

Java



Wrapper 클래스

Wrapper 클래스에 대하여
Wrapper 클래스의 사용 방법
Wrapper 클래스의 정적 메소드의 상수
자동 Boxing과 자동 Unboxing



Wrapper 클래스

01. Wrapper 클래스에 대하여

Wrapper 클래스란?

•• 프리미티브 타입을 객체로 표현하는 데 사용되는 다음 클래스들의 통칭

이름에 주의

클래스 이름	해당 프리미티브 타입
Byte	byte
Short	short
Integer	int
Long	long
Character	char
Float	float
Double	double
Boolean	boolean

- 위 클래스들은 모두 `java.lang` 패키지에 속함

Wrapper 클래스

01. Wrapper 클래스에 대하여

Wrapper 클래스란?

- 프리미티브 타입을 데이터를 감싸는 역할을 하는 Wrapper 클래스

<p>Byte 객체</p> <p>byte 타입 데이터</p> <p>3</p>	<p>Short 객체</p> <p>short 타입 데이터</p> <p>123</p>	<p>Integer 객체</p> <p>int 타입 데이터</p> <p>12000</p>	<p>Long 객체</p> <p>long 타입 데이터</p> <p>10000000000L</p>
<p>Character 객체</p> <p>char 타입 데이터</p> <p>'꽃'</p>	<p>Float 객체</p> <p>float 타입 데이터</p> <p>11.5f</p>	<p>Double 객체</p> <p>double 타입 데이터</p> <p>1.000005</p>	<p>Boolean 객체</p> <p>boolean 타입 데이터</p> <p>true</p>

Wrapper 클래스

02. Wrapper 클래스의 기본적인 사용 방법

Wrapper 클래스의 기본적인 사용 방법

- 래퍼 객체(wrapper object)를 만드는 방법

```
Byte      obj1 = new Byte((byte) 1);  
Short     obj2 = new Short((short) 123);  
Integer   obj3 = new Integer(12345);  
Long      obj4 = new Long(1234567890L);  
Float     obj5 = new Float(1.5f);  
Double    obj6 = new Double(1.00005);  
Character obj7 = new Character('꽃');  
Boolean   obj8 = new Boolean(true);
```

Wrapper 클래스의 생성자

Wrapper 클래스

02. Wrapper 클래스의 기본적인 사용 방법

Wrapper 클래스의 기본적인 사용 방법

- 래퍼 객체 안에 있는 프리미티브 값을 가져오는 방법

```
byte    num1 = obj1.byteValue();  
short   num2 = obj2.shortValue();  
int      num3 = obj3.intValue();  
long     num4 = obj4.longValue();  
float    num5 = obj5.floatValue();  
double   num6 = obj6.doubleValue();  
char     ch  = obj7.charValue();  
boolean  truth = obj8.booleanValue();
```

Wrapper 객체 안에 있는
프리미티브 타입의 값을
가져오는 메소드

Wrapper 클래스

02. Wrapper 클래스의 기본적인 사용 방법

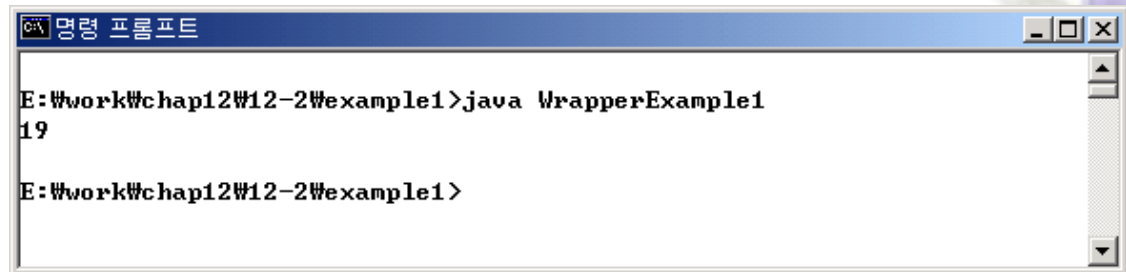
Wrapper 클래스의 기본적인 사용 방법

- [예제 12-1] Wrapper 클래스의 사용 예

```
1 class WrapperExample1 {  
2     public static void main(String args[]) {  
3         Integer obj1 = new Integer(12);  
4         Integer obj2 = new Integer(7);  
5         int sum = obj1.intValue() + obj2.intValue();  
6         System.out.println(sum);  
7     }  
8 }
```

} Integer 객체를 생성합니다.

Integer 객체 안에 있는 int 값을 가져옵니다



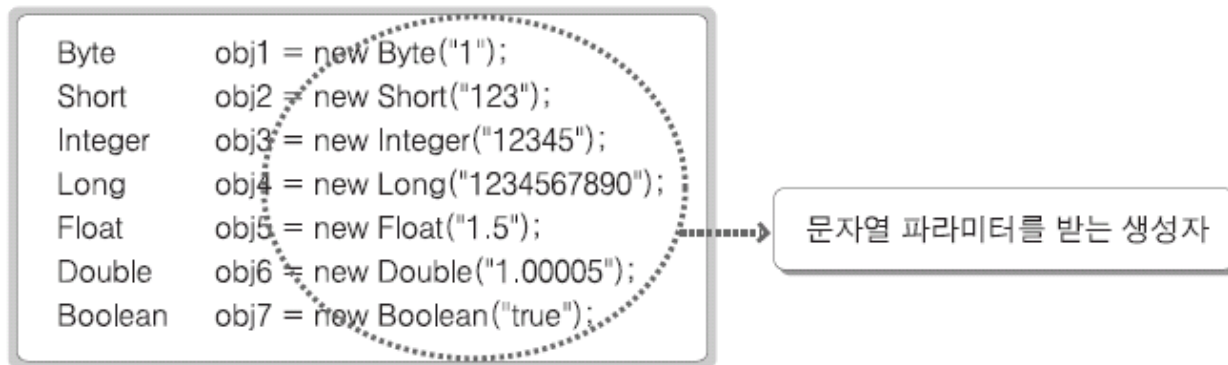
```
명령 프롬프트  
E:\work\chap12\12-2\example1>java WrapperExample1  
19  
E:\work\chap12\12-2\example1>
```

Wrapper 클래스

02. Wrapper 클래스의 기본적인 사용 방법

문자열 파라미터를 받는 생성자

- 문자열 파라미터를 프리미티브 값으로 바꾸어서 래퍼 객체 안에 저장하는 생성자
- 사용 방법



- 래퍼 클래스의 `Value` 메소드와 함께 사용하면 문자열을 프리미티브 값으로 바꿀 수 있음

Wrapper 클래스

02. Wrapper 클래스의 기본적인 사용 방법

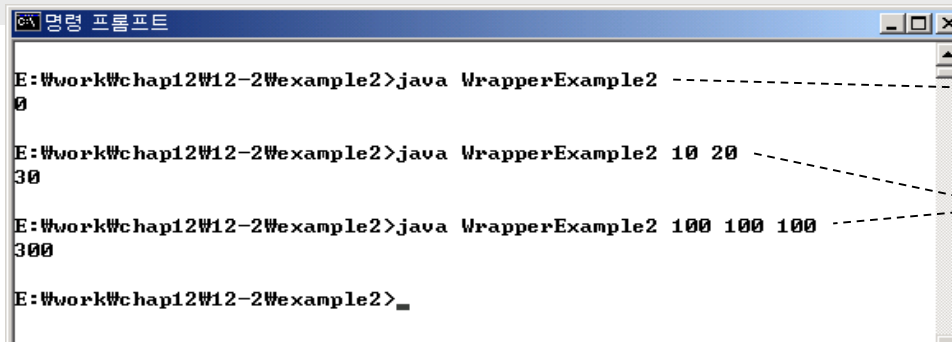
문자열 파라미터를 받는 생성자

- [예제 12-2] Wrapper 클래스의 문자열 파라미터를 받는 생성자 사용 예

```
1  class WrapperExample2 {  
2      public static void main(String args[]) {  
3          int total = 0;  
4          for (int cnt = 0; cnt < args.length; cnt++) {  
5              Integer obj = new Integer(args[cnt]);  
6              total += obj.intValue();  
7          }  
8          System.out.println(total);  
9      }  
10 }
```

명령형 파라미터로 받은 문자열을 가지고 Integer 객체를 생성합니다.

Integer 객체 안에 있는 int 값을 가져옵니다.



```
명령 프롬프트  
E:\work\chap12\12-2\example2>java WrapperExample2  
0  
E:\work\chap12\12-2\example2>java WrapperExample2 10 20  
30  
E:\work\chap12\12-2\example2>java WrapperExample2 100 100 100  
300  
E:\work\chap12\12-2\example2>
```

파라미터로 아무 값도 넘겨주지 않으면 0을 출력합니다.

파라미터로 정수를 넘겨주면 합계를 출력합니다.

Wrapper 클래스

03. Wrapper 클래스의 정적 메소드와 상수

정적 메소드

- 프리미티브 타입의 비트 패턴을 문자열로 리턴하는 메소드

```
String str1 = Integer.toBinaryString(9);           // "1001"을 리턴  
String str2 = Long.toBinaryString(1000000000L);  
// "101111101011110000100000000"을 리턴
```

- 부동소수점수와 똑같은 비트 패턴을 갖는 정수를 리턴하는 메소드

```
int  num1 = Float.floatToRawIntBits(1.5f);  
// 1.5f와 똑같은 비트 패턴의 int 값을 리턴  
long num2 = Double.doubleToRawLongBits(1.0005);  
// 1.0005와 똑같은 비트 패턴의 long 값을 리턴
```

Wrapper 클래스

03. Wrapper 클래스의 정적 메소드와 상수

정적 메소드

- 문자열을 프리미티브 타입으로 바꾸는 메소드

```
byte    num1 = Byte.parseByte("1");  
short   num2 = Short.parseShort("123");  
int      num3 = Integer.parseInt("12345");  
long     num4 = Long.parseLong("1234567890");  
float    num5 = Float.parseFloat("1.5");  
double   num6 = Double.parseDouble("1.00005");  
boolean  truth = Boolean.parseBoolean("true");
```

문자열을 프리미티브 타입의
값으로 바꾸는 메소드

Wrapper 클래스

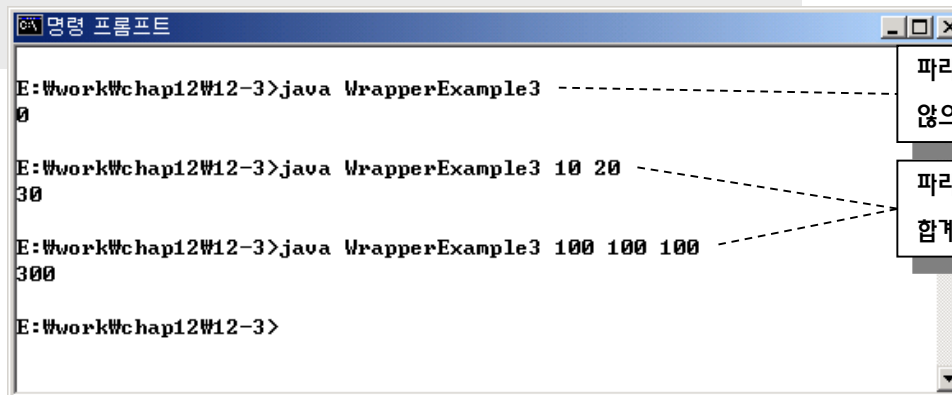
03. Wrapper 클래스의 정적 메소드와 상수

정적 메소드

- [예제 12-3] parse-메소드의 사용 예

```
1 class WrapperExample3 {  
2     public static void main(String args[]) {  
3         int total = 0;  
4         for (int cnt = 0; cnt < args.length; cnt++)  
5             total = Integer.parseInt(args[cnt]);  
6         System.out.println(total);  
7     }  
8 }
```

명령행 파라미터로 받은 문자열을
int 타입의 값으로 바꾸어서 합산합니다.



```
E:\work\chap12\12-3>java WrapperExample3  
0  
  
E:\work\chap12\12-3>java WrapperExample3 10 20  
30  
  
E:\work\chap12\12-3>java WrapperExample3 100 100 100  
300  
  
E:\work\chap12\12-3>
```

파라미터로 아무 값도 넘겨주지
않으면 0을 출력합니다.

파라미터로 정수를 넘겨주면
합계를 출력합니다.

Wrapper 클래스

03. Wrapper 클래스의 정적 메소드와 상수

정적 메소드

- Wrapper 클래스의 생성자를 대신하는 메소드

```
Byte    obj1 = Byte.valueOf((byte) 1);  
Short   obj2 = Short.valueOf((short) 123);  
Integer obj3 = Integer.valueOf(12345);  
Long    obj4 = Long.valueOf(1234567890L);  
Float   obj5 = Float.valueOf(1.5f);  
Double  obj6 = Double.valueOf(1.00005);  
Character obj7 = Character.valueOf('꽃');  
Boolean obj8 = Boolean.valueOf(true);
```

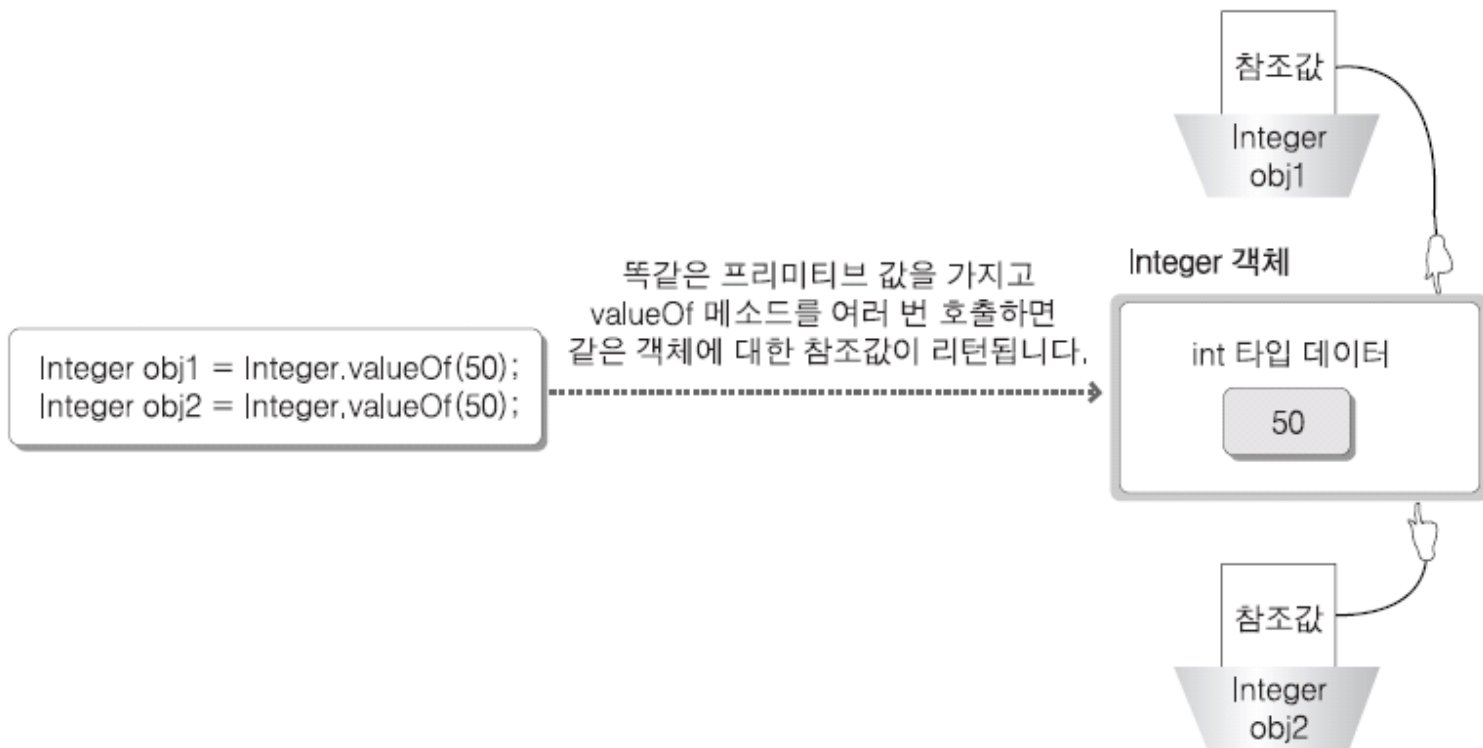
Wrapper 클래스의 생성자를
대신하는 메소드

Wrapper 클래스

03. Wrapper 클래스의 정적 메소드와 상수

정적 메소드

- valueOf 메소드의 작동 방식



Wrapper 클래스

03. Wrapper 클래스의 정적 메소드와 상수

상수

- Wrapper 클래스에 선언되어 있는 상수들

Double (Java 2 Platform SE 5.0) - Microsoft Internet Explorer

Overview Package Class Use Tree Deprecated Index Help

java.lang Class Double

All Implemented Interfaces: Serializable, Comparable<Double>

public final class Double
extends Number
implements Comparable<Double>

The Double class wraps a value of the primitive type double in an object. An object of type Double is double.

In addition, this class provides several methods for converting a double to a String and a String constants and methods useful when dealing with a double.

Since: JDK1.0
See Also: Serialized Form

Field Summary

static double	MAX_VALUE	A constant holding the largest positive finite value of type double, (2-2 ⁵²).
static double	MIN_VALUE	A constant holding the smallest positive nonzero value of type double, 2 ⁻²⁵² .
static double	NaN	A constant holding a Not-a-Number (NaN) value of type double.
static double	NEGATIVE_INFINITY	A constant holding the negative infinity of type double.
static double	POSITIVE_INFINITY	A constant holding the positive infinity of type double.
static int	SIZE	The number of bits used to represent a double value.

Integer (Java 2 Platform SE 5.0) - Microsoft Internet Explorer

Overview Package Class Use Tree Deprecated Index Help

java.lang Class Integer

All Implemented Interfaces: Serializable, Comparable<Integer>

public final class Integer
extends Number
implements Comparable<Integer>

The Integer class wraps a value of the primitive type int in an object. An object of type Integer contains a single field whose type is int.

In addition, this class provides several methods for converting an int to a String and a String to an int, as well as other constants and methods useful when dealing with an int.

Implementation note: The implementations of the "bit twiddling" methods (such as `highestOneBit` and `numberOfTrailingZeros`) are based on material from Henry S. Warren, Jr.'s *Hacker's Delight*. (Addison Wesley, 2002).

Since: JDK1.0
See Also: Serialized Form

Field Summary

static int	MAX_VALUE	A constant holding the maximum value an int can have, 2 ³¹ -1.
static int	MIN_VALUE	A constant holding the minimum value an int can have, -2 ³¹ .
static int	SIZE	The number of bits used to represent an int value in two's complement binary form.
static Class	TYPE	The Class instance representing the primitive type int.

부동소수점 타입이 표현할 수 있는 특별한 의미의 데이터를 선언한 상수

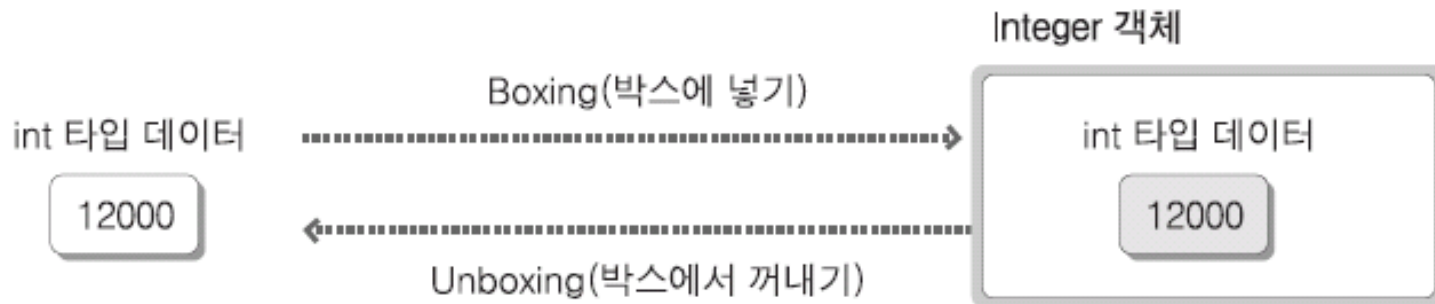
프리미티브 타입이 표현할 수 있는 최대값, 최소값을 선언한 상수

Wrapper 클래스

04. 자동 Boxing과 자동 Unboxing

Boxing과 Unboxing

- Boxing과 Unboxing의 개념도



[예]

```
Integer obj = new Integer(12000);  
int num = obj.intValue();
```

```
// Boxing  
// Unboxing
```


● 04. 자동 Boxing과 자동 Unboxing

● 자동 Boxing과 자동 Unboxing

● 자동 Boxing

- 래퍼 객체를 써야할 자리에 프리미티브 값을 썼을 때 일어나는 래퍼 객체로의 자동 변환

● 자동 Unboxing

- 프리미티브 값을 써야할 자리에 래퍼 객체를 썼을 때 일어나는 프리미티브 값으로의 자동 변환

Wrapper 클래스

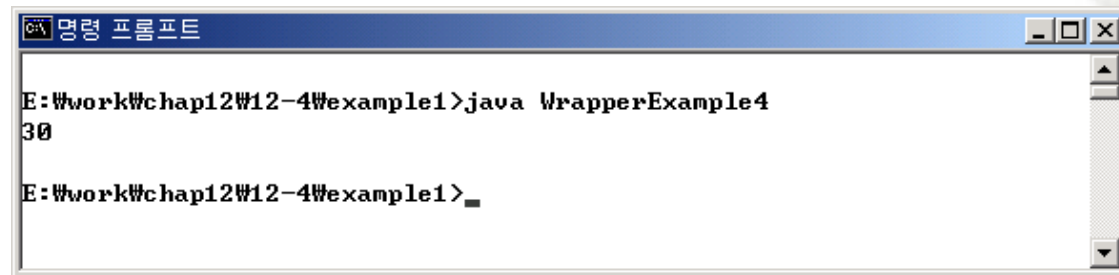
04. 자동 Boxing과 자동 Unboxing

자동 Unboxing

- [예제 12-4] 자동 Unboxing이 일어나는 경우

```
1 class WrapperExample4 {  
2     public static void main(String args[]) {  
3         Integer obj = new Integer("10");  
4         int sum = obj + 20; -----  
5         System.out.println(sum);  
6     }  
7 }
```

Integer 객체와 int 타입의 값을 더하는 명령문



```
명령 프롬프트  
E:\work\chap12\12-4\example1>java WrapperExample4  
30  
E:\work\chap12\12-4\example1>
```

Wrapper 클래스

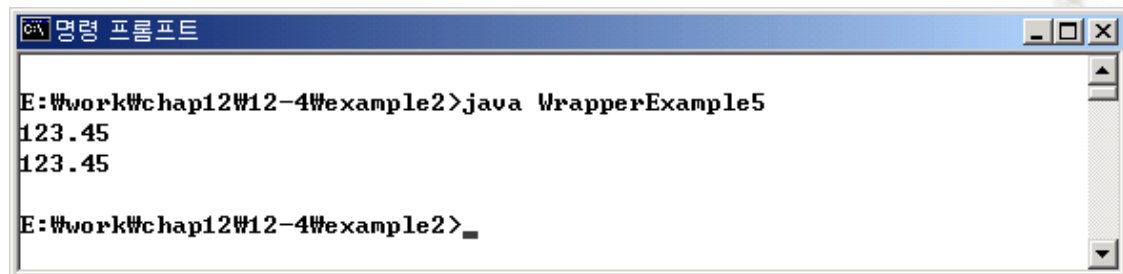
04. 자동 Boxing과 자동 Unboxing

자동 Boxing

- [예제 12-5] 자동 Boxing이 일어나는 경우

```
1    class WrapperExample5 {  
2        public static void main(String args[]) {  
3            printDouble(new Double(123.45));  
4            printDouble(123.45);  
5        }  
6        static void printDouble(Double obj) {  
7            System.out.println(obj);  
8        }  
9    }
```

double 타입의 값을 가지고
Double 타입 파라미터를 받는 메소드를 호출합니다.



```
명령 프롬프트  
E:\work\chap12\12-4\example2>java WrapperExample5  
123.45  
123.45  
E:\work\chap12\12-4\example2>
```

어제

많은

일을

이루고

싶습니다~