

نویسنده



مهدی عادل‌فر

۱,۷۳۸

۳۲۵

۱۹۸

۲۴۳

مهدی عادل‌فر، بنیان‌گذار TOSINSO، کارشناس ارشد نرم‌افزار کامپیوتر از دانشگاه صنعتی امیرکبیر و #C و جاوا و اندروید کار می‌کنم. در زمینه‌های موبایل و وب و ویندوز فعالیت دارم و به طراحی نرم‌افزار و اصول مهندسی نرم‌افزار علاقه مندم.



## آموزش جاوا (Java) قسمت ۳۰ : معرفی و شرح Exception ها قسمت ۲



۲۷۸۸



با سلام به همه دوستان و همراهان عزیز Itpro. در مطلب قبلی به مفاهیم exception ها پرداختیم و در این مطلب قصد داریم که مبحث Exception ها را بیشتر پیش ببریم و با این مبحث از برنامه نویسی جاوا بیشتر آشنا شویم.

### متدهای exception ها

گفتیم که همه ی Exception ها از کلاس Throwable مشتق شده اند و متدهای زیر متدهای کلاس Throwable می باشد که در همه ی exception ها وجود دارد.

- متد getMessage این متد پیامی را در مورد exception رخ داده شده را در قالب String برمی گرداند این پیام در constructor کلاس Throwable قرار داده شده است.
- متد getCause این متد دلیل رخ دادن exception به وجود آمده را در قالب یک شیء Throwable بر می گرداند.
- متد printStackTrace این متد نه ورودی دارد و نه مقدار بازگشتی دارد. وظیفه این متد این است که سلسله مراتب کلاس ها و محل هایی از کلاس ها که exception در آنها رخ داده است را به ترتیب در خروجی چاپ می کند و می توان با استفاده از اطلاعات این متد فهمید که خطا و exception در چه محلی رخ داده است.
- متد getStackTrace این متد قبلی سلسله مراتب محل خطاها را چاپ می کرد حال اگر بخواهیم این اطلاعات را در یک آرایه داشته باشیم که بتوانیم بر روی آنها اعمال مختلفی انجام دهیم از این متد استفاده می کنیم. این متد همه ی اطلاعات stack trace را در یک آرایه از اشیاء StackTraceElement قرار می دهد و این آرایه را برمی گرداند.
- متد fillInStackTrace این متد ورودی ندارد و شیء جاری را با stackTrace جاری پر می کند.

### مدیریت exception ها (به دام انداختن)

برای این که بتوانیم در هنگام وقوع یک exception آن را تشخیص داده و عمل خاصی را انجام دهیم از کلمات کلیدی try, catch استفاده می کنیم. Try, catch اسم بلوک های کد است که برای مدیریت exception ها به کار می رود. کدهایی را که ممکن است exception تولید کنند را داخل بلوک try قرار می دهیم و عمل مربوط به هرکدام از exception ها را داخل بلوک های catch می نویسیم. نحوه نوشتن و استفاده از این بلوک ها در برنامه جاوا به شکل زیر خواهد بود:

```
try
{
    //Protected code
}catch(ExceptionName e1)
{
    //Catch block
}
```

به کدی که داخل بلوک try قرار می گیرد کد محافظت شده می گویند. دقت داشته باشید که بلوک های catch باید بلافاصله بعد از بلوک try قرار بگیرند. بعد از هر بلوک try هم باید بلوک های catch و یا بلوک finally بیاید. بلوک های catch برای درگیر شدن با exception ها به کار می روند. اصولاً هر بلوک catch برای مدیریت یک Exception به کار می رود. وقتی که در داخل کدهای بلوک try خطایی رخ دهد بلوک catch مربوط به آن بلوک اجرا خواهد شد. در کد زیر مثالی از مدیریت exception ها را با استفاده از بلوک های try, catch آورده ایم. در این مثال ما یک آرایه ۲ خانه ای تعریف کرده ایم ولی در داخل برنامه قصد داریم که به خانه سوم آن دسترسی داشته باشیم که برنامه exception تولید می کند و ما آن exception را مدیریت می کنیم.

```
import java.io.*;
public class ExceptTest{

    public static void main(String args[]){
        try{
            int a[] = new int[2];
            System.out.println("Access element three : " + a[3]);
        }catch(ArrayIndexOutOfBoundsException e){
            System.out.println("Exception thrown ." + e);
        }
        System.out.println("Out of the block");
    }
}
```

خروجی برنامه بالا متن زیر خواهد بود:

```
Exception thrown :java.lang.ArrayIndexOutOfBoundsException: 3
Out of the block
```

## استفاده از چندین بلوک catch

یک بلوک try می تواند چندین بلوک Catch را به دنبال داشته باشد. سینتکس نوشتاری چندین بلوک catch به شکل زیر خواهد بود:

```
try
{
    //Protected code
}catch(ExceptionType1 e1)
{
```

```
//Catch block
}catch(ExceptionType2 e2)
{
//Catch block
}catch(ExceptionType3 e3)
{
//Catch block
}
```

در کد بالا ۳ بلوک catch وجود دارد که هرکدام از آنها برای یک Exception خاص است. شما به هر تعداد که بخواهید می توانید بلوک catch بعد از یک بلوک try داشته باشید. زمانی که در کد داخل بلوک catch یک exception رخ دهد برنامه به دنبال اولین بلوک catch می رود. اگر نوع exception مشخص شده در این بلوک Catch با نوع exception رخ داده یکی بود کد های داخل بلوک اجرا خواهند شد. در غیر این صورت به دنبال بلوک catch دوم خواهد رفت و در صورت عدم تطبیق نوع exception ها به سراغ exception بعدی خواهد رفت. این عمل به همین ترتیب انجام خواهد شد تا بلوک مورد نظر پیدا شود. در صورتی که بلوک Catch مورد نظر پیدا نشد خطایی صادر خواهد شد و اجرای برنامه متوقف خواهد شد. برای مثال به کد زیر دقت کنید

```
try
{
file = new FileInputStream(fileName);
x = (byte) file.read();
}catch(IOException i)
{
i.printStackTrace();
return -1;
}catch(FileNotFoundException f) //Not valid!
{
f.printStackTrace();
return -1;
}
```

## مدیریت همزمان بیش از یک exception

از نسخه ۷ جاوا به بعد این امکان به وجود آمده است که به عنوان ورودی یک بلوک Catch بتوان چند exception را فرستاد و هرکدام از exception ها را با استفاده از | جدا نمود. برای مثال به کد زیر دقت کنید:

```
catch (IOException|FileNotFoundException ex) {
logger.log(ex);
throw ex;
```

### کلمات کلیدی throw, throws

اگر داخل متدی از کدهایی استفاده کردیم که امکان رخ دادن exception چک شده داشته باشد و آن را داخل catch, try قرار ندهیم باید از کلمه کلیدی throws استفاده کنیم. این کلمه کلیدی هنگام تعریف تابع و در خط اول قرار می گیرد. اما کاربرد کلمه

کلیدی throw به طور کامل با throws تفاوت دارد. این کلمه کلیدی برای زمانی است که بخواهیم به طور دستی یک exception را بسازیم و با استفاده از throw می توان کاری کرد که یک exception رخ دهد. به عبارت دیگر کلمه کلیدی throws برای این است که نوشتن try, catch را به عقب بیندازیم ولی کلمه throw برای ساختن یک exception به طور دستی است. یک مثال از کلمه کلیدی throws و throw برای RemoteException در کد زیر آورده ایم.

```
import java.io.*;
public class className
{
    public void deposit(double amount) throws RemoteException
    {
        // Method implementation
        throw new RemoteException();
    }
    //Remainder of class definition
}
```

متدی که می سازیم ممکن است که چندین exception را تولید کند که می توان هنگام تعریف آن در جلوی کلمه کلیدی throws این exception ها را به ترتیب آورد. برای مثال به کد زیر دقت کنید که دو exception را در جلوی throws آورده است.

```
import java.io.*;
public class className
{
    public void withdraw(double amount) throws RemoteException,
        InsufficientFundsException
    {
        // Method implementation
    }
    //Remainder of class definition
}
```

تا به این قسمت از مطالب به معرفی و نحوه مدیریت exception ها در جاوا پرداختیم. در قسمت بعدی مطالب بیشتری در مورد Exception ها خواهیم گفت **Itpro باشید**

نویسنده: مهدی عادل فر

منبع: [جزیره برنامه نویسی و توسعه نرم افزار وب سایت توسینسو](#)

هرگونه نشر و کپی برداری بدون ذکر منبع و نام نویسنده دارای اشکال اخلاقی می باشد.

عنوان





رایگان

۲

آموزش جاوا (Java) قسمت ۲ : شی گرایی قس ...



رایگان

۳

آموزش جاوا (Java) قسمت ۳ : شی گرایی قسم...



رایگان

۴

آموزش جاوا (Java) قسمت ۴ : شی گرایی قس ...



رایگان

۵

آموزش جاوا (Java) قسمت ۵ : انواع داده



رایگان

۶

آموزش جاوا (Java) قسمت ۶ : انواع متغیر



رایگان

۷

آموزش جاوا (Java) قسمت ۷ : Modifier ها



رایگان

۸

آموزش جاوا (Java) قسمت ۸ : عملگرها قسم...



رایگان

۹

آموزش جاوا (Java) قسمت ۹ : عملگرها قسم...



رایگان

۱۰

آموزش جاوا (Java) قسمت ۱۰ : حلقه های تکرار



رایگان

۱۱

۱۱

آموزش جاوا (Java) قسمت ۱۱ : عبارات شرطی



رایگان

۱۲

آموزش جاوا (Java) قسمت ۱۲ : کلاس Wrappers...



رایگان

۱۳

آموزش جاوا (Java) قسمت ۱۲ : کلاس Wrappers...



رایگان

۱۴

آموزش جاوا (Java) قسمت ۱۴: کلاس ...



رایگان

۱۵

آموزش جاوا (Java) قسمت ۱۵ : کلاس String ...



رایگان

۱۶

آموزش جاوا (Java) قسمت ۱۶ : کلاس String ...



رایگان

۱۷

آموزش جاوا (Java) قسمت ۱۷ : کلاس String ...



رایگان

۱۸

آموزش جاوا (Java) قسمت ۱۸ : کلاس String ...



رایگان

۱۹

آموزش جاوا (Java) قسمت ۱۹ : StringBuilder ...












رایگان










۲۰

آموزش جاوا (Java) قسمت ۲۰ : استفاده از آرآی ...



رایگان

۲۱	آموزش جاوا (Java) قسمت ۲۱ : آموزش کار با ...
	رایگان
۲۲	آموزش جاوا (Java) قسمت ۲۲ : عبارات منظم ...
	رایگان
۲۳	آموزش جاوا (Java) قسمت ۲۳ : تعریف و است ...
	رایگان
۲۴	آموزش جاوا (Java) قسمت ۲۴ : تعریف و است ...
	رایگان
۲۵	آموزش جاوا (Java) قسمت ۲۵ : کار با استریم ...
	رایگان
۲۶	آموزش جاوا (Java) قسمت ۲۶ : کار با دایرکتور...
	رایگان
۲۷	آموزش جاوا (Java) قسمت ۲۷ : کلاس ByteAr ...
	رایگان
۲۸	آموزش جاوا (Java) قسمت ۲۸ : معرفی کلاس ...
	رایگان
۲۹	آموزش جاوا (Java) قسمت ۲۹ : معرفی و شرح...
	رایگان
۳۰	آموزش جاوا (Java) قسمت ۳۰ : معرفی و شرح...
	

رایگان
<p>۳۱</p> <p>آموزش جاوا (Java) قسمت ۳۱ : معرفی و شرح...</p> <p></p> <p>رایگان</p>
<p>۳۲</p> <p>آموزش جاوا (Java) قسمت ۳۲ : استفاده از انو...</p> <p></p> <p>رایگان</p>
<p>۳۳</p> <p>آموزش جاوا (Java) قسمت ۳۳ : وراثت کلاس ...</p> <p></p> <p>رایگان</p>
<p>۳۴</p> <p>آموزش جاوا (Java) قسمت ۳۴ : Override کرد...</p> <p></p> <p>رایگان</p>
<p>۳۵</p> <p>آموزش جاوا (Java) قسمت ۳۵ : چند ریختی ...</p> <p></p> <p>رایگان</p>
<p>۳۶</p> <p>آموزش جاوا (Java) قسمت ۳۶ : Enum و ویژگی ...</p> <p></p> <p>رایگان</p>
<p>۳۷</p> <p>آموزش جاوا (Java) قسمت ۳۷ : معرفی Abstra...</p> <p></p> <p>رایگان</p>
<p>۳۸</p> <p>آموزش جاوا (Java) قسمت ۳۸ : استفاده از کپ ...</p> <p></p> <p>رایگان</p>
<p>۳۹</p> <p>آموزش جاوا (Java) قسمت ۳۹ : استفاده از ایند ...</p> <p></p> <p>رایگان</p>
<p>۴۰</p> <p>آموزش جاوا (Java) قسمت ۴۰ : پکیج(Packag)...</p>





رایگان

۴۱

آموزش جاوا (Java) قسمت ۴۱ : معرفی داده س ...



رایگان

۴۲

آموزش جاوا (Java) قسمت ۴۲ : داده ساختار B...



رایگان

۴۳

آموزش جاوا (Java) قسمت ۴۳ : داده ساختار V...



رایگان

۴۴

آموزش جاوا (Java) قسمت ۴۴ : داده ساختار S...



رایگان

۴۵

آموزش جاوا (Java) قسمت ۴۵ : داده ساختار D...



رایگان

۴۶

آموزش جاوا (Java) قسمت ۴۶ : داده ساختار H...



رایگان

۴۷

آموزش جاوا (Java) قسمت ۴۷ : داده ساختار P...



رایگان

۴۸

آموزش جاوا (Java) قسمت ۴۸ : معرفی و آمو...



رایگان

۴۹

آموزش جاوا (Java) قسمت ۴۹ : برنامه نویسی...



رایگان

۵۰

آموزش جاوا (Java) قسمت ۵۰ : برنامه نویسی...



رایگان

۵۱

آموزش جاوا (Java) قسمت ۵۱ : استفاده از Lam.



رایگان

۵۲

آموزش جاوا (Java) قسمت ۵۲ : Thread و Mu...



رایگان

۵۳

آموزش جاوا (Java) قسمت ۵۳ : Runnable در ...



رایگان

۵۴

آموزش جاوا (Java) قسمت ۵۴ : کلاس ...



رایگان

زمان و قیمت کل

”

°

نظر °

هیچ نظری ارسال نشده است! اولین نظر برای این مطلب را شما ارسال کنید...

نظر شما

برای ارسال نظر باید وارد شوید.

از سرتاسر توسینسو

