

# JAVA



خطاها و اشتباهاتی که ممکن است

هنگام برنامه نویسی جاوا رخ دهد

اشتباه صدا زدن یک عضو

غیراستاتیک در یک متد استاتیک



Core Java

آموزش ساده و آسان جاوا

به نام خدا

تقدیرم به هموطنان عزیزم

جاوا را با لذت یاد بگیرید!

# Core Java

## آموزش ساده و آسان جاوا

عنوان : اشتباه صدا زدن یک عضو غیراستاتیک در یک متد استاتیک

مولف: رحمان زارعی

تاریخ انتشار: ۲۴ دی ۹۸

قیمت: رایگان

راه های ارتباط با مدرس و نویسنده کتاب:

سایت: [www.javapro.ir](http://www.javapro.ir)

ایمیل: [Rahman.zarie92@gmail.com](mailto:Rahman.zarie92@gmail.com)

کانال تلگرام: <https://t.me/javalike>

آی دی تلگرام: <https://t.me/RZUTAB>

اینستاگرام: <https://www.instagram.com/javapro.ir>

اگر میخوای از مولف این کتاب رایگان حمایت مالی کنی اینجا کلیک کن.

تقدیم به هموطنان عزیزم

# Core Java

## آموزش ساده و آسان جاوا

سخن مدرس:

با عرض سلام و ادب خدمت مخاطبان ارزشمند سایت آموزشی جاوا پرو، بنده رحمان زارعی هستم، قصد داریم در این مقاله آموزشی رایگان به خطاها و اشتباهات متداولی که هر برنامه نویس جاوا هنگام کد زدن و برنامه نویسی برایش پیش میاد را بررسی کنیم.

دوست عزیز! توجه داشته باش که اشتباه کردن اینطور که میگن هم اصلا بد نیست! خطا کردن برای هر فردی که میخواهد روز به روز بهتر بشه و تجربه اش بیشتر و بیشتر بشه خیلی خوبه! چرا؟ چون درک شما رو نسبت به مشکلات و چالش های برنامه نویسی بالا می برد و همچنین در بهبود مهارت های برنامه نویسی شما را بهتر و بهترتر می کند!

از اشتباه کردن نترسید حالا در هر زمینه ای از زندگی که میخواهید باشه و مهم اینه که شما از اشتباهات خود درس بگیرید و رفیق اینو از من داشته باش! که هر چقدر اشتباهات بیشتری انجام بدی، درس های بیشتری نسبت به اونایی که اشتباه نمی کنند یاد می گیری و در نهایت به یک فرد باتجربه و باخردتر در حوزه کار تخصصی خود خواهی شد.

صرف نظر از این که چقدر در زمینه جاوا حرفه ای هستی محال است که یکی از اشتباهاتی که جلوتر به آنها خواهیم پرداخت رو در هنگام یادگیری جاوا انجام نداده باشی :-)

# Core Java

## آموزش ساده و آسان جاوا

اشتباه اول:

1. Accessing non-static members from static methods (mostly from the main method)
  - صدا زدن یک عضو غیراستاتیک در یک متد استاتیک که متداول ترین ان صدا زدن متغیر نمونه غیر استاتیک یا یک متد غیراستاتیک کلاس در متد استاتیک main می باشد:

مثال: به قطعه کد زیر توجه کنید:

```
package www.javapro.ir;

public class Test1 {

    public int number; // instance variable

    public static void main(String[] args) {

        number = 10; // compile error

        System.out.println(number); // compile error
    }

}
```

خروجی: **خطای کامپایل داده می شود.**

توضیحات:

```
number = 10; // compile error
System.out.println(number); // compile error
```

# Core Java

## آموزش ساده و آسان جاوا

- دلیل خطای کامپایل این است که متغیر number یک متغیر نمونه کلاس غیر استاتیک می باشد و در متد استاتیک main صدا زده شده است و ما نمی توانیم یک متغیر یا متد غیر استاتیک را در یک متد استاتیک صدا بزنینم.
- برای صدا زدن عضو غیر استاتیک number در متد استاتیک main باید از کلاس مون شی ایجاد کنیم و با شی ایجاد شده متغیر number که غیر استاتیک می باشد را صدا بزنینم:

```
package www.javapro.ir;

public class Test1 {

    public int number; // instance variable

    public static void main(String[] args) {
        Test1 t=new Test1();
        t.number = 10;
        System.out.println(t.number);
    }
}
```

خروجی: برنامه صحیح و بدون خطای کامپایل می باشد.

10

- راه حل دیگر این است که متغیر یا متد کلاس مون را از نوع استاتیک تعریف کنیم و از آن در متدهای استاتیک خود نظیر متد main استفاده کنیم، البته همه این دو روش بستگی به هدف شما در ساخت برنامه موردنظرتون دارد:

# Core Java

## آموزش ساده و آسان جاوا

```
package www.javapro.ir;

public class Test1 {

    public static int number; // static variable

    public static void main(String[] args) {

        number = 10;
        System.out.println(number);
    }

}
```

خروجی: برنامه صحیح و بدون خطای کامپایل می باشد.

10

- در این قطعه کد بالا یک متغیر استاتیک با نام number در بدنه کلاس تعریف کرده و آن را در بدنه متد استاتیک main صدا زده ایم.

خب تا اینجا برای صدا زدن متغیر نمونه غیر استاتیک یک کلاس در یک متد استاتیک و خطای کامپایل پیش آمده و رفع خطای آن بایکدیگر گفتگو کردیم. حالا قصد داریم این مسئله رو برای استفاده از متدهای غیر استاتیک در متدهای استاتیک یک کلاس بررسی کنیم:

مثال: به قطعه کد زیر توجه کنید:

```
package www.javapro.ir;

public class Test1 {

    // instance method:
    public int sum(int a, int b) {
        return a + b;
    }

}
```

# Core Java

## آموزش ساده و آسان جاوا

```
public static void main(String[] args) {  
  
    int a = 399;  
    int b = 256;  
  
    int c = sum(a, b); // compile error  
  
    System.out.println(c);  
}  
}
```

خروجی: خطای کامپایل می دهید و اگر برنامه را run کنیم با خطای زیر روبرو می شویم:

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:  
    Cannot make a static reference to the non-static method sum(int, int) from the type Test1  
  
    at www.javapro.ir.Test1.main(Test1.java:16)
```

### توضیحات:

- در کد بالا متد sum در بدنه کلاس Test1 غیراستاتیک تعریف شده است و آن را در بدنه متد استاتیک main صدا زده ایم، همان طور که گفتیم یک عضو غیر استاتیک کلاس را نمی توانیم در یک متد استاتیک صدا بزیم و از آن استفاده کنیم، به همین علت برنامه خطای کامپایل داده است.
- برای رفع خطای کامپایل رخ داده در برنامه بالا به صورت زیر عمل می کنیم:

```
package www.javapro.ir;  
  
public class Test1 {  
  
    // instance method:  
    public int sum(int a, int b) {  
        return a + b;  
    }  
  
    public static void main(String[] args) {
```



# Core Java

## آموزش ساده و آسان جاوا

```
int a = 399;
int b = 256;
Test1 t = new Test1();
int c = t.sum(a, b);

System.out.println(c);
}
}
```

خروجی:

655

توضیحات:

- به دلیل این که متد sum غیر استاتیک می باشد برای صدا زدن آن در متد استاتیک main ابتدا از کلاس Test1 شی ساخته و از طریق شی ایجاد شده متد sum را صدا زده ایم.
- شیوه دیگر استفاده متد sum در متد استاتیک main این است که متد sum را از نوع استاتیک تعریف کنیم:

```
package www.javapro.ir;

public class Test1 {

    // static method:
    public static int sum(int a, int b) {
        return a + b;
    }

    public static void main(String[] args) {

        int a = 399;
        int b = 256;

        int c = sum(a, b);
    }
}
```

# Core Java

## آموزش ساده و آسان جاوا

```
        System.out.println(c);
    }
}
```

خروجی:

655

توضیحات:

- به دلیل این که متد sum استاتیک می باشد می توانیم بدون شی ساختن از کلاس Test1 آن را در متد استاتیک main صدا بزنینم.

خب ما اینجا تمامی مثال هامون پیرامون متد استاتیک main کلاسمون بود در حالی که صدا زدن عضو غیر استاتیک در هر متد استاتیکی خطای کامپایل میده به مثال زیر توجه کنید:

```
package www.javapro.ir;

public class Test1 {

    int a = 50; // instance variable
    int b = 60;
    int sum;

    // static method:
    public static void get(int c, int d) {
        a = c;
        b = d;

        printSum();
    }

    public void printSum() { // instance method:
        sum = a + b;
    }
}
```

# Core Java

## آموزش ساده و آسان جاوا

```

        System.out.println(sum);
    }

    public static void main(String[] args) {
        get(90, 90);
    }
}

```

خروجی: **خطای کامپایل می دهد**، چرا؟ به دلیل این که متغیرهای a و b و متد printSum و که غیراستاتیک هستند در متد استاتیک get صدا زده شده اند. دو راه حل برای برطرف کردن خطای کامپایل بالا موجود است:

راه اول: متغیرهای a و b و متد printSum و به تبع آن متغیر sum را از نوع استاتیک تعریف کنیم، چرا متغیر sum را باید استاتیک تعریف کنیم؟ زیرا قرار است متد printSum را استاتیک تعریف کنیم و از جهتی قرار است متغیر sum درون متد printSum صدا بزیم پس باید sum را از نوع استاتیک تعریف کنیم. برای درک بهتر برا خودتون تمرین کنید مثلا متد printSum را استاتیک کنید و متغیر sum را درونش صدا بزید و همین طور برا خودتون آزمون و خطا کنید که یادگیرید دلیلش چیه 😊

```

package www.javapro.ir;

public class Test1 {

    static int a = 50;
    static int b = 60;
    static int sum;

    public static void get(int c, int d) {
        a = c;
        b = d;

        printSum();
    }

    public static void printSum() {

```

# Core Java

## آموزش ساده و آسان جاوا

```
        sum = a + b;
        System.out.println(sum);
    }

    public static void main(String[] args) {
        get(90, 90);
    }
}
```

خروجی:

180

راه حل دوم: با ایجاد شی از کلاس Test1، متغیرهای غیراستاتیک a و b و متد غیراستاتیک printSum را در درون متد استاتیک get صدا میزنیم.

```
package www.javapro.ir;

public class Test1 {

    int a = 50; // instance variable
    int b = 60;
    int sum;

    // static method:
    public static void get(int c, int d) {
        Test1 t = new Test1();
        t.a = c;
        t.b = d;

        t.printSum();
    }

    public void printSum() { // instance method:
        sum = a + b;
        System.out.println(sum);
    }
}
```

# Core Java

## آموزش ساده و آسان جاوا

```
public static void main(String[] args) {  
    get(90, 90);  
}  
}
```

خروجی:

180

هر چند که امروزه این اشتباه توسط IDE های امروزی تشخیص داده می شود و به برنامه نویسی هشدار می دهد اما خب یکی از اشتباهات متداول و زیادی بود که توسط برنامه نویسان جاوا ممکنه است رخ دهد.

قصد داریم در سلسله آموزش هایی نکات کاربردی، خطاها و اشتباهات یا به زبان ساده گیرهایی که ممکنه در هنگام کدزدن به زبان جاوا براتون پیش بیاد را بررسی کنیم، پس با ما همراه باشید و به سایت جاواپرو سرزنید.

نظرات شما بسیار برای ما مهم است، لطفا در نظرتون رو در مورد این آموزش جاوا برای ما ارسال کنید.

دوست من آرزوی سعادت و کامیابی رو برات دارم، خدانگهدار

**Core Java**

آموزش ساده و آسان جاوا

سایت آموزش زبان جاوا به زبان ساده، آسان و شیرین!!!

[www.JAVAPRO.ir](http://www.JAVAPRO.ir)

آموزش جاوا SE را با تجربه شفقی و به زبان خودمونی یاد بگیرید!!!!

بازدید از کانال

بازدید از سایت

هر روز مفاهیم و مثال های جدید به سایت اضافه می شود برای اطلاع از مطالب جدید روی سایت عضو کانال شوید.

دقل و تصرف ، ویرایش و کپی زدن تمامی آموزش های جاوا لایک به دور از افلاق حرفه ای ست و مرام می باشد.