Type conversion & casting in Java

By
M. BABY ANUSHA,
ASST.PROF IN CSE DEPT.,
RGUKT,NUZVID.

Type conversion :

- When you assign value of one data type to another, the two types might not be compatible with each other.
- If the data types are compatible, then Java will perform the conversion automatically known as Automatic Type Conversion and if not then they need to be casted or converted explicitly.
- For example, assigning an int value to a long variable.

Continued…

- In Java, there are two types of casting:

1. **Widening Casting** (automatically) - converting a smaller type to a larger type size
   byte -> short -> char -> int -> long -> float -> double

2. **Narrowing Casting** (manually) – converting a larger type to a smaller size type
   double -> float -> long -> int -> char -> short ->

Widening or Automatic Type Conversion

Widening conversion takes place when two data types are automatically converted. This happens when:

- The two data types are compatible.
- When we assign value of a smaller data type to a bigger data type.

Continued…

- For Example, in java the numeric data types are compatible with each other but no automatic conversion is supported from numeric type to char or boolean.
- Also, char and boolean are not compatible with each other.

Byte –> Short –> Int –> Long – > Float –> Double

Widening or Automatic Conversion

Widening or Automatic Type Conversion- Example 1

```
class Test
{
    public static void main(String[] args)
    {
        int i = 100;

        // automatic type conversion
        long l = i;

        // automatic type conversion
        float f = l;
        System.out.println("Int value "+i);
        System.out.println("Long value "+l);
        System.out.println("Float value "+f);
    }
}
```

Output :
Int value 100
Long value 100
Float value 100.0

Widening or Automatic Type Conversion- Example 2

```
public class Main
{
   public static void main(String[] args)
   {
       int myInt = 9;
       double myDouble = myInt;
      // Automatic casting: int to double
  System.out.println(myInt); // Outputs 9          System.out.
println(myDouble);  // Outputs 9.0
 }
}
```

Output:

9

9.0

Narrowing or Explicit Conversion :

- If we want to assign a value of larger data type to a smaller data type we perform explicit type casting or narrowing.
- This is useful for incompatible data types where automatic conversion cannot be done.
- Here, target-type specifies the desired type to convert the specified value to.

Continued…

- Narrowing casting must be done manually by placing the type in parentheses in front of the value.
- For Example, char and number are not compatible with each other. Let's see when we try to convert one into other.

Double –> Float –> Long –> Int –> Short –> Byte

Narrowing or Explicit Conversion

Example Without Explicit Conversion:

```
// Java program to illustrate incompatible data
// type for explicit type conversion
public class Test
{
  public static void main(String[] argv)
  {
    char ch = 'c';
    int num = 88;
    ch = num;
  }
}
```

Output:

Error:

7: error: incompatible types: possible lossy conversion from int to char ch = num; ^ 1 error

Narrowing or Explicit Conversion : Example 1

```
// Java program to illustrate explicit type conversion
class Test
{
    public static void main(String[] args)
    {
        double d = 100.04;
        //explicit type casting
        long l = (long)d;
        //explicit type casting
        int i = (int)l;
        System.out.println("Double value "+d);
        //fractional part lost
        System.out.println("Long value "+l);
        //fractional part lost
        System.out.println("Int value "+i);
    }
}
```

Output:
Double value 100.04
Long value 100
Int value 100

Narrowing or Explicit Conversion : Example 2

```
//Java program to illustrate Conversion of int and double to byte
class Test
{
    public static void main(String args[])
    {
        byte b;
        int i = 257;
        double d = 323.142;
        System.out.println("Conversion of int to byte.");
        //i%256
        b = (byte) i;
        System.out.println("i = " + i + " b = " + b);
        System.out.println("\nConversion of double to byte.");
        //d%256
        b = (byte) d;
        System.out.println("d = " + d + " b= " + b);
    }
}
```

Output:

```
Conversion of int to byte.
i = 257
b = 1
Conversion of double to byte.
d = 323.142
b = 67
```

Narrowing or Explicit Conversion : Example 2

```
public class Main
 {
public static void main(String[] args)
{
double myDouble = 9.78d;
 int myInt = (int) myDouble; // Manual casting: double to int
System.out.println(myDouble); // Outputs 9.78 System.out.println(myInt); // Outputs 9
}
}
```

Output:

9.78
9


Thank you!
shutterstock.com · 567687052