

به نام خدا

آموزش سی و چهارم

اهداف آموزشی این قسمت عبارتند از:

۱. معرفی Cast در زبان برنامه نویسی جاوا

پس از آشنایی با مبحث وراثت در زبان برنامه نویسی جاوا در آموزش سی و دوم و سی و سوم، حال نوبت به توضیح پیرامون Cast کردن در برنامه نویسی می رسد (در زبان انگلیسی واژه Cast دارای معانی متفاوتی است و این در حالی است که در زبان برنامه نویس جاوا این واژه تعریف خاصی دارد. از این رو از ترجمه لغوی این اصلاح خودداری می کنیم). گاهی اوقات در برنامه نویسی برای ما پیش می آید که نیاز داریم تا جنس داده های خود را تغییر دهیم که اصطلاحاً به فرایند تغییر جنس داده ای به داده ای دیگر در زبان برنامه نویسی جاوا Cast کردن گفته می شود. به طور مثال فرض کنیم که ما نیاز داریم تا یک متغیر از جنس int که برای اعداد صحیح مورد استفاده قرار می گیرد را به متغیر double که برای اعداد اعشاری مورد استفاده قرار می گیرد مبدل سازیم که برای این کار می بایست از عمل Cast کردن استفاده کنیم. لازم به ذکر است پس از آنکه ما مثلاً یک متغیر را به یک متغیر دیگر Cast کردیم در واقع ما متغیر قبلی خود را تبدیل به متغیر جدید نکرده ایم بلکه یک متغیر جدید از روی متغیر قبلی خود ساخته ایم که دارای همان جنسی است که مد نظر ما است. از میان کلیه متغیرها تنها متغیری که قابلیت Cast شدن ندارد متغیرهایی از جنس boolean هستند که در آموزش های پیشین با آن آشنا شده ایم (به علاقمندان به طراحی اپلیکیشن های اندروید توصیه می شود این مبحث را با دقت فرا گیرند چرا که از Cast به کرات در کدهای خود که برای سیستم عامل اندروید طراحی می کنند استفاده خواهند کرد).

در حقیقت داده اولیه که ما می خواهیم آن ها را به داده ای از جنس دیگری تبدیل کنیم در اصطلاح برنامه نویسی داده Source به معنی "منبع" نامیده می شود و داده جدید ایجاد شده داده Destination به معنی "مقصد" نامیده می شود. نکته ای که در اینجا حائز اهمیت است این است که اگر داده های Destination یا همان "مقصد" از داده Source بزرگ تر باشند فرایند تبدیل

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

به خوبی انجام خواهد شد. برای روشن شدن این مطلب یک مثال می زنیم. فرض کنیم که داده Source ما داده ای از جنس یک متغیر از نوع byte می باشد که قصد داریم تا آن را به داده ای از جنس متغیر دیگری از نوع int تبدیل کنیم (برای آشنایی بیشتر می توان به مبحث Primitive Type ها در آموزش پنجم مراجعه کرد). همانطور که می دانیم متغیر int نسبت به متغیری از جنس byte که هر دو برای ذخیره سازی اعداد صحیح به کار می روند به مراتب مقادیر بزرگ تری را در خود جای می دهد.

گاهی اوقات نیز عکس این قضیه اتفاق می افتد. به عبارت دیگر فضایی که داده Destination از حافظه اشغال می کند نسبت به فضای اشغال شده توسط داده Source کوچکتر است. به طور مثال فرض کنیم که می خواهیم متغیری از جنس int را به متغیری از جنس short که برای ذخیره سازی اعداد صحیح استفاده می شود Cast کنیم. از آن جاییکه فضای ذخیره سازی متغیری از جنس short از متغیری از جنس int به مراتب کمتر است، جاوا این تبدیل را برای ما انجام خواهد داد اما این در حالی است که بایستی شاهد نتایج کمی غیر قابل انتظار باشیم.

حال برای روش شدن مطالب تئوری فوق، پروژه ای جدید تحت عنوان 34th Session به معنی "جلسه سی و چهارم" ایجاد کرده سپس کلاسی به نام Casting در آن ایجاد می کنیم. حال کد اولیه ما به شکل زیر خواهد بود:

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

حال یک متغیر از جنس double تحت عنوان doubleVariable به مقدار اولیه 10.2 ایجاد می کنیم. کد تکمیل شده ما به شکل زیر خواهد بود:

```
public class Casting {  
    public static void main(String[] args) {  
        double doubleVariable = 10.2;  
  
    }  
}
```

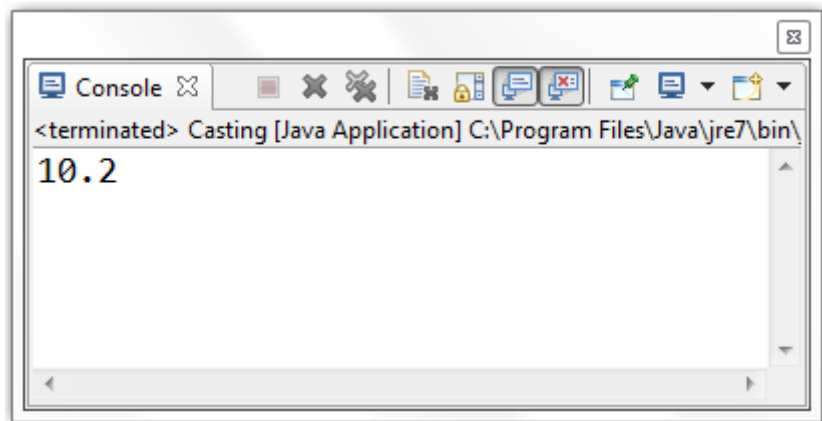
متغیر double ایجاد شده به منزله داده Source ما خواهد بود. حال قصد داریم تا این داده را به داده ای از جنس متغیر int که برای ذخیره سازی اعداد صحیح مورد استفاده قرار می گیرد Cast کنیم. برای این منظور کد فوق را به شکل زیر تکمیل می کنیم:

```
public class Casting {  
    public static void main(String[] args) {  
        double doubleVariable = 10.2;  
        int integerVariable = (int) doubleVariable;  
    }  
}
```

همانطور که در کد فوق مشاهده می شود ما یک متغیر از جنس int تحت عنوان integerVariable ایجاد کرده و مقدار آن را معادل با مقدار داده Source که در مرحله قبل ایجاد کردیم قرار می دهیم. در واقع در این مرحله ما قصد داریم داده Source که همان متغیر double است را به داده Destination که همان متغیر int است Cast کنیم برای همین منظور پس از علامت مساوی مقابل نام داده Destination یک علامت () قرار داده و نوع داده ای را که می خواهیم داده اولیه به آن Cast شود را می نویسیم و از آن جا که در این مثال می خواهیم داده ما به داده ای از جنس متغیر int تبدیل شود داخل پرانتز نوع متغیر مد نظر که int می باشد را می نویسیم. در اینجا کار ما به اتمام رسیده است. حال برای آن که خروجی برنامه خود را ببینیم نیاز به دستور System.out.println(); داریم. در واقع در مرحله اول نام متغیر اولیه خود یا همان متغیر Source را به صورت زیر می نویسیم:

```
public class Casting {  
    public static void main(String[] args) {  
        double doubleVariable = 10.2;  
        int integerVariable = (int) doubleVariable;  
        System.out.println(doubleVariable);  
    }  
}
```

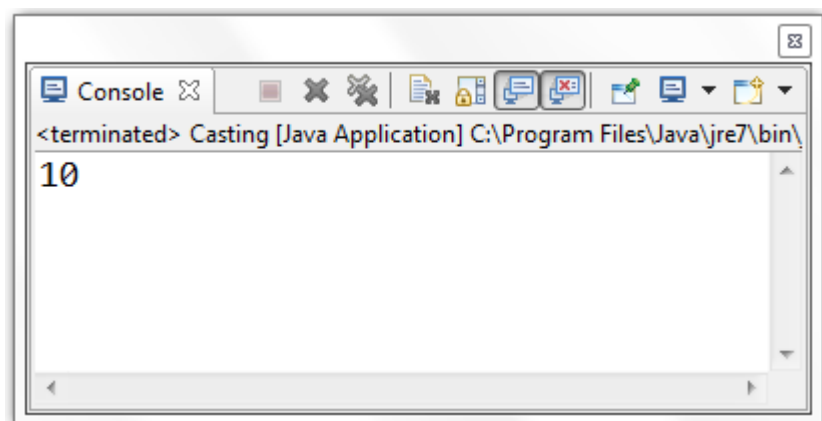
حال اگر برنامه را اجرا کنیم خروجی زیر قابل مشاهده خواهد بود:



در واقع از آنجا که داده اولیه ما از جنس متغیری از نوع `double` است، ما در خروجی برنامه یک عدد اعشاری خواهیم داشت. حال اگر داده `Destination` یا همان داده نهایی که پس از `Cast` شدن داده اولیه در قالب یک متغیر از جنس `int` قرار گرفته است را در دستور `System.out.println()` به صورت زیر قرار دهیم متوجه عملکرد `Cast` خواهیم شد:

```
public class Casting {
    public static void main(String[] args) {
        double doubleVariable = 10.2;
        int integerVariable = (int) doubleVariable;
        System.out.println(integerVariable);
    }
}
```

در واقع در کد فوق نام متغیر `Destination` که `Cast` شده می باشد را داخل پرانتز می نویسیم. خروجی برنامه به شکل زیر خواهد بود:



در تصویر فوق می بینیم که جاوا به صورت خودکار پس از `Cast` کردن داده ای از جنس عدد اعشاری به داده ای از جنس عدد صحیح، عدد را رند کرده است.

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

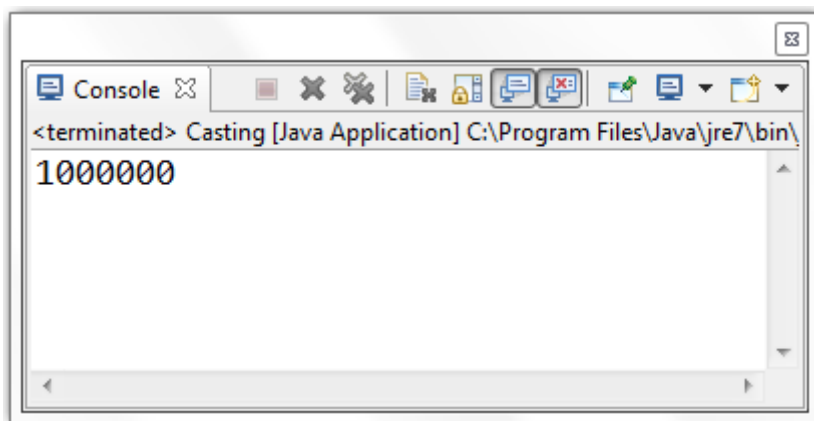
حال بینیم عکس این قضیه چه طور ممکن است اتفاق افتد. در این مثال قصد داریم تا Source یی بزرگ تر را به Destination یی کوچک تر از خود Cast کنیم. برای این منظور کد خود را به شکل زیر بازنویسی می کنیم:

```
public class Casting {  
    public static void main(String[] args) {  
        int integerValue = 1000000;  
    }  
}
```

در واقع در مثال فوق داده اولیه ما از جنس متغیری از نوع int است که مقدار اولیه آن معادل با 1000000 است. متغیری از جنس int همواره نسبت به متغیری از جنس short دارای ظرفیت بیشتری برای ذخیره سازی اعداد صحیح می باشد. حال می خواهیم متغیر int مثال فوق را به متغیر short که نام آن shortVariable است Cast کنیم. برای این منظور کد فوق را به شکل زیر تکمیل می کنیم:

```
public class Casting {  
    public static void main(String[] args) {  
        int integerValue = 1000000;  
        short shortVariable = (short) integerValue;  
    }  
}
```

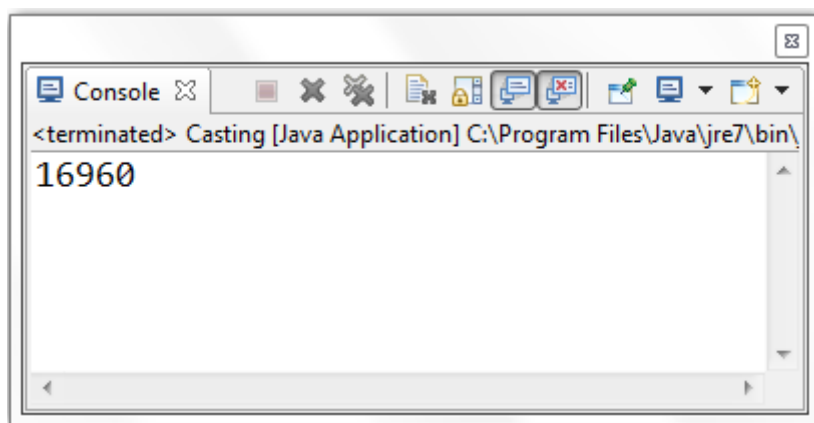
حال در مرحله اول برای به نمایش در آوردن خروجی برنامه در دستور System.out.println(); نام متغیر خود از جنس int را می نویسیم و سپس برنامه را اجرا می کنیم:



می بینیم که همان Value یی که برای متغیر int خود در نظر گرفته بودیم در پنجره Console به نمایش در آمده است. حال نام متغیر short که Cast شده است را به صورت زیر فرا می خواهیم:

```
public class Casting {  
    public static void main(String[] args) {  
        int integerValue = 1000000;  
        short shortVariable = (short) integerValue;  
        System.out.println(shortVariable);  
    }  
}
```

پس از اجرای برنامه خروجی زیر مشاهده خواهد شد:



همانطور که ملاحظه می شود از آنجا که ظرفیت متغیری از جنس short نسبت به متغیری از جنس int به مراتب کمتر است، خروجی برنامه نیز عددی به مراتب کوچکتر از مقدار اولیه متغیر int پیش از Cast شدن خواهد بود. در آموزش قسمت سی و پنجم با نحوه Cast کردن یک Object به یک Variable و بالعکس آشنا خواهیم شد.

پس از مطالعه این آموزش انتظار می رود بتوانیم به سؤالات زیر پاسخ بدهیم:

۱. چرا در جاوا از عمل Cast استفاده می کنیم؟
۲. اگر یک متغیر بزرگ را به متغیر کوچک تری Cast کنیم چه اتفاقی روی خواهد داد؟
۳. کدام متغیر است که به هیچ وجه نمی توان آنرا Cast کرد؟

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی