



기본API 클래스

안 화 수

기본 API클래스

- ❖ 날짜, 시간 클래스
Date, Timestamp, Calendar
- ❖ 문자열 클래스
String, StringBuffer, StringTokenizer
- ❖ Wrapper 클래스

날짜, 시간 클래스

❖ 날짜, 시간 클래스

- Date : 날짜와 시간 정보를 처리하는 클래스
- Timestamp : 날짜와 정밀한 시간을 처리하는 클래스
- Calendar : 운영체제의 날짜와 시간을 처리하는 클래스

날짜, 시간 클래스

❖ 날짜, 시간 클래스

```
Date d = new Date();
```

```
Timestamp ts = new Timestamp(System.currentTimeMillis());
```

```
Calendar c = Calendar.getInstance();
```

날짜, 시간 클래스

❖ 날짜 형식 클래스 : SimpleDateFormat

```
SimpleDateFormat sdf = new SimpleDateFormat("yyyy 년 MM 월 dd 일");  
String strDate = sdf.format(new Date());
```

패턴 문자	의미	패턴 문자	의미
y	년	H	시(0~23)
M	월	h	시(1~12)
d	일	K	시(0~11)
D	월 구분이 없는 일(1~365)	k	시(1~24)
E	요일	m	분
a	오전/오후	s	초
w	년의 몇번째 주	S	밀리세컨드(1/1000 초)
W	월의 몇번째 주		

날짜, 시간 클래스

❖ 날짜, 시간 클래스 : Date, Timestamp

```
public class DateEx {  
  
    public static void main(String[] args) {  
  
        // 날짜 시간 관련 클래스 : Date, Timestamp  
  
        // 1. Date  
        // 년, 월, 일, 시, 분, 초, 요일로 날짜를 출력  
        SimpleDateFormat sd =  
            new SimpleDateFormat("yyyy년 MM월 dd일 a hh:mm:ss EEE요일");  
        // hh : 12시간제 시간  
        // HH : 24시간제 시간  
  
        Date d = new Date();  
        System.out.println(d);  
        System.out.println(sd.format(d));  
  
        // 2. Timestamp  
        // 년, 월, 일, 시, 분, 초 로 날짜를 출력  
        SimpleDateFormat sf =  
            new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");  
  
        Timestamp ts = new Timestamp(System.currentTimeMillis());  
        System.out.println(ts);  
        System.out.println(sf.format(ts));  
    }  
}
```

날짜, 시간 클래스

❖ 날짜, 시간 클래스 : Calendar

```
public class CalendarEx {  
  
    public static void main(String[] args) {  
        // 3. Calendar 클래스  
        //Calendar c1 = new Calendar();           // 오류 발생  
        Calendar c2 = new GregorianCalendar();     // 업캐스팅  
  
        Calendar c = Calendar.getInstance();  
        System.out.println(c);                    // 1673401352299  
        System.out.println(Calendar.YEAR);        // 1  
  
        int y = c.get(Calendar.YEAR);             // 연 : 2023  
        int m = c.get(Calendar.MONTH) + 1;        // 월 : 0 ~ 11  
        int d = c.get(Calendar.DATE);             // 일  
        System.out.println(y+"-"+m+"-"+d);  
        int h1 = c.get(Calendar.HOUR);            // 12시간  
        int h2 = c.get(Calendar.HOUR_OF_DAY);     // 24시간  
        int ap = c.get(Calendar.AM_PM);           // 0:오전, 1:오후  
        if(ap == 0) {  
            System.out.print("오전 ");  
        }else {  
            System.out.print("오후 ");  
        }  
        int mm = c.get(Calendar.MINUTE);          // 분  
        int s = c.get(Calendar.SECOND);           // 초  
        System.out.println(h1+": "+mm+": "+s);    // 12시간  
        System.out.println(h2+": "+mm+": "+s);    // 24시간  
    }  
}
```

문자열 클래스

❖ 문자열 클래스

- String
- StringBuffer
- StringTokenizer

문자열 클래스

❖ String 클래스

리턴타입	메소드명(매개변수)	설명
char	charAt(int index)	특정 위치의 문자 리턴
boolean	equals (Object anObject)	두 문자열을 비교
byte[]	getBytes()	byte[]로 리턴
byte[]	getBytes(Charset charset)	주어진 문자셋으로 인코딩한 byte[]로 리턴
int	indexOf (String str)	문자열내에서 주어진 문자열의 위치를 리턴
int	length ()	총 문자의 수를 리턴
String	replace(CharSequence target, CharSequence replacement)	target 부분을 replacement 로 대치한 새로운 문자열을 리턴
String	substring (int beginIndex)	beginIndex 위치에서 끝까지 잘라낸 새로운 문자열을 리턴
String	substring (int beginIndex, int endIndex)	beginIndex 위치에서 endIndex 전까지 잘라낸 새로운 문자열을 리턴
String	toLowerCase ()	알파벳 소문자로 변환한 새로운 문자열을 리턴
String	toUpperCase ()	알파벳 대문자로 변환한 새로운 문자열을 리턴
String	trim()	앞뒤 공백을 제거한 새로운 문자열을 리턴
String	valueOf(int i) valueOf(double d)	기본 타입값을 문자열로 리턴

문자열 클래스

❖ String 클래스

```
class StringTest00 {  
    public static void main(String[] args) {  
  
        String str1 = new String("Java Programming");  
        String str2 = new String("Java Programming");  
        if (str1 == str2) {           // 주소비교  
            System.out.println("같다");  
        } else {  
            System.out.println("다르다");  
        }  
  
        String str3 = "Java Programming";  
        String str4 = "Java Programming";  
        if (str3 == str4) {           // 주소비교  
            System.out.println("같다");  
        } else {  
            System.out.println("다르다");  
        }  
  
        if (str1 == str3) {           // 주소비교  
            System.out.println("같다");  
        } else {  
            System.out.println("다르다");  
        }  
  
        if (str1.equals(str3)) {      // 값비교  
            System.out.println("같은 값");  
        } else {  
            System.out.println("다른 값");  
        }  
    }  
}
```

문자열 클래스

❖ String 클래스

```
public class ConnectString {  
  
    public static void main(String[] args) {  
        // String 객체 생성  
        String gemini = "gemini";  
        String johnharu = "johnharu";  
  
        // 두 String 객체를 "+" 연산 수행  
        String tempString1 = gemini + johnharu;  
        System.out.println(tempString1);  
        System.out.println("gemini" + "johnharu");  
  
        // String + 정수형  
        String tempString2 = tempString1 + 100;  
        System.out.println(tempString2);  
    }  
}
```

문자열 클래스

❖ String 클래스 : toUpperCase(), toLowerCase()

```
class StringTest01 {  
  
    // 1. String객체를 생성한 후에 메소드에 의해서 값의 변화가 일어나면  
    //     변경된 값을 힙메모리 영역에 다시 저장 한다.  
    // 2. Garbage Collection 기능 (쓰레기 수집 기능)  
    //     재사용할 수 없는 힙메모리 영역의 데이터를 모아서 지워주는 기능  
  
    public static void main(String[] args) {  
  
        String str1 = "Java Programming";  
  
        str1.toUpperCase();           // 메서드 호출 후에도  
        System.out.println(str1);     // str1의 내용은 수정되지 않는다.  
        System.out.println(str1.toUpperCase());  
  
        String str2 = str1.toUpperCase(); // 메소드의 처리 결과를 str2에 저장  
        System.out.println(str2);  
    }  
}
```

문자열 클래스

❖ String 클래스 : equals(), trim()

```
public class TrimTest {  
    // trim() : 문자열 좌,우의 공백을 없애주는 역할  
    // boolean equals() : 문자열 값을 비교하는 역할  
  
    public static void main(String[] args) {  
        String str1 = new String("gemini ");  
        String str2 = new String("  gemini ");  
  
        System.out.println(str1.equals(str2));  
        System.out.println(str1.trim().equals(str2.trim()));  
    }  
}
```

문자열 클래스

❖ String 클래스 : length(), charAt()

```
public class FindBlankIndex {  
    public static void main(String[] args) {  
  
        // length() : 문자열의 길이를 구해주는 역할  
  
        String message = "Java program creates many objects.";  
  
        // message의 길이를 구함.  
        int len = message.length();  
  
        System.out.println(len); // len = 34;  
  
        // message 중에서 ' '을 찾음  
        for (int i = 0; i < len; i++) {  
            char c = message.charAt(i);  
            if (c == ' ') {  
                System.out.println("index = " + i);  
            }  
        }  
    }  
}
```

문자열 클래스

❖ String 클래스 : indexOf()

```
public class IndexOfTest {  
    public static void main(String[] args) {  
        // indexOf() : 특정 문자의 인덱스 번호를 구해주는 메소드  
  
        String message = "Java program creates many objects.";   
  
        // 가장 먼저 나오는 'a'의 인덱스 번호를 구해준다.  
        int index1 = message.indexOf('a'); // 1  
        // 10진수 아스키코드값 (97)에 해당하는 문자('a')의 인덱스 번호를 구해준다.  
        int index2 = message.indexOf(97); // 1  
  
        System.out.println(index1);  
        System.out.println(index2);  
  
        // index번호 13번째 이후에서 a의 인덱스 번호를 구해준다.  
        int index3 = message.indexOf('a', 13); // 16  
        System.out.println(index3);  
  
        int index4 = message.indexOf("av"); // 1  
        System.out.println(index4);  
  
        // index번호 12번 이후에서 'man'의 인덱스 번호를 구해준다.  
        int index5 = message.indexOf("man", 12); // 21  
        System.out.println(index5);  
  
        // 찾는 문자가 없을 경우에는 -1을 리턴한다.  
        int index6 = message.indexOf("java"); // -1  
        System.out.println(index6);  
    }  
}
```

문자열 클래스

❖ String 클래스 : substring()

```
public class SubStringTest {  
    public static void main( String[] args ) {  
  
        // substring() : 전체 문자열에서 특정 범위의 문자를 추출하는 역할.  
  
        // substring(int beginIndex) : beginindex번호부터 끝까지 문자를 추출.  
        // substring(int beginIndex, int endIndex)  
        // : beginIndex번호부터 endIndex-1번 까지 문자를 추출.  
  
        String message = "Java program creates many objects.";  
  
        // 인덱스번호 13번부터 끝까지 문자를 추출  
        String str1 = message.substring( 13 );  
        System.out.println( str1 );  
  
        // 인덱스번호 13번부터 15번까지 문자를 추출  
        String str2 = message.substring( 13, 16 );  
        System.out.println( str2 );  
    }  
}
```


문자열 클래스

❖ String 클래스 : substring(), equals()

```
public class SubstringEx {  
  
    public static void main(String[] args) {  
  
        // 아래와 같은 주민번호가 있을때, 남자인지 여자인지를 판별하는 프로그램 작성하세요?  
  
        String jumin = "950101-2234567";  
  
        String gender = jumin.substring(7, 8);  
        System.out.println(gender);  
  
        if(gender.equals("1") || gender.equals("3")) {  
            System.out.println("남자");  
        }else if(gender.equals("2") || gender.equals("4")){  
            System.out.println("여자");  
        }else {  
            System.out.println("잘못된 정보 입니다.");  
        }  
    }  
}
```

문자열 클래스

❖ String 클래스 : replace()

```
public class ReplaceEx {  
  
    public static void main(String[] args) {  
  
        // replace("자바", "JAVA") : 자바를 JAVA로 치환해주는 역할  
  
        String oldStr = "자바는 객체지향 언어 입니다.";   
  
        String newStr = oldStr.replace("자바", "JAVA");  
        System.out.println(oldStr);  
        System.out.println(newStr);  
  
    }  
}
```

문자열 클래스

❖ String 클래스 : split()

```
public class SplitEx {  
  
    public static void main(String[] args) {  
  
        // String[] split(String regex)  
  
        String jumin = "950101-1234567";  
        String[] j = jumin.split("-");  
        System.out.println("주민번호 앞자리:" + j[0]);  
        System.out.println("주민번호 뒷자리:" + j[1]);  
        for(int i=0; i<j.length; i++)  
            System.out.println(j[i]);  
  
        for(String s : j)                                // 향상된 for문  
            System.out.println(s);  
  
        String tel = "010-1234-5678";  
        String[] t = tel.split("-");  
        System.out.println("전화번호 앞자리:" + t[0]);    // 010  
        System.out.println("전화번호 중간자리:" + t[1]); // 1234  
        System.out.println("전화번호 끝자리:" + t[2]);    // 5678  
  
        String email = "totoro@naver.com";  
        String[] e = email.split("@");  
        System.out.println("아이디:" + e[0]);             // totoro  
        System.out.println("도메인:" + e[1]);             // naver.com  
    }  
}
```

문자열 클래스

❖ String 클래스 : split()

```
public class SplitEx1 {  
    public static void main(String[] args) {  
  
        // String[] split(String regex)  
        String text = "홍길동&이수홍, 박연수, 김자바-최명호";  
        String[] names = text.split("&|,|-");  
  
        // 기본 for문  
        for(int i=0; i<names.length; i++) {  
            System.out.println(names[i]);  
        }  
        System.out.println();  
  
        // 향상된 for문  
        for(String name : names) {  
            System.out.println(name);  
        }  
    }  
}
```

문자열 클래스

❖ 문자열 클래스 : String

키보드로 주민번호를 입력 받아서 남자인지, 여자인지를 판별하는 프로그램을 작성하세요?

단, 앞자리는 6자리, 뒷자리는 7자리 인지를 유효성 검사를 한다.

문자열 클래스

❖ String 클래스 : substring(), equals()

```
public class JuminCheck {  
  
    public static void main(String[] args) {  
  
        Scanner sc = new Scanner(System.in);  
        try {  
            System.out.println("주민번호 앞자리를 입력 하세요?");  
            String jumin1 = sc.nextLine();  
            System.out.println("주민번호 뒷자리를 입력 하세요?");  
            String jumin2 = sc.nextLine();          // jumin2="1234567"  
  
            String g = jumin2.substring(0, 1);      // 예외가 발생함  
  
            if(jumin1.equals("")) {  
                System.out.println("주민번호 앞자리를 입력하세요.");  
            }else if(jumin1.length() != 6) {  
                System.out.println("주민번호 앞자리 6자리를 입력하세요.");  
            }else if(jumin2.equals("")) {  
                System.out.println("주민번호 뒷자리를 입력하세요.");  
            }else if(jumin2.length() != 7) {  
                System.out.println("주민번호 뒷자리 7자리를 입력하세요.");  
            }else if(g.equals("1") || g.equals("3")) {  
                System.out.println("남자 입니다.");  
            }else if(g.equals("2") || g.equals("4")) {  
                System.out.println("여자 입니다.");  
            }else {  
                System.out.println("똑바로 입력 하세요.");  
            }  
        }catch(Exception e) {  
            System.out.println("잘못된 정보 입니다.");  
        }  
    }  
}
```

문자열 클래스

❖ StringBuffer 클래스

메소드
append(...)
insert(int offset, ...)
delete(int start, int end)
deleteCharAt(int index)
replace(int start, int end, String str)
StringBuilder reverse()
setCharAt(int index, char ch)

문자열 클래스

❖ StringBuffer 클래스

```
public class StringBufferTest {  
    public static void main(String[] args) {  
  
        StringBuffer sb1 = new StringBuffer("gemini");  
        System.out.println("sb1.length() : " + sb1.length());  
        System.out.println("sb1.capacity() : " + sb1.capacity());  
  
        sb1.append("A string buffer implements" + "a mutable sequence of characters");  
        System.out.println("sb1.length() : " + sb1.length());  
        System.out.println("sb1.capacity() : " + sb1.capacity());  
  
        StringBuffer sb2 = new StringBuffer();  
        System.out.println("sb2.length() : " + sb2.length());  
        System.out.println("sb2.capacity() : " + sb2.capacity());  
    }  
}
```


문자열 클래스

❖ StringBuffer 클래스

```
public class StringBufferTest2 {  
    public static void main(String[] args) {  
  
        // StringBuffer 객체 생성  
        StringBuffer sb1 = new StringBuffer("gemini");  
        System.out.println("sb1 = " + sb1);  
  
        // StringBuffer sb1에 문자열을 추가해 새로운 객체 생성  
        StringBuffer sb2 = sb1.append(" is beautiful");  
        System.out.println("sb2 = " + sb2);  
        System.out.println("sb1 = " + sb1);  
        if(sb1 == sb2) { // 주소값을 비교  
            System.out.println("같은 주소");  
        }else {  
            System.out.println("다른 주소");  
        }  
  
        // 정수형 데이터 형을 추가  
        System.out.println(sb1.append(1004));  
        System.out.println("sb1 = " + sb1);  
        System.out.println("sb2 = " + sb2);  
  
        // StringBuffer를 String으로 변환  
        String str = new String(sb1);  
        System.out.println(str.toUpperCase());  
    }  
}
```

문자열 클래스

❖ StringBuffer 클래스

```
public class StringBufferTest3 {  
    public static void main(String[] args) {  
  
        StringBuffer sb1 = new StringBuffer("gemini is beautiful");  
        System.out.println(sb1);  
  
        sb1.insert(10, "very"); // 인덱스 10 위치에 very 문자를 삽입  
        System.out.println(sb1);  
  
        sb1.insert(0, 1004);  
        System.out.println(sb1);  
    }  
}
```

문자열 클래스

❖ StringTokenizer 클래스

StringTokenizer 클래스는 구분기호를 이용해서 문자를 분리 시키는 역할을 한다.

```
StringTokenizer st = new StringTokenizer("문자열", "구분기호");
```

문자열 클래스

❖ StringTokenizer 클래스

메소드		설명
int	countTokens()	꺼내지 않고 남아있는 토큰의 수
boolean	hasMoreTokens()	남아 있는 토큰이 있는지 여부
String	nextToken()	토큰을 하나씩 꺼내옴

```
String text = "홍길동/이수홍/박연수";  
StringTokenizer st = new StringTokenizer(text, "/");
```

```
while( st.hasMoreTokens() ) {  
    String token = st.nextToken();  
    System.out.println(token);  
}
```

문자열 클래스

❖ StringTokenizer 클래스

```
class StringTokenizer01 {  
  
    public static void main(String[] args) {  
        StringTokenizer str =  
            new StringTokenizer("이순신#율지문덕#강감찬#광개토대왕", "#");  
  
        //      System.out.println(str.nextToken());      // 이순신  
        //      System.out.println(str.nextToken());      // 율지문덕  
        //      System.out.println(str.nextToken());      // 강감찬  
        //      System.out.println(str.nextToken());      // 광개토대왕  
        //      System.out.println(str.nextToken());      // 예외발생  
  
        // 파싱된 문자열이 모두 몇 개인지 구한다.  
        int cnt = str.countTokens();  
        System.out.println("파싱할 문자열의 총갯수-> " + cnt);  
  
        System.out.println(str.nextToken());  
        while (str.hasMoreTokens()) {                      // 토큰이 있으면  
            System.out.print(str.nextToken());              // 차례대로 파싱된 문자열을 얻어온다.  
        }  
    }  
}
```

문자열 클래스

❖ StringTokenizer 클래스

```
public class StringTokenizer02 {  
  
    public static void main(String[] args) {  
  
        String source1 = "한국 미국 태국 중국 이란";  
        StringTokenizer st1 = new StringTokenizer(source1, " ");  
        while (st1.hasMoreTokens()) {  
            System.out.println("st1.token:" + st1.nextToken());  
        }  
        System.out.println();  
  
        String source2 = "푸들,삼살개,풍산개,진돗개";  
        StringTokenizer st2 = new StringTokenizer(source2, ",");  
        while (st2.hasMoreTokens()) {  
            System.out.println("st2.token:" + st2.nextToken());  
        }  
        System.out.println();  
  
        // true : 구분기호(,)도 토큰에 포함을 해서 처리하라는 의미  
        StringTokenizer st3 = new StringTokenizer(source2, ",", true);  
        while (st3.hasMoreTokens()) {  
            System.out.println("st3.token:" + st3.nextToken());  
        }  
    }  
}
```

Wrapper 클래스

❖ Wrapper 클래스

Wrapper 클래스는 다양한 자료형 변환을 하기 위해서 사용된다.

기본 타입	포장 클래스
byte	Byte
char	Character
short	Short
int	Integer
long	Long
float	Float
double	Double
boolean	Boolean

Wrapper 클래스

❖ Wrapper 클래스

```
public class WrapperEx {  
  
    public static void main(String[] args) {  
  
        // int형 변수의 최대값과 최소값  
        System.out.println("max="+ Integer.MAX_VALUE);  
        System.out.println("min="+ Integer.MIN_VALUE);  
  
        // String형을 int형으로 형변환 : "20" ---> 20  
        int n = Integer.parseInt("20");  
        System.out.println(n);           // 20   숫자  
        System.out.println(n+10);        // 30   산술연산 가능함  
  
        // 10진수를 2진수로 변환  
        System.out.println("2진수:"+ Integer.toBinaryString(10));  
  
        // 10진수를 8진수로 변환  
        System.out.println("8진수:"+ Integer.toOctalString(10));  
  
        // 10진수를 16진수로 변환  
        System.out.println("16진수:"+ Integer.toHexString(10));  
    }  
}
```


Wrapper 클래스

❖ 박싱(Boxing)과 언박싱(Unboxing)

■ 박싱(Boxing)

1. 기본 자료형의 값을 포장 객체로 만드는 과정
2. heap메모리를 박스로 생각하고, stack메모리에 저장된 데이터를 heap메모리에 복사 하는것. (heap 메모리 박스에 집어 넣는다.)

■ 언박싱(Unboxing)

1. 포장 객체에서 기본 자료형의 값을 얻어내는 과정
2. heap메모리에 있는 데이터를 stack메모리로 가져오는 것.

Wrapper 클래스

❖ 박싱(Boxing)하는 방법

1. 생성자를 이용하는 방법

기본 타입의 값을 줄 경우	문자열을 줄 경우
<code>Byte obj = new Byte(10);</code>	<code>Byte obj = new Byte("10");</code>
<code>Character obj = new Character('가');</code>	
<code>Short obj = new Short(100);</code>	<code>Short obj = new Short("100");</code>
<code>Integer obj = new Integer(1000);</code>	<code>Integer obj = new Integer("1000");</code>
<code>Long obj = new Long(10000);</code>	<code>Long obj = new Long("10000");</code>
<code>Float obj = new Float(2.5F);</code>	<code>Float obj = new Float("2.5F");</code>
<code>Double obj = new Double(3.5);</code>	<code>Double obj = new Double("3.5");</code>
<code>Boolean obj = new Boolean(true);</code>	<code>Boolean obj = new Boolean("true");</code>

2. `valueOf()` 메소드를 이용하는 방법

```
Integer obj = Integer.valueOf(1000);  
Integer obj = Integer.valueOf("1000");
```

Wrapper 클래스

❖ 언박싱(Unboxing)하는 방법

기본 자료형 + Value() : **intValue()**

기본 타입의 값을 이용

byte num = obj.byteValue();

char ch = obj.charValue();

short num = obj.shortValue();

int num = obj.intValue();

long num = obj.longValue();

float num = obj.floatValue();

double num = obj.doubleValue();

boolean bool = obj.booleanValue();

Wrapper 클래스

❖ 박싱과 언박싱

```
public class WrapperEx1 {  
  
    public static void main(String[] args) {  
  
        // Integer num = new Integer(); // 오류발생  
  
        // 박싱(boxing) : heap메모리를 박스로 생각하고, stack메모리에 저장된 10을  
        //                      heap메모리에 복사 하는것. (박스에 집어 넣는다.)  
        int n = 10; // 지역변수(stack영역에 저장)  
        Integer num01 = new Integer(n); // 박싱(boxing)  
  
        // 언박싱(unboxing) : heap메모리에 있는 데이터를 stack메모리로 가져오는 것.  
        int n01 = num01.intValue(); // 언박싱(unboxing)  
  
        // 자료형 변환 : "20" --> 20  
        String s = "20";  
        Integer num02 = new Integer(s); // 박싱(boxing)  
        int n02 = num02.intValue(); // 언박싱(unboxing)  
    }  
}
```

Wrapper 클래스

❖ 박싱과 언박싱

```
public class WrapperEx2 {  
  
    public static void main(String[] args) {  
  
        // 박싱(boxing)  
        Integer obj1 = new Integer(10);  
        Integer obj2 = new Integer("200");  
        Integer obj3 = Integer.valueOf("300");  
  
        // 언박싱(unboxing)  
        int value1 = obj1.intValue();  
        int value2 = obj2.intValue();    // 자료형 변환 : "200" -> 200  
        int value3 = obj3.intValue();    // 자료형 변환 : "300" -> 300  
  
        System.out.println(value1);  
        System.out.println(value2);  
        System.out.println(value3);  
    }  
}
```

Wrapper 클래스

❖ 자동 박싱과 자동 언박싱

■ 자동 박싱

```
Integer obj = new Integer( 100 );           // 박싱
```

```
Integer obj = 100;                           // 자동 박싱
```

■ 자동 언박싱

```
int value = obj.intValue();                  // 언박싱
```

```
int value = obj;                            // 자동 언박싱
```

Wrapper 클래스

❖ 자동 박싱과 자동 언박싱

```
public class WrapperEx3 {  
  
    public static void main(String[] args) {  
  
        // 자동 박싱  
        // Integer obj = new Integer(100); // 박싱  
        Integer obj = 100; // 자동 박싱  
        System.out.println("언박싱:" + obj.intValue());  
        System.out.println("자동언박싱:" + obj);  
  
        // 언박싱  
        int value1 = obj.intValue();  
  
        // 자동 언박싱  
        int value2 = obj;  
  
        // 자동 언박싱  
        int result = obj + 100;  
        System.out.println("result:" + result);  
    }  
}
```

Wrapper 클래스

❖ 자동 박싱과 자동 언박싱

```
public class WrapperEx4 {  
  
    public static void main(String[] args) {  
  
        // Double d = new Double();           // 오류발생  
  
        Double d1 = new Double(3.14);          // 박싱  
        Double d11 = 3.14;                     // 자동 박싱  
  
        double n1 = d1.doubleValue();           // 언박싱  
        double n11 = d1;                       // 자동 언박싱  
  
        // 1.자료형 변환 : "42.195" ---> 42.195  
        Double d2 = new Double("42.195");      // 박싱  
        // Double d22 = "42.195";              // 오류발생  
  
        double n2 = d2.doubleValue();           // 언박싱  
        double n22 = d2;                       // 자동 언박싱  
  
        // 2.자료형 변환 : "42.195" ---> 42.195  
        double num = Double.parseDouble("42.195");  
        System.out.println("num="+ num);  
    }  
}
```


Wrapper 클래스

- ❖ 문자형을 기본 자료형으로 형변환

```
int num = Integer.parseInt(String s);
```

기본 타입의 값을 이용

```
byte    num = Byte.parseByte("10");
```

```
short   num = Short.parseShort("100");
```

```
int      num = Integer.parseInt("1000");
```

```
long     num = Long.parseLong("10000");
```

```
float    num = Float.parseFloat("2.5F");
```

```
double   num = Double.parseDouble("3.5");
```

```
boolean  bool = Boolean.parseBoolean("true");
```

Wrapper 클래스

❖ 문자형을 기본 자료형으로 형변환

```
public class WrapperEx5 {  
  
    public static void main(String[] args) {  
  
        // 문자형을 기본 자료형으로 변환  
  
        int value1 = Integer.parseInt("10");  
        double value2 = Double.parseDouble("3.14");  
        boolean value3 = Boolean.parseBoolean("true");  
  
        System.out.println("value1:" + value1);  
        System.out.println("value2:" + value2);  
        System.out.println("value3:" + value3);  
    }  
}
```

Wrapper 클래스

❖ Wrapper 클래스 응용

```
class TestClass {  
    private int member;                // 필드  
  
    public void setValue(int value) {  
        member = value;  
    }  
  
    public int getValue() {  
        return member;  
    }  
}  
  
public class GenericTest01 {  
    public static void main(String[] args) {  
        TestClass obj01 = new TestClass();  
        obj01.setValue(3);              // 정상적인 호출가능  
        System.out.println("되돌리는 값은->" + obj01.getValue());  
        obj01.setValue(3.4);            // 에러발생  
        System.out.println("되돌리는 값은->" + obj01.getValue());  
        obj01.setValue("이해할 수 있다."); // 에러발생  
        System.out.println("되돌리는 값은->" + obj01.getValue());  
    }  
}
```

Wrapper 클래스

❖ Wrapper 클래스 응용

```
class TestClass2 {
    private Object member;
    public void setValue(Object value) {
        member = value;
    }
    public Object getValue() {
        return member;
    }
}

public class GenericTest02 {
    public static void main(String[] args) {
        TestClass2 obj01 = new TestClass2();

        // Object value = new Integer(3);      업캐스팅 + 박싱
        // Object value = 3;                    업캐스팅 + 자동박싱
        obj01.setValue(3);
        System.out.println("되돌리는 값은->" + obj01.getValue());

        // 다운 캐스팅 + 언박싱
        int n = ((Integer) (obj01.getValue())).intValue();

        // Object value = new Double(3.14);     업캐스팅 + 박싱
        // Object value = 3.14;                 업캐스팅 + 자동박싱
        obj01.setValue(3.4);
        System.out.println("되돌리는 값은->" + obj01.getValue());

        // 다운 캐스팅 + 언박싱
        double d = ((Double) (obj01.getValue())).doubleValue();

        // Object value = new String("이해할 수 있다."); 업캐스팅 + 박싱
        obj01.setValue("이해할 수 있다.");
        System.out.println("되돌리는 값은->" + obj01.getValue());

        // 다운 캐스팅
        String str = (String) obj01.getValue();
    }
}
```