر روش ارسال با ارجاع را نشان میدهد:

```
static void ChangeReferenceType(Student std2)
```

```
{
```

```
    std2.StudentName = "Steve";
```

```
}
```

```
static void Main(string[] args)
```

```
{
```

```
    Student std1 = new Student();
```

```
    std1.StudentName = "Bill";
```

```
    ChangeReferenceType(std1);
```

```
    Console.WriteLine(std1.StudentName);
```

```
}
```

در نمونه مثال فوق ، از آنجایی که Student یک کلاس (class) است هنگامی که شی ایی از کلاس Student به نام std1 را به عنوان آرگومان به متد ChangeReferenceType() ارسال می کنیم ، آن چیزی که در عمل ارسال میشود آدرس حافظه ی شی std1 است. بنابراین وقتی متد ChangeReferenceType() فیلد StudentName را تغییر میدهد ، مقدار اصلی فیلد StudentName از شی std1 را تغییر میدهد. به همین دلیل شی std1 و آرگومان std2 هر دو به یک آدرس در حافظه اشاره میکنند. بنابراین این خروجی برابر با رشته ی steve است:

Steve