

به نام خدا

آموزش پنجاهم

اهداف آموزشی این قسمت عبارتند از:

۱. معرفی Constant ها در زبان برنامه نویسی جاوا

۲.

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 ۲۴ ساعت است و یا تعداد اضلاع یک مربع را مد نظر قرار داد که هیچ وقت بیش از ۴ 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;  
    }  
}
```

دوره آموزش جاوا

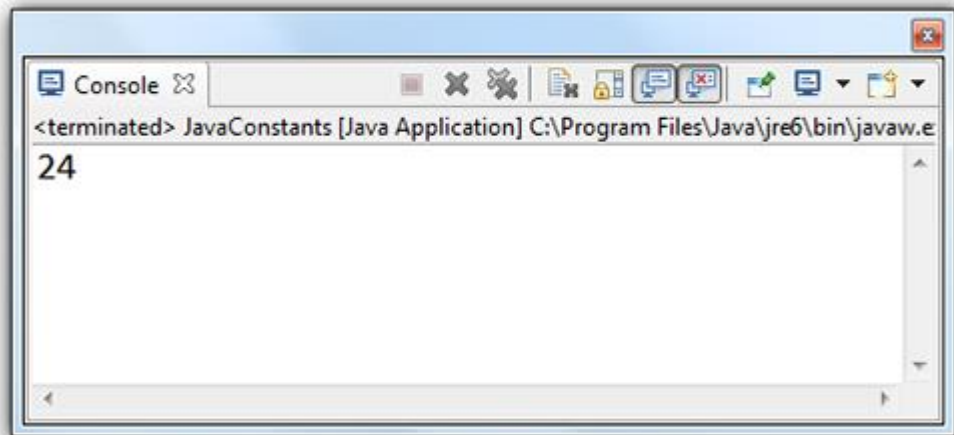
کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

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 ۲۴ 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);  
    }  
}
```

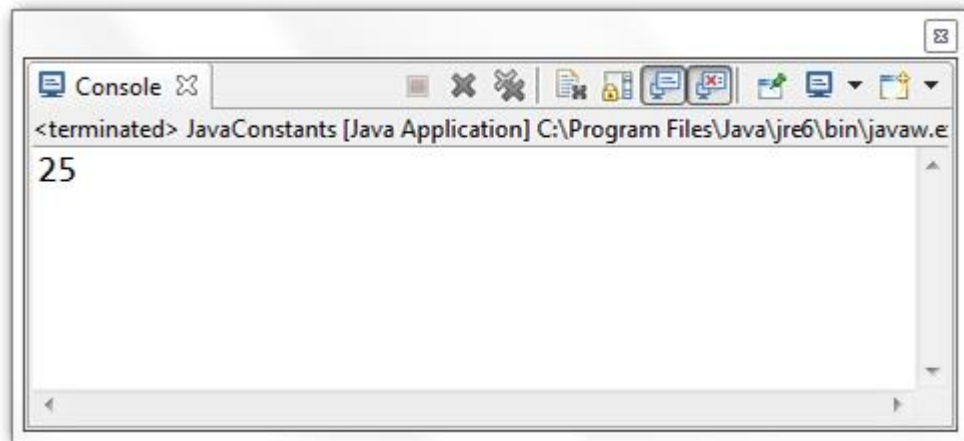
به طور خلاصه با نوشتن دستور `System.out.println` و قرار دادن نام متغیر خود در آن این امکان را خواهیم داشت که مقدار اولیه متغیر خود را در پنجره `Console` به نمایش در آوریم:



همانطور که ملاحظه می شود عدد ۲۴ به نمایش در آمد اما سوال اینجا است که آیا همانطور که در دنیای واقعی تعداد ساعات یک روز ۲۴ 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 نوشته 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);  
    }  
}
```

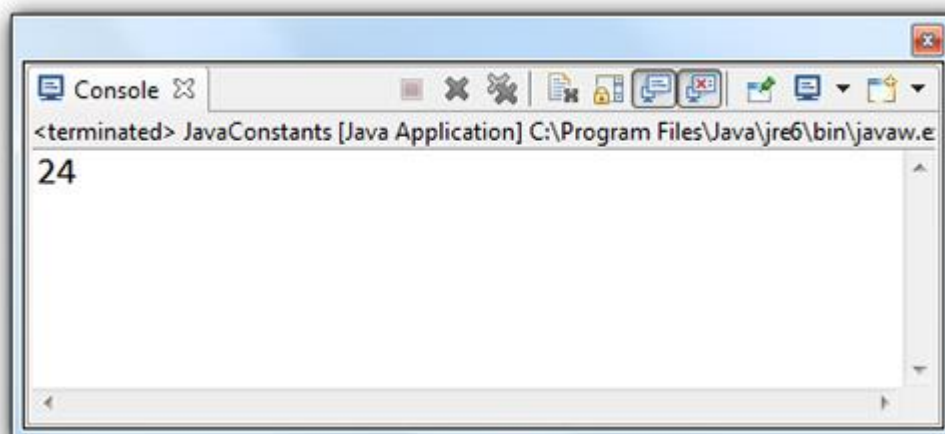
در کد فوق دو تغییر عمده دیده می شود اول اینکه کلید واژه final را به متغیر خود اضافه کرده ایم که این تضمین را به ما خواهد داد که مقدار اختصاص داده شده به این متغیر هرگز تغییر نخواهد کرد، دوم آنکه نحوه نوشتن نام متغیر خود به صورت camelCase را تغییر داده و کلیه حروف را به صورت بزرگ نوشته و آنها را با یک _ 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.

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

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:

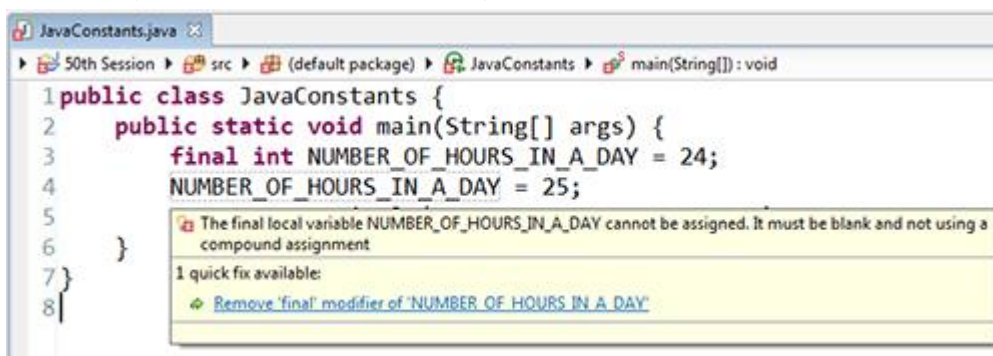
A screenshot of a Java IDE showing the code for 'Constants.java'. The code is as follows:
1 public class JavaConstants {
2 public static void main(String[] args) {
3 final int NUMBER_OF_HOURS_IN_A_DAY = 24;
4 NUMBER_OF_HOURS_IN_A_DAY = 25;
5 System.out.println(NUMBER_OF_HOURS_IN_A_DAY);
6 }
7 }
A red arrow points to the file name 'Constants.java' in the editor tab. A red circle highlights the line 'NUMBER_OF_HOURS_IN_A_DAY = 25;' on line 4, indicating a reassignment of a final variable.

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

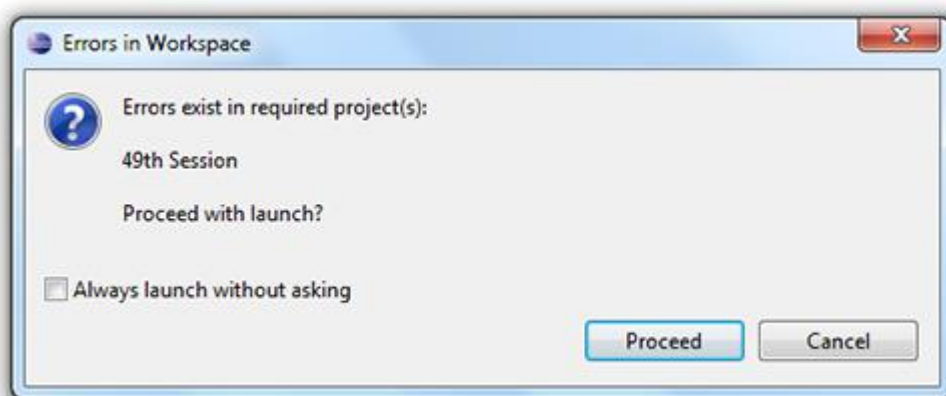
مدرس: بهزاد مرادی

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:

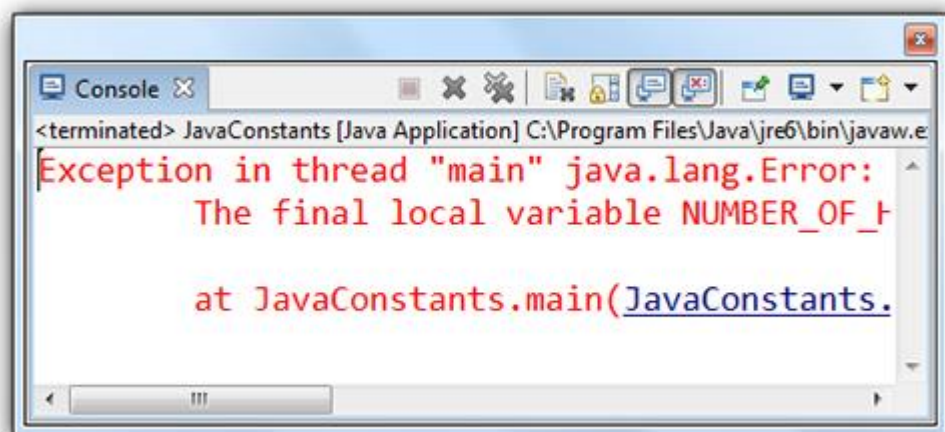


دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

در این پنجره اکتیپس 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 بی 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:

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

```
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:

۱. چرا در برنامه نویسی نیاز به Constant ها داریم؟
۲. نحوه ایجاد یک Constant در جاوا چگونه است؟
۳. چه فرقی مابین ایجاد یک Constant در سطح کلاس و یک Constant در سطح متد وجود دارد؟
۴. اگر مقدار یک Constant را تغییر دهیم چه اتفاقی روی خواهد داد؟