

به نام خدا

آموزش پنجاه و پنجم

اهداف آموزشی این قسمت عبارتند از:

۱. معرفی Exception Object

۲. آشنایی با مفهوم Handle کردن

۳. نحوه حذف یک Exception

در آموزش گذشته پروژه ای طراحی کردیم که در آن یک `ArithmeticException` وجود داشت بنابراین برنامه ما `Crash` کرد.

اکنون بینیم اگر همین برنامه را به استفاده از دستورات `try, catch, finally` بنویسیم چه اتفاقی خواهد افتاد:

```
public class ExceptionsInJava {  
    public static void main(String[] args) {  
        System.out.println("This is my first output!");  
        try {  
            int numberOne = 10;  
            int numberTwo = 0;  
            int result = numberOne / numberTwo;  
            System.out.println(result);  
        } catch (ArithmeticException myProblem) {  
            System.out.println(myProblem);  
        } finally {  
            System.out.println("This is my final output!");  
        }  
    }  
}
```

همانطور که می بینیم ابتدا دستوری که قرار است عبارت `This is my first output!` را در پنجره Console به نمایش در آورد را نوشته ایم. سپس دستوری که در کد قبل مشکل را بود را داخل `try` نوشته ایم. بلافاصله پس از دستور `try` دستور `catch` آمده است که این وظیفه را دارا است تا اگر در دستور `try` مشکلی وجود داشت آن مشکل را به کاربر نشان دهد و از `Crash`

دوره آموزش جاوا

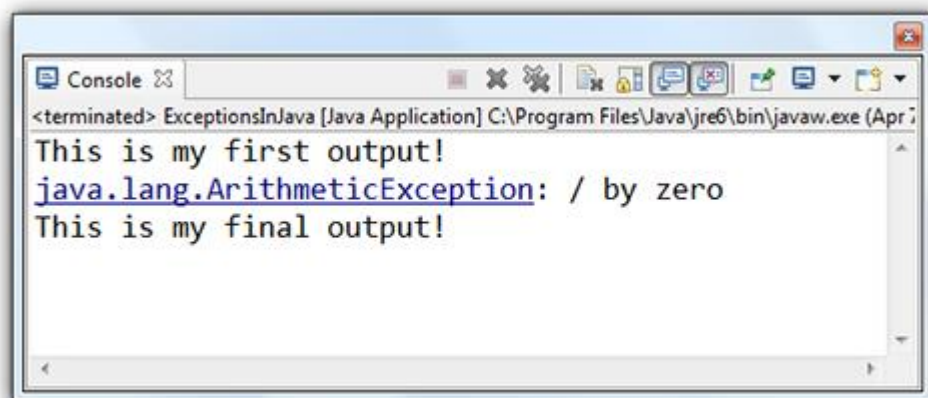
کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

کردن برنامه جلوگیری به عمل آورد. همانطور که ملاحظه می شود پارامتری که برای دستور catch در نظر گرفته شده است کلاسی تحت عنوان ArithmeticException است و نامی هم که برای شیء ساخته شده از روی این کلاس در نظر گرفته ایم myProblem به معنی "مشکل من" است (لازم به ذکر است اکثر برنامه نویسان دنیا برای نام کلاس های مرتبط با Exception نام e را در نظر می گیرند که حرف اول Exception می باشد اما به منظور جلوگیری از سردرگمی برنامه نویسان مبتدی در این آموزش نامی گویاتر همچون myProblem در نظر گرفته شده است).

سپس با استفاده از دستور System.out.println و قرار دادن نام شیء ساخته شده از روی کلاس ArithmeticException داخل این دستور از برنامه خود می خواهیم که چنانچه مشکلی در دستور try بود آن را با استفاده از این دستور نمایش دهد. در خاتمه در دستور finally دستوری که قصد داریم در پایان برنامه اجرا شود را می نویسیم که این دستور همان به نمایش در آوردن عبارت This is my final output! است.

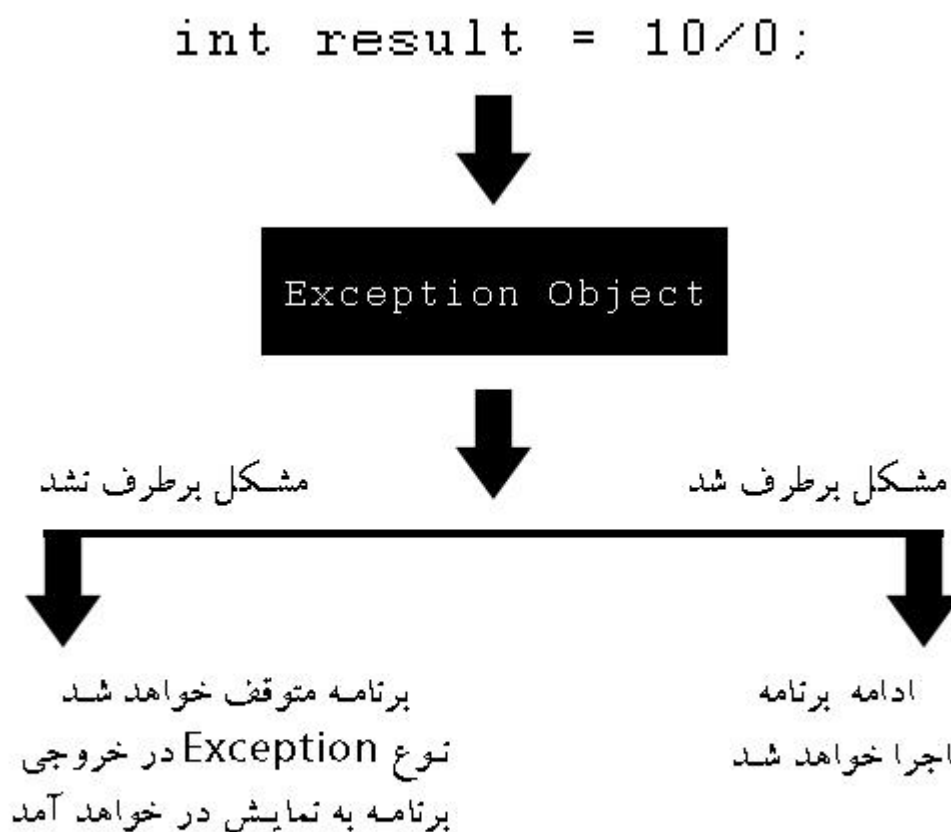
مجدد برنامه خود را اجرا می کنیم:



می بینیم که دستور اول که مسئول به نمایش در آوردن عبارت This is my first output! بود به نمایش در آمده است. حال نوبت به اجرای دستور داخل try می رسد و از آنجا که دستور داخل try یک دستور غیر منطقی است برنامه مستقیماً وارد دستور catch می شود و همانطور که می بینیم نوع Exception نوشته شده است و مقابل آن عبارت / by zero آمده است. چنانچه بخواهیم این عبارت را به زبان فارسی ترجمه کنیم بایستی گفت "یک مشکل محاسباتی بوجود آمده است که علت آن هم تقسیم شدن عددی بر صفر است". در نهایت برنامه علیرغم اینکه دارای

یک مشکل است اما به هیچ وجه Crash نکرده و ادامه پیدا می کند تا جاییکه دستور به نمایش در آوردن عبارت `This is my final output!` هم اجرا خواهد شد.

در صورتیکه بخواهیم اجرای برنامه خود را به صورت یک نمودار نشان دهیم، می توانیم نموداری همانند نمودار زیر در نظر بگیریم:



همانطور که در نمودار فوق مشخص است در صورت وجود یک دستور نادرست از لحاظ ریاضیاتی مثل تقسیم کردن عدد ده بر عدد صفر یک `Exception Object` ایجاد می شود که به ماشین مجازی جاوا یا JVM فرستاده می شود. حال دو اتفاق ممکن است روی دهد: اگر این `Exception` اصطلاحاً `Handle` شود یا رفع شود برنامه به کار خود ادامه خواهد داد و مابقی دستورات اجرا خواهند شد (مورد سمت راست) و اگر هم که این `Exception` اصطلاحاً `Handle` نشود برنامه متوقف خواهد شد، در پنجره `Console` نوع `Exception` به نمایش در خواهد آمد و مابقی دستورات داخل برنامه به هیچ وجه اجرا نخواهند شد.

دوره آموزش جاوا

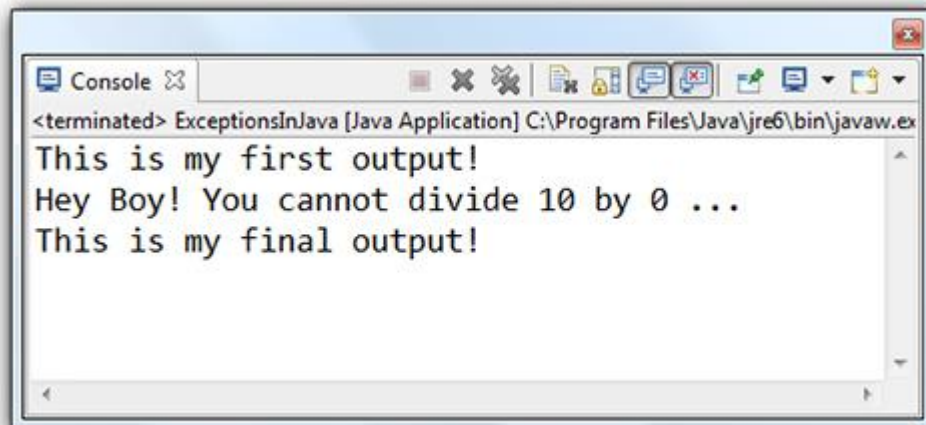
کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

به نظر می رسد پس از اجرای برنامه خود در مرحله دوم کاملاً به اهمیت استفاده از دستورات `try`, `catch`, `finally` بیشتر پی ببریم. همانطور که دیدیم برنامه بدون آنکه `Crash` کند اجرا شده و در جایی از کد هم که مشکلی وجود داشت آن مشکل به اطلاع کاربر رسید. اکنون فرض کنیم که در صورت وجود یک `Exception` در برنامه خود نمی خواهیم که نوع `Exception` و دلیل آن در پنجره `Console` به نمایش در آید بلکه قصد داریم چنانچه برنامه مشکلی داشت جمله ای را به نمایش در آوریم. برای این منظور کد خود را به صورت زیر ویرایش می کنیم:

```
public class ExceptionsInJava {  
  
    public static void main(String[] args) {  
  
        System.out.println("This is my first output!");  
        try {  
            int numberOne = 10;  
            int numberTwo = 0;  
            int result = numberOne / numberTwo;  
            System.out.println(result);  
  
        } catch (ArithmeticException myProblem) {  
            System.out.println("Hey Boy! You cannot divide 10 by 0 ...");  
        } finally {  
            System.out.println("This is my final output!");  
        }  
    }  
}
```

می بینیم که داخل دستور `System.out.println` یی که داخل `catch` قرار گرفته است به جای نوشتن نام شیئی ساخته شده از روی کلاس `ArithmeticException` عبارت `Hey Boy!` `You cannot divide 10 by 0...` به معنی "هی پسر! تو نمی توانی عدد ده رو بر صفر تقسیم کنی ..." را نوشته ایم. یک بار دیگر برنامه خود را اجرا می کنیم:



می بینیم که برنامه ما پس از اجرا با مشاهده کردن مشکلی در کد مرتبط با try عبارتی را نمایش خواهد داد که در دستور catch نوشته ایم. حال فرض کنیم که به هیچ وجه نمی خواهیم برنامه ما پس از مشاهده مشکلی در کد مرتبط با try چیزی در پنجره Console نمایش دهد. برای این منظور کد فوق را به صورت زیر بازنویسی می کنیم:

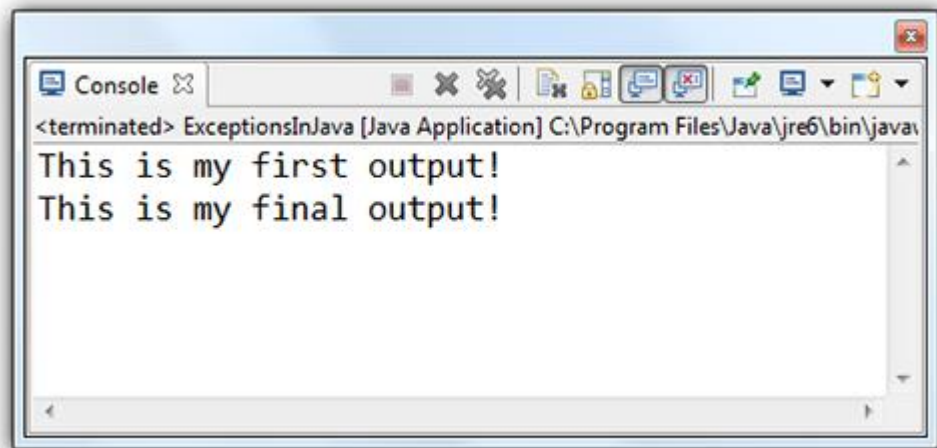
```
public class ExceptionsInJava {  
  
    public static void main(String[] args) {  
  
        System.out.println("This is my first output!");  
        try {  
            int numberOne = 10;  
            int numberTwo = 0;  
            int result = numberOne / numberTwo;  
            System.out.println(result);  
  
        } catch (ArithmeticException myProblem) {  
        } finally {  
            System.out.println("This is my final output!");  
        }  
    }  
}
```

در واقع کاری که در کد بالا انجام داده ایم این است که دستوری که داخل {} مرتبط با catch بود را از برنامه حذف کرده ایم. با اینکار به برنامه خود دستور می دهیم که چنانچه مشکلی در دستور try بود آن مشکل را در دستور catch بررسی کند اما به هیچ وجه چیزی مرتبط با دستور catch در پنجره Console به نمایش در نیاورد. حال مجدد برنامه خود را اجرا می کنیم:

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی



همانطور که ملاحظه می شود علیرغم اینکه برنامه ما دارای یک Exception است اما برنامه Crash نکرده و چیزی هم حاکی از آنکه برنامه دارای مشکلی است به اطلاع کاربر نخواهد رسید. پس از مطالعه این آموزش انتظار می رود بتوانیم به سؤالات زیر پاسخ بدهیم:

۱. منظور از Exception Object چیست؟
۲. مفهوم ArithmeticException چیست؟
۳. به چه شکل می توان یک Exception را از خروجی برنامه حذف کرد؟

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی