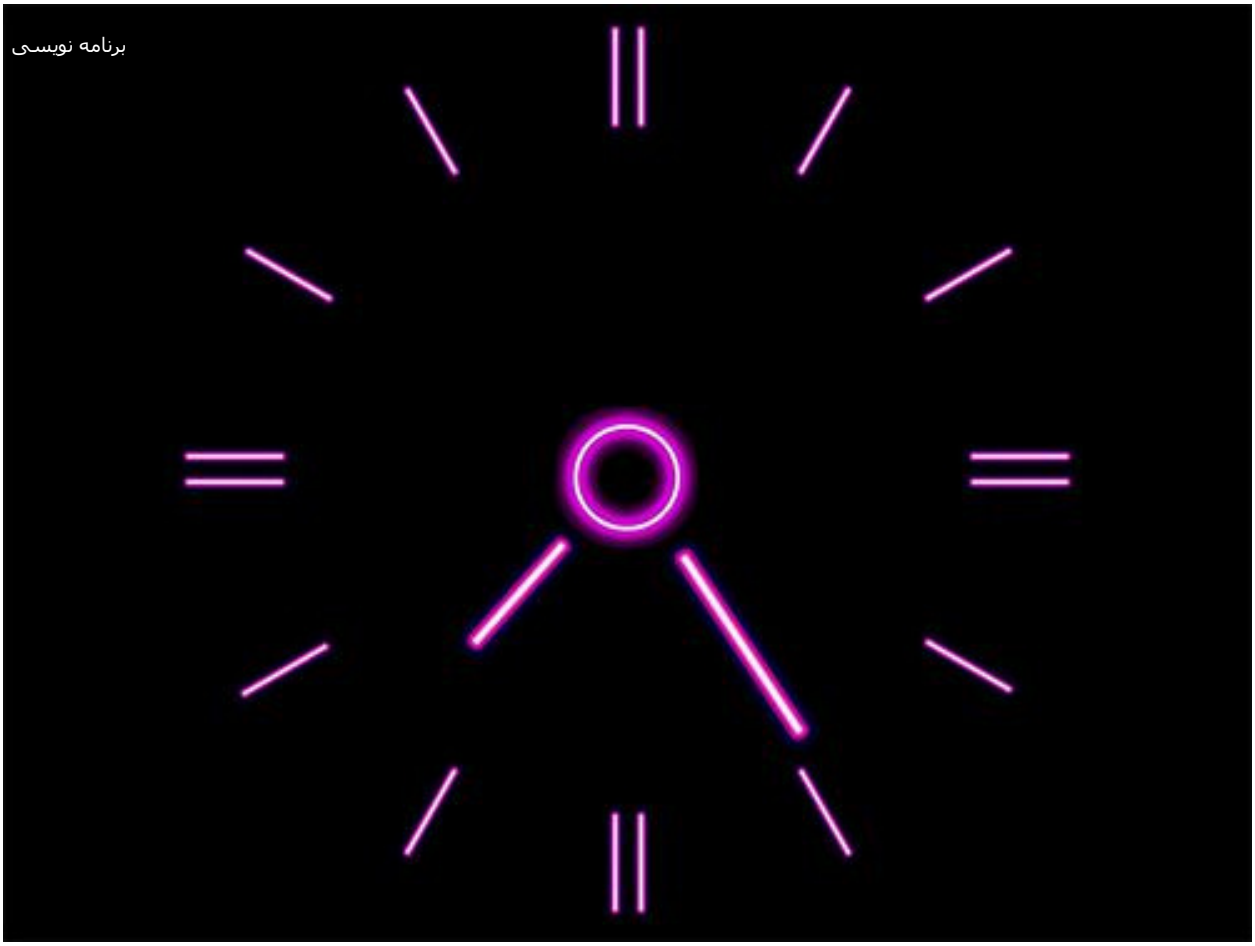


آموزش برنامه نویسی جاوا – قسمت ۵۰: معرفی Constant ها در زبان برنامه نویسی جاوا

بهزاد مرادی - چهارشنبه ۱ آبان ۱۳۹۲ - ۱۷:۰۰

اهداف آموزشی این قسمت عبارتند از:
1. معرفی Constant ها در زبان برنامه نویسی جاوا
2. نحوه ایجاد یک Constant

در آموزش قسمت چهل و نهم دیدیم که به چه نحوی می توان یک متد را final کرد. از سوی دیگر به این نکته اشاره کردیم که علاوه بر امکان استفاده از کلید واژه final به همراه متدها، این امکان برای برنامه نویس زبان جاوا فراهم شده که بتواند از کلید واژه final در کنار نام کلاس ها و متغیرها هم استفاده کند. در این آموزش قصد داریم تا نحوه به کارگیری کلید واژه final را در کنار متغیرها مورد

بررسی قرار دهیم.

برای شروع کار پروژه ای تحت عنوان 50th Session ایجاد کرده و کلاسی در آن به نام JavaConstants ایجاد می کنیم (به خاطر داشته باشیم که در حین ساخت این کلاس گزینه public static void main را تیک دار کنیم):

```
public class JavaConstants {
	public static void main(String[] args) {
	}
}
```

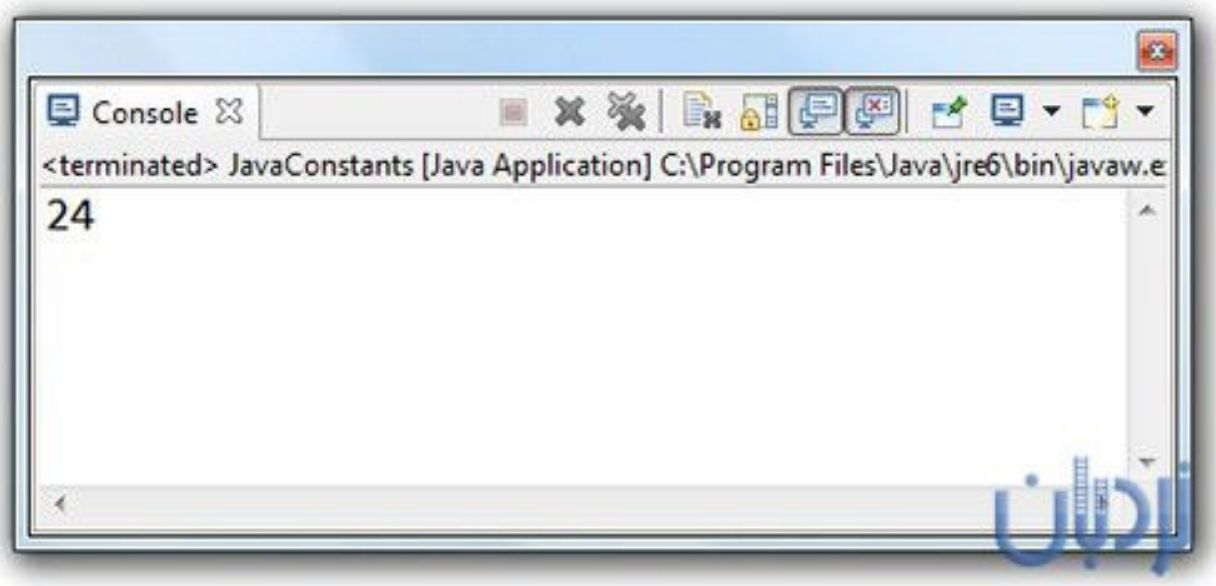
در دنیای واقعی همواره چیزهایی وجود دارند که مقادیر آنها ثابت است که از آن جمله می توان به تعداد ساعات یک روز اشاره کرد که همواره 24 ساعت است و یا تعداد اضلاع یک مربع را مد نظر قرار داد که هیچ وقت بیش از 4 ضلع نخواهد بود. در ادامه آموزش تعداد ساعات یک روز را مد نظر قرار داده و برنامه خود را بر اساس آن می نویسیم:

```
public class JavaConstants {
	public static void main(String[] args) {
		int numberOfHoursInADay = 24;
	}
}
```

در کد فوق یک متغیر از جنس int تحت عنوان numberOfHoursInADay به معنی "تعداد ساعات یک روز" ایجاد کرده و Value اولیه آن را معادل با 24 قرار داده ایم. حال می خواهیم این متغیر را روی صفحه مانیتور نمایش دهیم. برای همین منظور کد فوق را به صورت زیر تکمیل می کنیم:

```
public class JavaConstants {
	public static void main(String[] args) {
		int numberOfHoursInADay = 24;
		System.out.println(numberOfHoursInADay);
	}
}
```

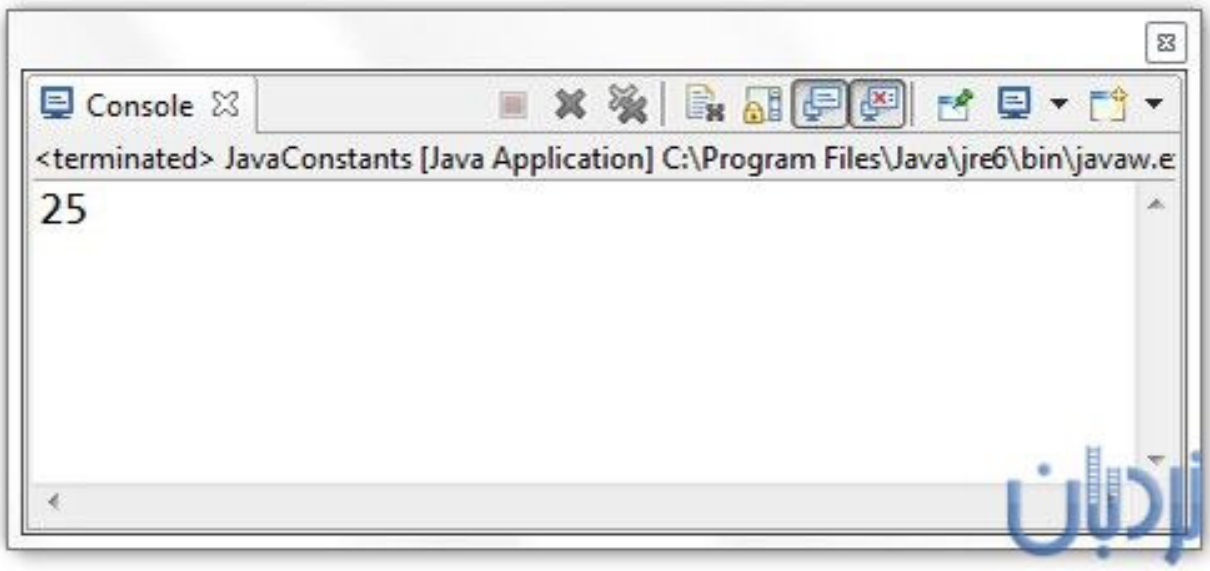
به طور خلاصه با نوشتن دستور System.out.println و قرار دادن نام متغیر خود در آن این امکان را خواهیم داشت که مقدار اولیه متغیر خود را در پنجره Console به نمایش در آوریم:



همانطور که ملاحظه می شود عدد 24 به نمایش در آمد اما سوال اینجا است که آیا همانطور که در دنیای واقعی تعداد ساعات یک روز 24 ساعت بوده و غیر قابل تغییر است، آیا در این برنامه ای هم که نوشتیم همین طور است یا خیر. برای این منظور برای تست کردن این مسئله کد خود را به صورت زیر تغییر می دهیم:

```
public class JavaConstants {
        public static void main(String[] args) {
                int numberOfHoursInADay = 24;
                numberOfHoursInADay = 25;
                System.out.println(numberOfHoursInADay);
        }
}
```

همانطور که ملاحظه می شود پس از اختصاص یک Value به متغیر خود مجدد نام متغیر را نوشته و این بار مقداری جدیدی برای آن در نظر گرفته ایم. حال مجدد برنامه خود را اجرا می کنیم:

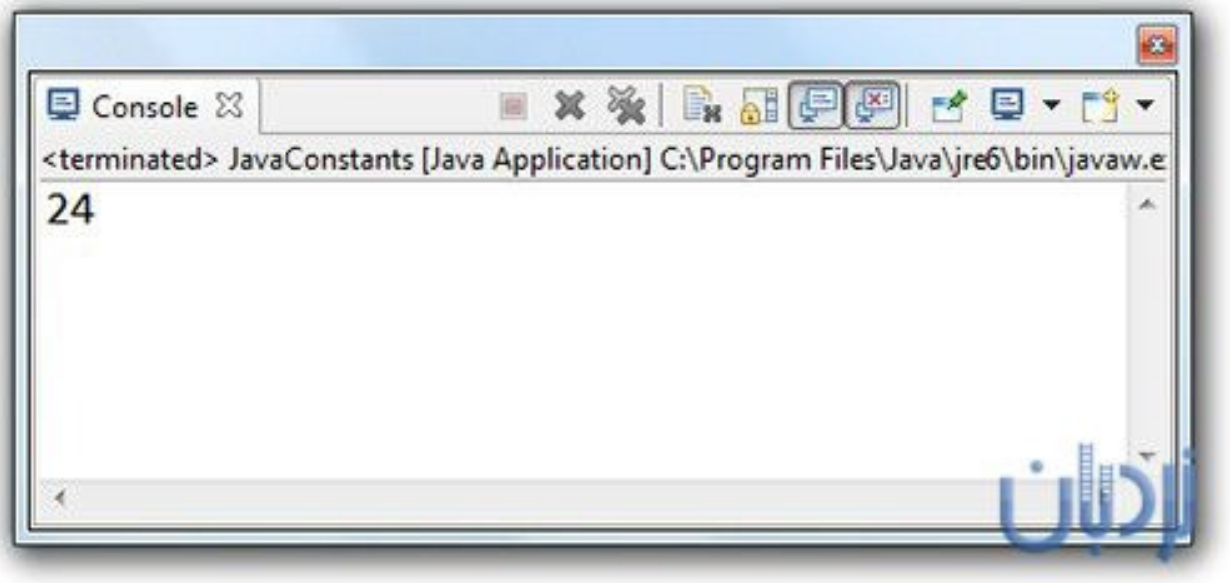


می بینیم که برنامه خوبی ننوشته ایم چرا که برخلاف دنیای واقعی، برنامه ما این امکان را فراهم آورده است که بتوانیم تعداد ساعات یک روز را تغییر دهیم.در حقیقت تنها راهی که از آن طریق می توانیم بر این مشکل غلبه کنیم این است از کلید واژه final استفاده کنیم. برای این منظور کد فوق را به صورت زیر بازنویسی می کنیم:

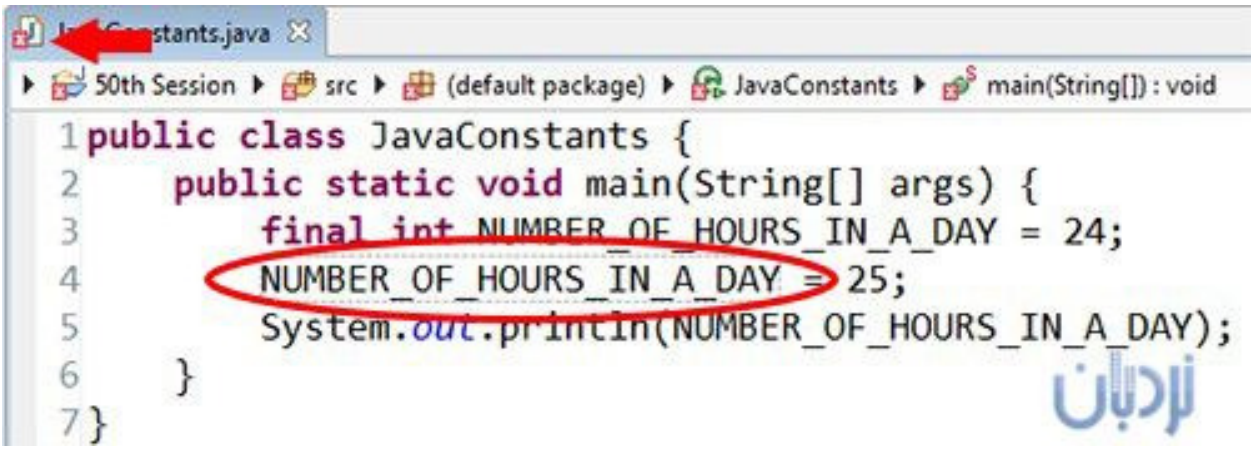
```
public class JavaConstants {
        public static void main(String[] args) {
                final int NUMBER_OF_HOURS_IN_A_DAY = 24;
                System.out.println(NUMBER_OF_HOURS_IN_A_DAY);
        }
}
```

در کد فوق دو تغییر عمده دیده می شود اول اینکه کلید واژه final را به متغیر خود اضافه کرده ایم که این تضمین را به ما خواهد داد که مقدار اختصاص داده شده به این متغیر هرگز تغییر نخواهد کرد، دوم آنکه نحوه نوشتن نام متغیر خود به صورت camelCase را تغییر داده و کلیه حروف را به صورت بزرگ نوشته و آنها را با یک _ از یکدیگر مجزا ساخته ایم (برای آشنایی بیشتر با مفهوم camelCase به آموزش دهم مراجعه نمایید). به خاطر داشته باشیم که نام متغیر خود داخل پرانتز در دستور System.out.println هم می بایست مطابق با نام جدید متغیر باشد.

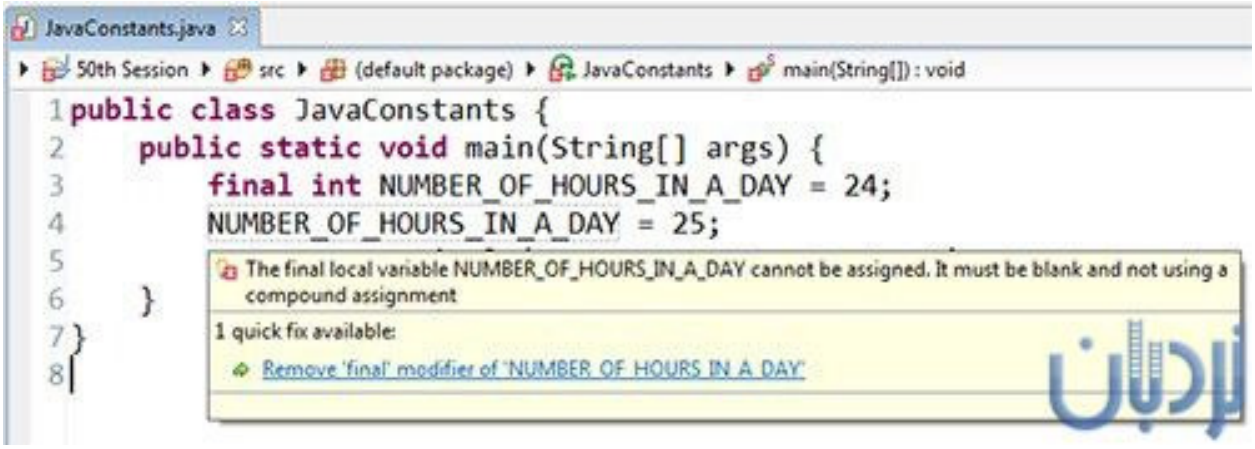
در این مرحله می توان گفت که ما موفق شده ایم که یک Constant در برنامه خود ایجاد کنیم. واژه انگلیسی Constant دارای معانی متعددی همچون ثابت، پایدار، دائمی، استوار و غیره است و به نظر می رسد طراحان زبان برنامه نویسی جاوا نام مناسبی برای متغیرهایی که همواره دارای مقدار ثابتی می باشند انتخاب کرده اند. از سوی دیگر علت تغییر در نوع نوشتن یک Constant نسبت به یک Variable در برنامه های جاوا این است که در برنامه خود بتوانیم به سادگی Constant ها را از Variable ها تشخیص دهیم و این یک سبک قراردادی در میان برنامه نویسان جاوا در سرتاسر جهان است اما این در حالی است که می توان برای Constant ها هم از همان سبک نامگذاری camelCase بدون هیچ مشکلی استفاده کرد. اکنون مجدد برنامه خود را اجرا می کنیم:



می بینیم که برنامه مثل قبل بدون هیچ مشکلی اجرا خواهد شد. اکنون برای آنکه ببینیم آیا می توان مقدار اولیه یک Constant و یا به عبارت دیگر یک متغیر از جنس final را تغییر داد یا خیر، کد خود را به صورت زیر بازنویسی می کنیم:

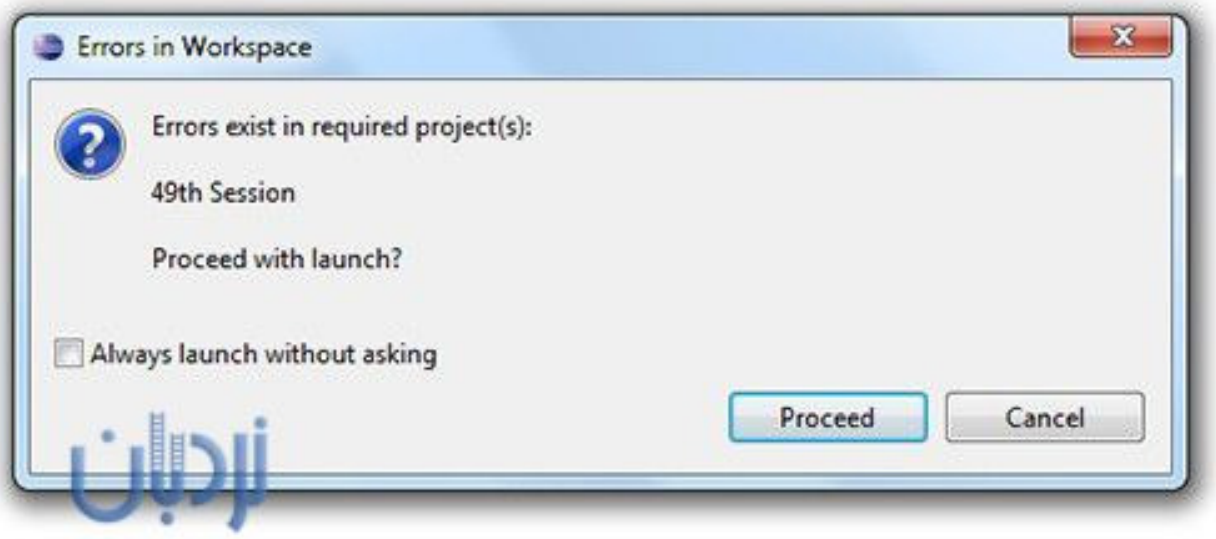


همانطور که در تصویر فوق با یک فلش قرمز رنگ مشخص شده است، در کنار نام کلاس این پروژه یک علامت ضربدر قرمز رنگ قرار گرفته است که این علامت نشانگر وجود مشکلی در برنامه ما است. حال اگر نگاهی به کدهای برنامه خود کنیم می بینیم که دور نام متغیر خود که مقدار جدیدی برای آن در نظر گرفته ایم یک نقطه چین قرار گرفته است. اکنون اگر نشانگر موس خود را روی آن نقطه چین قرار دهیم با تصویر زیر مواجه خواهیم شد:

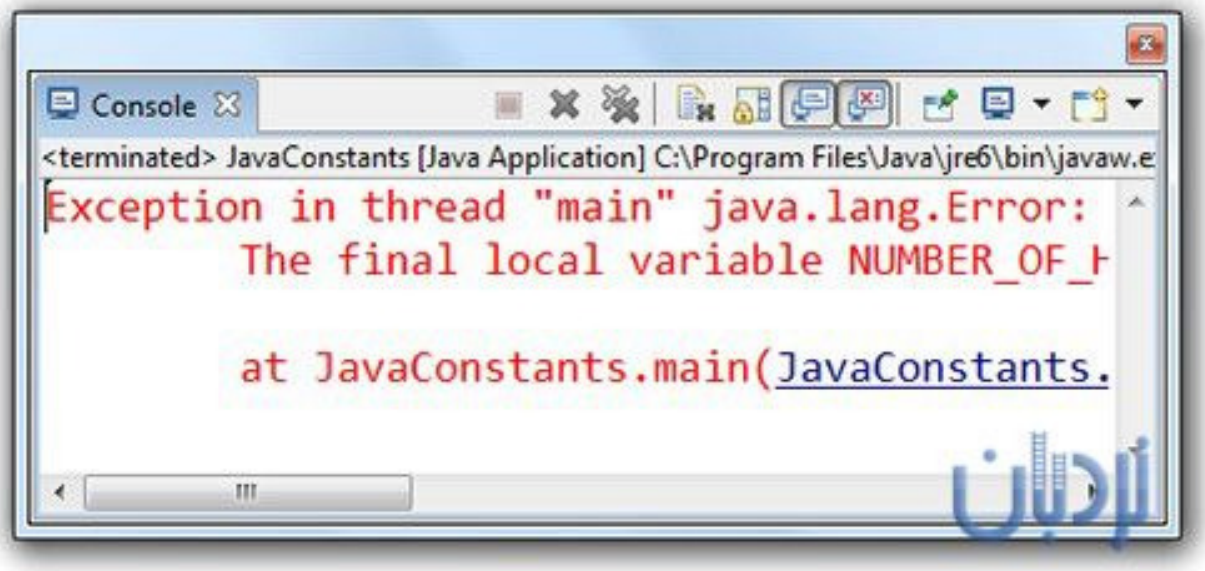


همانطور که در تصویر فوق مشخص است، یک باکس زرد رنگ نمایان می شود که حاکی از آن است که "مقدار جدید نمی توان به متغیر NUMBER_OF_HOURS_IN_A_DAY که از جنس final است اختصاص داد". راه کاری هم که اکلیپس به ما می دهد و با خط آبی رنگ نشان داده شده است این است که کلید واژه final را از کنار نام متغیر خود حذف کنیم تا مشکل برنامه برطرف گردد.

اکنون اگر با وجود این ایراد در برنامه اقدام به اجرای آن نماییم با تصویر زیر رو به رو خواهیم شد:



در این پنجره اکلیپس به ما اخطار می دهد که در پروژه خود دارای مشکلی هستیم اما امکان ادامه کار را نیز با وجود این مشکل برای ما فراهم کرده است. در واقع با کلیک بر وری کلید Proceed می توانیم برنامه خود را Compile کنیم که در این صورت اجرای برنامه ما به این شکل خواهد بود:



این پیغام خطا به طور خلاصه حاکی از آن است که برنامه به درستی Compile نشده است.

پیش از پایان این آموزش یادآوری چند نکته ضروری به نظر می رسد. در واقع اگر بخواهیم که در برنامه خود یک Constant داشته باشیم که توسط کلیه متدهای موجود در کلاس قابل دسترسی باشد پس نیاز است تا آن را در کلاس تعریف کنیم:

```
public class JavaConstants {
        final int NUMBER_OF_HOURS_IN_A_DAY = 24;
}
```

اما اگر بخواهیم یک Constant یی داشته باشیم که فقط در سطح یک متد قابل دسترسی باشد پس نیاز است تا آن را داخل متد خود تعریف کنیم:

```
public class JavaConstants {
        public void showConstant() {
                final int NUMBER_OF_HOURS_IN_A_DAY = 24;
        }
}
```

از سوی دیگر اگر بخواهیم که Constant ما توسط دیگر بخش های برنامه قابل دسترسی باشد می توان از کلید واژه public در کنار آن استفاده کرد:

```
public class JavaConstants {
        public final int NUMBER_OF_HOURS_IN_A_DAY = 24;
}
```

اگر بخواهیم Constant ما فقط محدود به کلاسی باشد که در آن Constant در آن قرار گرفته است می بایست از کلید واژه private استفاده کرد:

```
public class JavaConstants {
        private final int NUMBER_OF_HOURS_IN_A_DAY = 24;
}
```

و در پایان اگر بخواهیم که Value این Constant با دیگر بخش های برنامه خود به اشتراک گذاشته شود به صورت زیر از کلید واژه static استفاده خواهیم کرد:

```
public class JavaConstants {
        static final int NUMBER_OF_HOURS_IN_A_DAY = 24;
}
```

برای آشنایی بیشتر با سطوح دسترسی در زبان برنامه نویسی جاوا به آموزش به آموزش سی و ششم و برای آشنایی بیشتر با کلید واژه static با آموزش های چهل و هفتم و چهل و هشتم مراجعه نمایید.

پس از مطالعه این آموزش انتظار می رود بتوانیم به سؤالات زیر پاسخ بدهیم:
1. چرا در برنامه نویسی نیاز به Constant ها داریم؟
2. نحوه ایجاد یک Constant در جاوا چگونه است؟
3. چه فرقی مابین ایجاد یک Constant در سطح کلاس و یک Constant در سطح متد وجود دارد؟
4. اگر مقدار یک Constant را تغییر دهیم چه اتفاقی روی خواهد داد؟

برچسب ها: آموزش جاوا اموزش برنامه نویسی جاوا برنامه نویسی اندروید جاوا متغییرها

به ارتباط نزدیک با به دنیای جدید ...

بحث و نظرات

#262781

مریم | **مریم:** 2013-10-24 17:59

سلام از سایت بسیار عالیتون تشکر میکنم.
لطفا وب سرویس در جاوا رو هم آموزش بدهید.

2 رای

ثبت نام

برای گذاشتن نظر لطفا وارد شوید یا ثبت نام کنید.

#262912

2 رای

ثبت نام

arash_scg | **آرش:** 19:20 2013-10-25

باز هم عالی
ممنون وقتی که برای آموزش میزارید

برای گذاشتن نظر لطفا وارد شوید یا ثبت نام کنید.

#263006

0 رای

ثبت نام

kavous_kiani | **کاوس:** 12:22 2013-10-27

دوستان!

در اندروید استودیو چگونه می‌توانم یک پروژه‌ی جاوا بسازم؟

گزینه‌ای به نام پروژه‌ی جاوا نمی‌بینم.

وقتی نیو پراجکت را می‌زنم همان صفحه‌ای را می‌آورد که برای اندروید پروژه تعریف می‌کنیم.

لطفاً راهنایی کنید.

ممنون

.

برای گذاشتن نظر لطفا وارد شوید یا ثبت نام کنید.

#263656

0 رای

ثبت نام

forogh | forogh: 18:36 2014-01-16

این طور که من فهمیدم constant
رو واسه مواردی استفاده میکنیم که ثابت هستن مثل ساعت شب و روز یا اضلاع هندسی
و با قرار دادن کلمه فینال اون رو ثابت نگه می داریم

برای گذاشتن نظر لطفا وارد شوید یا ثبت نام کنید.

بیشتر بخوانید

آموزش برنامه نویسی جاوا - قسمت ۴۹: آشنایی با

آموزش برنامه نویسی جاوا - قسمت ۴۷: نحوه به

آموزش برنامه نویسی جاوا - قسمت ۴۸: نحوه به

همین الان به رایگان مشترک مطالب نردبان شوید تا اخبار داغ را از دست ندهید

Email:

۱- از ایمیل شما استفاده تبلیغاتی نخواهد شد.
۲- هر زمان که مایل باشید می توانید اشتراک ایمیلی خود را قطع کنید.

اشتراک مطالب از طریق فید

دنبال کنید: