بسم اللّه الرحمن الرحیم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

بررسی خطای جاوا

گردآورنده : مهندس افشین رفوآ

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 561 - واحد 7
88146323 - 88446780 - 88146330

بررسی خطای جاوا

در حالت کلی خطاها در دو دسته قرار می گیرند: خطاهای طراحی زمان (Design-time) و خطاهای منطقی (Logical). متوقف کردن خطاهای Design-time آسان می باشد، زیرا NetBeans زیر آنها خط می کشد. اگر خطا مانع اجرای برنامه شود، NetBeans زیر آن را با قرمز خط می کشد. خطاهای Logical خطاهایی هستند که شما به عنوان یک برنامه نویس ایجاد می کنید. برنامه اجرا خواهد شد، اما از آنجایی که شنا در برنامه نویسی اشتباه کرده اید، باعث می شود که کل برنامه دچار مشکل شود. به طور مختصر مثال هایی از خطاهای زمان اجرا را مشاهده خواهید کرد. همچنین چگونگی بررسی آنها را نیز فراخواهید گرفت. اما ابتدا به چگونگی رسیدگی جاوا به این خطاها خواهیم پرداخت.

استثنائات (Exceptions)

در جاوا خطاها توسط یک آبجکت Exception مورد بررسی قرار می گیرند. گفته می شود که Exception وارد می شوند و کار شما گرفتن آنها می باشد. شما می توانید این کار را با بلوک try … catch انجام دهید. بلوک try … catch مانند زیر می باشد.

```
try {
        }
        catch ( ExceptionType error_variable ) {
        }
```

قسمت try از try … catch به معنای امتحان کردن این کد می باشد. اگر اشتباهی رخ دهد، جاوا وارد بخش catch می شود. در این بخش جاوا موارد بین پرانتزها را بررسی می کند تا رسیدگی به خطاها را کنترل کند. اگر نوع اصلاح سازی Exception را داشته باشید، هر موردی که بین کروشه های catch می باشد، اجرا خواهد شد. اگر نوع اصلاح سازی Exception را نداشته باشید، جاوا از مورد پیش فرض برای نمایش یک پیغام خطا استفاده می کند.

به عنوان مثال، یک برنامه ی جدید کنسول (console application) ایجاد کنید. آن را هر چه میلید نام گذاری کنید. در کد مربوط به متود Main کد زیر را وارد کنید.

```
try {
        int x = 10;
        int y = 0;
        int z = x / y;
        System.out.println( z );
        }
        catch ( Exception err ) {
        System.out.println( err.getMessage( ) );
        }
```

در بخش try از try … catch، سه عدد صحیح x، y و z را تنظیم کرده ایم. سعی داریم y را به x تقسیم کنیم و سپس پاسخ را چاپ کنیم.

اگر اشتباهی رخ بدهد، بخش catch موجود می باشد. بین پرانتزهای این بخش عبارت زیر موجود می باشد.

```
Exception err
```

نوع Exception که استفاده می کنید، در ابتدا قرار می گیرد. در این مورد از آبجکت Exception error استفاده می کنیم. این آبجکت یک نوع "catch all" از Exception می باشد و تمرین خوبی برای برنامه نویسی نیست. ما آن را در یک لحظه به یک نوع خاص تغییر خواهیم داد.

پس از نوع Exception خود، یک فاصله و سپس نام یک متغیر را دارید. ما متغیر خود را err نامیده ایم، اما شما می توانید به دلخواه خود آن را نامگذاری کنید.

در کروشه های catch، یک عبارت چاپ داریم. اما به آنچه بین پرانتزهای println وجود دارد دقت کنید.

```
err.getMessage( )
```

getMessage متودی است در دسترس آبجکت های Exception. همانطور که از نام آن پیداست، پیغام خطای مربوط به Exception را دریافت می کند.

برنامه ی خود را اجرا کرده و آن را امتحان کنید. کد شما باید مشابه کد زیر باشد.

```java
package errorhandling;

public class errorChecking {

    public static void main(String[] args) {

        try {
            int x = 10;
            int y = 0;
            int z = x / y;

            System.out.println( z );
        }
        catch (Exception err) {
            System.out.println(err.getMessage());
        }
    }
}
```

و پنجره ی Output باید مورد زیر را نمایش دهد.

```
run:
        / by zero
        BUILD SUCCESSFUL (total time: 1 second)
```

خود خطا، خطایی که به وسیله ی getMessage تولید شد، خط وسط می باشد.

```
/ by zero
```

به عبارت دیگر یک تقسیم به وسیله ی خطای صفر. جاوا به شما اجازه ی تقسیم عدد به صفر را نمی دهد، از این رو پیغام خطا می دهد.

کد خود را به شکل زیر تغییر دهید.

```java
double x = 10.0;
        double y = 0.0;
        double z = x / y;
```

بقیه ی کد بدون تغییر می باشند. برنامه ی خود را اجرا کرده و آن را امتحان کنید.

مجددا یک پیغام خطا در پنجره ی Output نمایش داده خواهد شد که به شکل زیر می باشد.

```
run:
        Infinity
        BUILD SUCCESSFUL (total time: 1 second)
```

این بار به خاطر اینکه نتیجه یک عدد خیلی بزرگ می باشد، جاوا برنامه را متوقف خواهد کرد.

خطاهایی که شامل اعداد می شوند، نباید توسط یک نوع "catch all" Exception بررسی شوند. یک نوع خاص به نام ArithmeticException وجود دارد. لغت Exception را از بین پرانتزهای بلوک catch حذف کنید. ArithmeticException را جایگزین آن کنید. اکنون برنامه را دوباره اجرا کنید.

نباید تفاوتی با خطای نمایش داده شده در پنجره ی Output مشاهده کنید. اما با محدود کردن نوع خطای مورد انتظار، در حال انجام یک تمرین برنامه نویسی خوب می باشید.

در بخش بعد به Stack Trace خواهیم پرداخت.