

# بسم الله الرحمن الرحيم

## آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

انتقال مقادیر به متدهای جاوا

گردآورنده : مهندس افشین رفوآ

## انتقال مقادیر به متدهای جاوا

شما می توانید مقادیر را به متدهای جاوای خود طوری انتقال دهید که با این مقدار کاری انجام شود. این مقدار در بین پرانتزهای متد قرار می گیرد.

به گروه **MyMethods** خود بازگردید. اکنون دومین متد **total** را اضافه کنید.

```
package prjmethods;

public class MyMethods {

    int total() {
        int a_Value = 10 + 10;

        return a_Value;
    }

    void print_text() {

        System.out.println( "Some Text Here" );
    }

    int total(int aNumber) {
        int a_Value = aNumber + 20;

        return a_Value;
    }
}
```

اکنون دو متد هم نام داریم: **total**. تفاوت بین این دو این است که متد جدید دارای مقداری در بین پرانتزها می باشد. جاوا اجازه ی این کار را به شما می دهد که **method overloading** نامیده می شود. شما می توانید هر تعداد متد هم نام که می خواهید با هر مقدار بازگشتی داشته باشید. به هر حال نمی توانید یک نوع متغیر مشابه بین پرانتزها داشته باشید. بنابراین نمی توانید دو متد **total** داشته باشید که مقادیر **int** را بازمی

گرداند که هر دوی آنها دارای مقادیر **int** در داخل پرانتزها هستند. به عنوان مثال نمی توانید این کار را انجام دهید.

```
int total( int aNumber ) {  
    int a_Value = aNumber + 20;  
    return a_Value;  
}  
int total( int aNumber ) {  
    int a_Value = aNumber + 50;  
    return a_Value;  
}
```

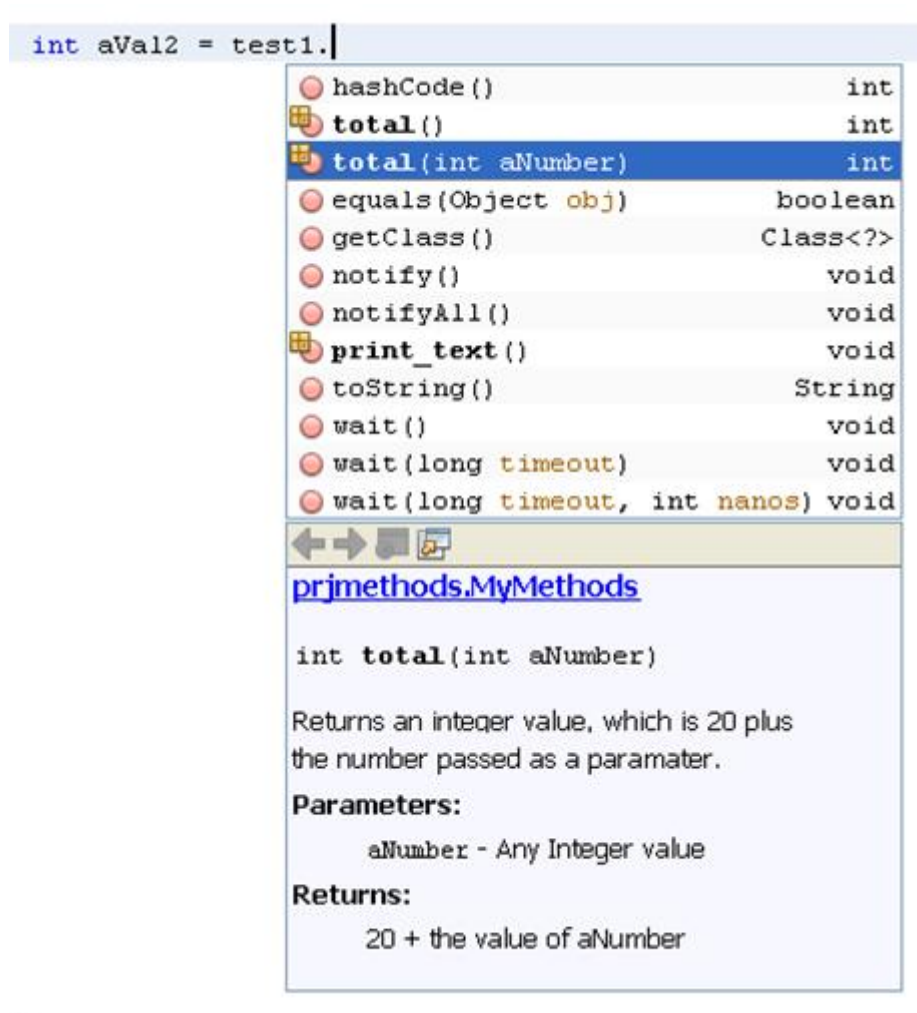
گرچه هر دو متد دو کار متفاوت انجام می دهند، هر دو دارای تیتريهای یکسان متد می باشند. قبل از اینکه متد جدید خود را امتحان کنید، چند کامنت مستقیما در بالای متد اضافه کنید.

```
/**  
 * Returns an integer value, which is  
 * 20 plus the number passed as a parameter.  
 * @param aNumber Any Integer value  
 * @return 20 + the value of aNumber  
 */  
int total(int aNumber) {  
    int a_Value = aNumber + 20;  
  
    return a_Value;  
}
```

مشاهده خواهید کرد که کامنتها در یک لحظه چگونه عمل می کنند. اما **param** در متدهای فوق مخفف **parameter** می باشد. یک پارامتر یک اصطلاح تخصصی برای مقدار بین پرانتزهای تیتريهای متد می باشد. پارامتر ما **aNumber** نامیده می شود و این پارامتر دارای مقادیر صحیح می باشد. به استفاده ی کاراکتر **@** قبل از **param** و **return** دقت داشته باشید. تمام کاری که ما با خود متد انجام می دهیم، انتقال یک مقدار صحیح و افزودن **20** به این مقدار انتقالی می باشد. مقدار بازگشتی مجموع هر دو می باشد. اکنون به کد خود بازگردید و خط زیر را به آن اضافه کنید.

```
int aVal2 = test1.total(30);
```

به محض اینکه نقطه را بعد از آبجکت **test1** تایپ کردید، لیست پیش رو دوباره مشاهده خواهید کرد. متد جدید شما روی آن خواهد بود. روی متد جدید کلیک کنید تا آن را مشخص کنید و **NetBeans** مورد زیر را نمایش خواهد داد.



اکنون کامنت های اضافه شده در باکس آبی در زیر لیست متدها نمایش داده می شوند. هرکس دیگری که وارد متد شما شود، باید قادر به تشخیص کاری که انجام می دهد، باشد. خطوط **@param** و **@return** از کامنت ها پر شده و حالت **bold** می شوند.

اما زمانی که متد **total2** را اضافه می کنید، عدد **30** را بین پرانتزها تایپ کنید. سپس یک نقطه ویرگول در انتهای خط تایپ کنید. اکنون متد اصلی شما باید مشابه زیر باشد.

```

public static void main(String[] args) {

    MyMethods test1 = new MyMethods();

    int aVal = test1.total();

    System.out.println( "Method result= " + aVal );

    test1.print_text();

    int aVal2 = test1.total(30);

}

```

30 تایپ شده بین پرانتزهای متد جدید **total** تحویل داده شده و در متغیر **aNumber** قرار می گیرد.

Value handed over

```
int aVal2 = test1.total(30);
```

Value is handed over here

```

int total(int aNumber) {
    int a_Value = aNumber + 20;

    return a_Value;
}

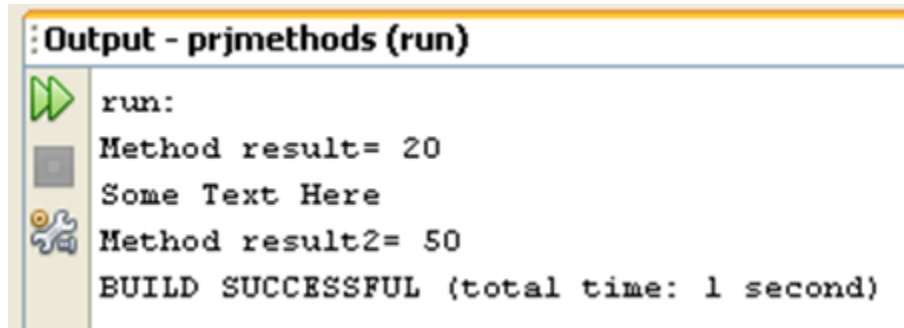
```

زمانی که مقدار را تحویل می دهید، متد می تواند وارد کار شود.

یک خط چاپی وارد کد خود کنید.

```
System.out.println( "Method result2= " + aVal2 );
```

سپس برنامه ی خود را اجرا کنید. پنجره ی **Output** صفحه ی زیر را نمایش می دهد.



```
Output - prjmethods (run)
run:
Method result= 20
Some Text Here
Method result2= 50
BUILD SUCCESSFUL (total time: 1 second)
```

بنابراین متد **total**، 30 را به 20 اضافه کرده و سپس پاسخ را به متغیری به نام **aVal2** بازمی گرداند.

در بخش بعد، در مورد چگونگی انتقال بیشتر از یک مقدار به متدهای خود را فرا خواهید گرفت.