

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

خطاهای منطقی در جاوا

گردآورنده : مهندس افشین رفوآ

خطاهای منطقی در جاوا

خطاهای منطقی خطاهایی هستند که شما به عنوان یک برنامه نویس دچار آنها می شوید، وقتی کد به شکلی که انتظار دارید اجرا نمی شود. بررسی این خطاها ممکن است دشوار باشد. خوشبختانه **NetBeans** دارای ابزاری داخلی برای کمک به شما در کنترل این مشکل می باشد.

ابتدا کد زیر را امتحان کنید.

```
package errorhandling2;

public class errorChecking2 {

    public static void main(String[] args) {
        int LetterCount = 0;
        String check_word = "Debugging";
        String single_letter = "";
        int i;

        for (i = 0; i < check_word.length(); i++) {

            single_letter = check_word.substring(i, i+1);

            if (single_letter.equals("g")) {
                LetterCount++;
            }
        }

        System.out.println("G was found " + LetterCount + " times.");
    }
}
```

با استفاده از برنامه ای که دارید یا با شروع یک پروژه ی جدید، آن را برای خود تایپ کنید. کاری که در اینجا سعی داریم انجام دهیم شمردن حرف **g** در لغت **"Debugging"** می باشد. مشخصا پاسخ **3** می باشد. به هر حال وقتی برنامه را اجرا می کنید، پنجره ی **Output** عبارت زیر را چاپ می کند.

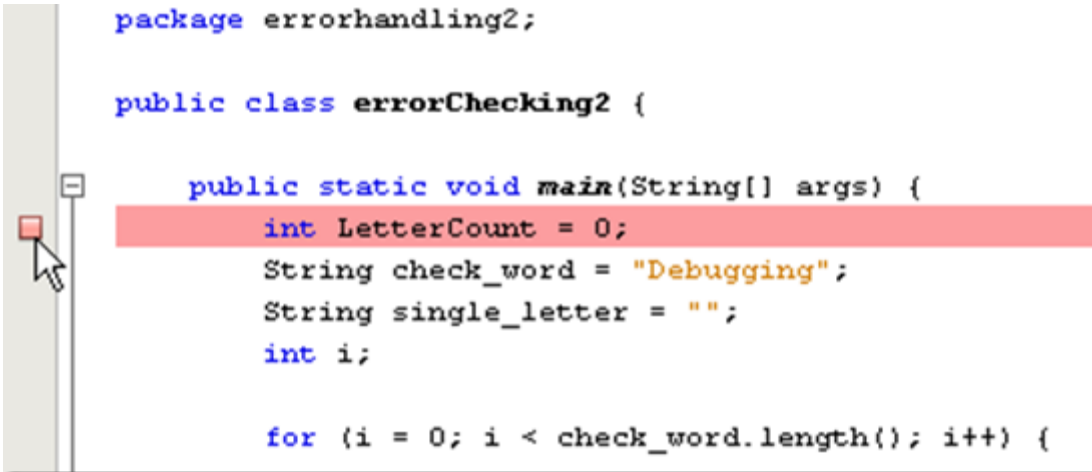
"G was found 0 times."

بنابراین ما جایی در کد خود خطایی انجام داده ایم. اما کجا؟ برنامه خوب اجرا می شود و

هیچ **Exceptions** برای ما در پنجره ی **Output** ارائه نمی دهد. بنابراین ما چه کار می توانیم بکنیم؟

برای بررسی مشکلات مربوط به کد خود، NetBeans به شما اجازه ی اضافه کردن چیزی به نام Breakpoint را می دهد.

برای افزودن یک Breakpoint جدید، روی حاشیه ی پنجره ی مربوط به کد کلیک کنید.



```
package errorhandling2;

public class errorChecking2 {

    public static void main(String[] args) {
        int LetterCount = 0;
        String check_word = "Debugging";
        String single_letter = "";
        int i;

        for (i = 0; i < check_word.length(); i++) {
```

از منوی NetBeans، روی **Debug > Debug errorhandling2** کلیک کنید (یا هر آنچه پروژه ی خود را نامیده اید). NetBeans وارد breakpoint خواهد شد. این برنامه حالا اجرای کد را متوقف کرده است. همچنین باید یک نوار ابزار ظاهر شده ی جدید مشاهده کنید.



سه دکمه ی اول به شما اجازه ی متوقف کردن کامل یا موقت و ادامه دادن بخش عیب زدایی (debugging) را می دهد. پنج دکمه ی دیگر اجازه ی وارد شدن به کد، رد شدن از کد، خروج از کد یا رسیدن به مکان نما می دهد.

می توانید با فشار دادن دکمه **F5** ادامه دهید. کد با **Breakpoint** نیز باید مانند قبل طبیعی اجرا شود. بنابراین بخش عیب زدایی به پایان خواهد رسید.

وقتی بخش عیب زدایی به پایان می رسد، در **Breakpoint** روی **back** کلیک کنید تا از آن رهایی یابید. اکنون به **for loop** یک **Breakpoint** اضافه کنید.

```

public class errorChecking2 {

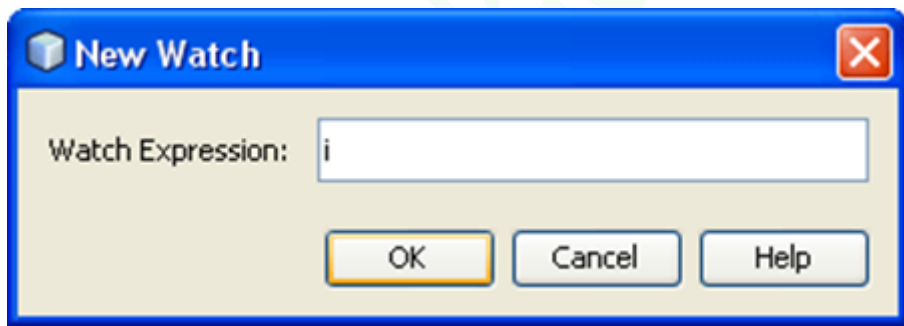
    public static void main(String[] args) {
        int LetterCount = 0;
        String check_word = "Debugging";
        String single_letter = "";
        int i;

        for (i = 0; i < check_word.length(); i++) {

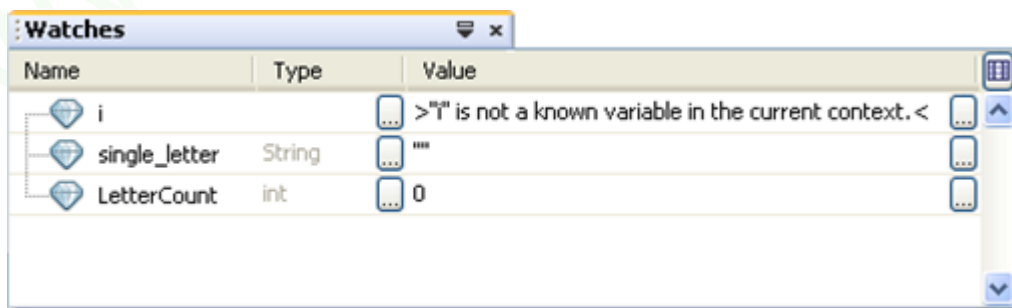
            single_letter = check_word.substring(1, 1);
        }
    }
}

```

اکنون روی **New Watch > Debug** کلیک کنید. یک **Watch** به شما اجازه ی پیگیری آنچه در یک متغیر است را می دهد. بنابراین حرف **i** در دیالوگ باکس **Watch** تایپ کرده و روی **OK** کلیک کنید.



یک **Watch** دیگر اضافه کرده و سپس **single_letter** را تایپ کنید. روی **OK** کلیک کنید. **Watch** سوم را اضافه کرده و **LetterCount** را تایپ کنید. شما باید سه متغیر زیر را در پایین صفحه ی خود داشته باشید.



اکنون در نوار ابزار روی آیکن **Step Into** فشار دهید.



یا فقط دکمه ی **F7** را فشار دهید. به فشار دادن دکمه ی **F7** ادامه دهید تا مشاهده کنید در پنجره ی **Watch** چه اتفاقی می افتد. باید متوجه شده باشید که متغیر **i** هر بار **1** واحد افزایش می یابد. اما دو متغیر دیگر بدون تغییر باقی می مانند.

Name	Type	Value
i		1
single_letter	String	""
LetterCount	int	0

از آنجایی که در متغیر تک حرفی چیزی وجود ندارد، بنابراین **LetterCount** نمی تواند بیشتر از صفر باشد. بنابراین مشکل خود را یافته ایم - استفاده از یک رشته ی زیرمجموعه (**substring**) ممکن است اشتباه باشد، چرا که هیچ کاراکتری نمی گیرد.

بخش عیب زدایی را متوقف کنید و خط رشته ی زیر مجموعه (**substring**) را مانند زیر تغییر دهید.

```
single_letter = check_word.substring( i, i + 1 );
```

اکنون عیب زدایی را دوباره آغاز کنید. به فشار دادن دکمه ی **F7** ادامه دهید تا روی خطوط مربوط به **for** **loop** بروید. این بار باید تغییر متغیرهای **single_letter** و **LetterCount** را مشاهده کنید.

وقتی کد به پایان می رسد، باید پنجره ی **Output** را مشاهده کنید که عبارت زیر را نمایش می دهد.

```
"G was found 3 times."
```

اکنون ما پاسخ صحیح را داریم.

بنابراین اگر موارد طوریکه با کد شما طراحی شدند، پیش نمی روند، تنظیم یک **Breakpoint** و چند **Watch** را برای متغیرهای خود امتحان کنید. سپس یک بخش اشکال زدایی را آغاز کنید.

در بخش بعد به بررسی یک موضوع متفاوت خواهیم پرداخت: چگونگی باز کردن فایل های متن در جاوا.

www.tahlildadeh.com