

The in, ref, and out Modifiers

Method parameters have modifiers available to change the desired outcome of how the parameter is treated. Each method has a specific use case:

- **ref** is used to state that the parameter passed *may* be modified by the method.
- **out** is used to state that the parameter passed *must* be modified by the method.
- **in** is used to state that the parameter passed *cannot* be modified by the method.

چگونه از يك تابع بيش از يك خروجي دريافت كنيم؟ظاهرا به نظر مي رسد كه توابع فقط مي توانند يك خروجي داشته باشند و بلافاصله پس از فراخواني return كار تابع پايان يافته است. اما در سي شارپ دو كلمه ي كليدي به نام هاي ref و out اضافه شده اند كه اين امر را ممكن مي سازند.

پارامترهای Out و Ref یکی از موضوعات C# هستند كه بيشترين اشتباه در آنها رخ مي دهد. در حالت پيش فرض پارامترها به صورت Value ارسال مي شوند و Out و Ref كمك مي كنند پارامتر از طريق منبع(Reference) به تابع ارسال شوند. در ادامه شيوه های مختلف ارسال پارامتر را به تفصيل توضيح مي دهيم.

به طور كلي سه روش براي ارسال پارامتر وجود دارد:

ارسال به صورت Value(مقدار)، ارسال به صورت Ref و ارسال به صورت Out

ارسال پارامتر به صورت: value

ارسال به روش value parameter به صورت پيش فرض مي باشد . هنگامي كه ما متغيري (variable) به ورودي يك متد ارسال مي كنيم ، در صورتي كه نوع آن پارامتر را ذكر نكنيم به صورت مقداري ارسال مي شود. به عبارت ديگر مقدار آن متغير به تابع فرستاده مي شود . در صورت تغيير مقدار متغير در بدنه تابع ، تغييری در پارامتر ارسال شده ، ايجاد نمی شود. هر گاه در اعلان پارامترها از كلمات كليدي ref و out استفاده نشود، به صورت value فرستاده می شود.

كد زير مثالی در اين مورد است.

تابع زير را در پنجره كنسول بنويسيد:

```
public void plus(int t)
{
    Console.WriteLine("the plus function run : "+ ++t);
}
```

```
}
```

فراخوانی تابع در بدنه:

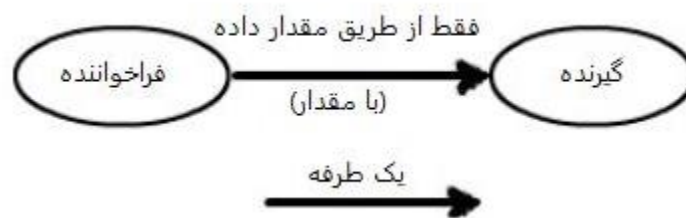
```
int t = 10;
    plus(t);
    Console.WriteLine("the t variable value : " + t);
    Console.ReadLine();
```

خروجی برنامه:

```
the plus function run : 11
the t variable value : 10
```

دیدید که هر تغییری در بدنه تابع تاثیری در پارمتر ندارد.

شکل زیر به طور خلاصه ارسال پارمتر با مقدار (Value) را نشان می دهد:



ارسال پارمتر به صورت: ref

وقتی تغییری را به صورت ref به یک تابع ارسال می کنیم ، مقدار متغیر ارسال نمی شود بلکه آدرس متغیر به بدنه متد فرستاده می شود و هر تغییری در متغیر محلی روی متغیر اصلی نیز اعمال می شود. به این نوع پارامترها ارجاعی می گویند.

برای مثال کد تابع زیر را در کنسول وارد کنید:

```
public void plus(ref int t)
{
    Console.WriteLine("the plus function run : "+ ++t);
}
```

فراخوانی تابع:

```
int t = 10;

plus(ref t);

Console.WriteLine("the t variable value : " + t);

Console.ReadLine();
```

خروجی:

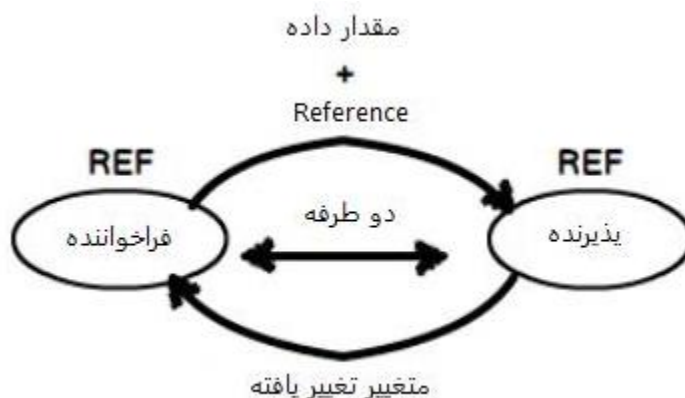
```
the plus function run : 11
the t variable value : 11
```

می بینید که تغییر در متغیر محلی روی پارمتر اصلی اعمال شد.

نکته: نوشتن کلمه کلیدی ref هم در ارسال به تابع وهم در دریافت پارمتر الزامی می باشد.

نکته: قبل از ارسال پارمتر به صورت ref حتما باید متغیر مقدار داشته باشد.

شکل زیر ارسال پارمتر به صورت منبع (Reference) را نشان می دهد:



ارسال پارمتر به صورت: out

این شیوه ارسال هم به صورت ارجاعی می باشد. این پارامترها زمانی استفاده می شود که قصد ارسال اطلاعاتی به متد را نداشته باشیم، بلکه می خواهیم اطلاعات از متد برگردد.

نیازی به مقدار دهی قبل از ارسال به متد نیست ولی حتما باید درون تابع (قبل از بازگشت به متدی فراخوانی شده) مقدار بگیرد.

تابع را زیر را مجدد در کنسول وارد کنید:

```
public void plus(out int t)
{
    // مقدار دهی درون تابع
    t = 10;
    Console.WriteLine("the plus function run : "+ ++t);
}
```

فراخوانی تابع:

```
// نیازی به گرفتن مقدار نیست
int t;
Program.plus(out t);
Console.WriteLine("the t variable value : " + t);
Console.ReadLine();
```

خروجی:

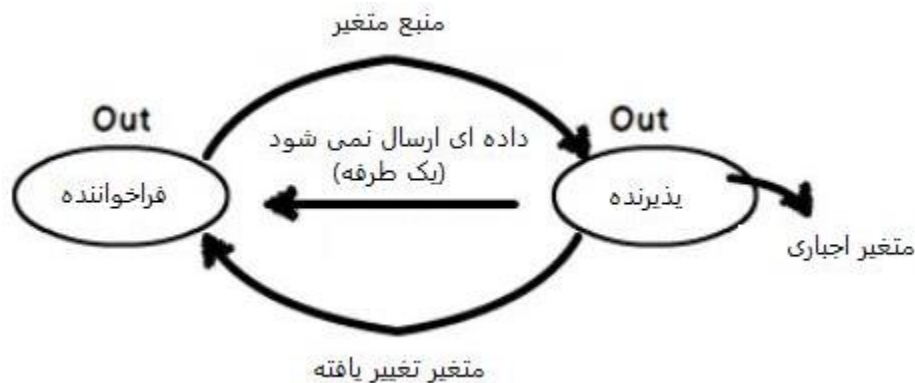
```
the plus function run : 11
the t variable value : 11
```

نکته:

آرگومان های out نیازی به مقداردهی اولیه ندارند و برای خروجی از آن استفاده می کنیم.

آرگومان های ref حتما باید مقدار دهی شوند و هنگامی که بخواهیم اطلاعاتی را به متد ارسال کنیم، پس از دستکاری شدن، اطلاعات برگشت داده شود.

شکل زیر ارسال پارامتر به صورت Out را به طور خلاصه نشان می دهد:



مطالب گفته شده را به صورت زیر می توان جمع بندی کرد:

1) Out و Ref به ارسال پارامترها توسط منبع (Reference) کمک می کنند.

2) در REF با داده های منبع نیز از فراخواننده به پذیرنده و بالعکس منتقل می شوند. این برای مثال از فراخواننده به پذیرنده و بالعکس انجام می گیرد. و به صورت روش دو طرفه است.

3) در OUT فقط منبع ارسال می شود و داده تغییر یافته از فراخواننده به پذیرنده ارسال می شود. این یک روش یک طرفه است.

4) در IOU اگر داده از تابع فراخواننده ارسال شود دور انداخته خواهد شد و متغیر داخل تابع مقدار دهی خواهد شد. تصویر زیر خلاصه ای از موارد گفته شده را نشان می دهد.