

제16장

java.lang패키지-String클래스

1. String 클래스

- String 클래스의 생성자들은 아래와 같이 오버로딩 되어 있다.

```
//기본생성자
String()
//바이트 배열로 String객체 생성하는 생성자
String(byte[] bytes)
//바이트 배열을 지정한 문자셋으로 디코딩하는 생성자
String(byte[] bytes, Charset charset)
//바이트 배열의 길이를 지정하고 지정한 문자셋으로 디코딩하는 생성자
String(byte[] bytes, int offset, int length, Charset charset)
```

파일의 내용을 읽거나,
네트워크를 통해 받은 데이터는
보통 **byte[]** 배열이므로
이것을 문자열로 변환하기 위해 사용

디코딩 : 사람이 알아보기 쉽게 만드는 과정(디지털 신호를 재생하는 것)
인코딩 : 기계가 알아보기 쉽게 만드는 과정(아날로그 신호 -> 디지털 신호로 압축)
예) 코덱(Codec) : 인코딩과 디코딩의 합성한 말

문자셋(Charset) 종류

1. UTF-8 : 유니코드 인코딩
2. EUC-KR : 한글완성형 인코딩
3. ISO8859 : 유럽쪽 언어 인코딩

- 키보드로부터 읽은 바이트 배열을 문자열로 변환

```
byte[] b = new byte[100];
int readCount = System.in.read();
String str = new String(b, 0, readCount-2);
System.out.println(str.toString());
```

입력 내용 : Java\r\n

여기서 \r\n은 엔터키를 의미한다.
엔터키는 \r(캐리지리턴), \n(라인피드)로
아스키코드 값은 13, 10이어서 읽은 바이트
에서 -2를 해주는 것에 주목하자.

2. String 클래스의 메서드 - 1

▣ 문자열의 추출이나 비교, 찾기, 분리, 대체 등과 같은 다양한 메서드들이 많다.

사용빈도가 높은 메서드를 위주로 살펴보자.(매우 중요-활용할 곳이 너무 많다.)

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

2. String 클래스의 메서드 - 2

■ 문자 추출(charAt(int index)) 메서드

- 매개값으로 주어진 인덱스의 문자를 리턴한다.

```
String str = "안녕하세요";  
char ch = str.charAt(2);  
System.out.println(ch);
```

Console
<terminated> CarExam
하

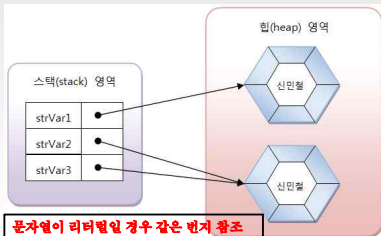
사용ex) 주민등록번호..

■ 문자열 비교(equals()) 메서드

```
String strVar1 = new String("신민철");  
String strVar2 = "신민철";  
String strVar3 = "신민철";
```

```
strVar1 == strVar2 → false  
strVar2 == strVar3 → true
```

```
strVar1.equals(strVar2) → true  
strVar2.equals(strVar3) → true
```



2. String 클래스의 메서드 - 3

- 바이트 배열로 변환시키는 메서드(**getBytes()**) - 매우 중요

```
//바이트 배열로 인코딩  
byte[] b = "안녕하세요".getBytes();  
//바이트 배열을 문자열로 디코딩  
String str = new String(b);
```

파일에 내용을 쓰거나,
네트워크를 통해 데이터를
보낼때 문자열을 **byte[]** 로
인코딩하기 위해 사용

- 특정 문자셋으로 인코딩 된 바이트 배열 얻는 방법

```
try {  
    //문자열을 특정 문자셋으로 인코딩  
    byte[] b1 = "안녕하세요".getBytes("EUC-KR");  
    byte[] b2 = "반갑습니다.".getBytes("UTF-8");  
    //특정 문자셋으로 바이트 배열 디코딩  
    String str = new String(b1, "EUC-KR");  
    System.out.println(str);  
} catch (UnsupportedEncodingException e) {}
```

2. String 클래스의 메서드 - 4

■ 특정 문자열 찾는 메서드(**indexOf()**)

- 매개값으로 들어가는 문자열의 시작되는 인덱스를 리턴한다.
- 주어진 문자열이 존재하지 않으면, -1을 리턴한다.

```
String str = "자바 공부 중입니다.";
int index = str.indexOf("공부");
System.out.println("시작문자 인덱스 : " + index);
```

```
Console
<terminated> CarExample [Java Applicati
시작문자 인덱스 : 3
```

■ 문자열의 길이를 구하는 메서드(**length()**)

```
String str = "자바 공부 중입니다.";
int length = str.length();
System.out.println("길이 : " + length);
```

```
Console
<terminated> CarExamp
길이 : 11
```

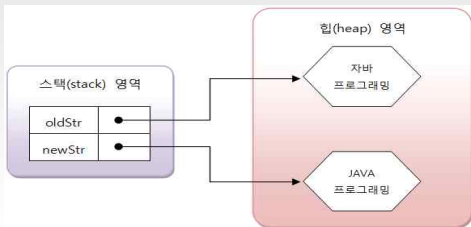
2. String 클래스의 메서드 - 5

■ 문자열을 대치시키는(`replace()`)

- 첫 번째, 매개 값인 문자열을 찾아서 두 번째, 매개 값인 문자열로 대치한 새로운 문자열을 리턴 해준다.

```
String oldStr = "자바 프로그래밍";  
String newStr = oldStr.replace("자바", "JAVA");  
System.out.println(newStr);
```

```
Console  
<terminated> Ex [Java Application]  
JAVA 프로그래밍
```



2. String 클래스의 메서드 - 6

▣ 문자열을 잘라내는 (**substring()**)

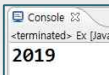
– substring(int beginIndex, int endIndex)


주어진 시작과 끝 인덱스 사이의 문자열을 추출한다.(단, **마지막 인덱스는 미포함**)

– substring(int beginIndex)

주어진 인덱스 이후부터 끝까지 문자열 추출

```
String serialNo = "20190828-075";  
String year = serialNo.substring(0, 4);  
System.out.println(year);
```



Console 
<terminated> Ex [Java A
2019

```
String serialNo = "20190828-075";  
String month = serialNo.substring(4);  
System.out.println(month);
```



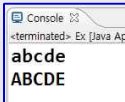
Console 
<terminated> Ex [Java A
0828-075

2. String 클래스의 메서드 - 7

■ 알파벳 대,소문자 변경하는 메서드 (**toLowerCase()**, **toUpperCase()**)

– 주어진 문자열들을 전부 소문자로 혹은 전부 대문자로 바꾼다.

```
String str = "abcDe";  
System.out.println(str.toLowerCase());  
System.out.println(str.toUpperCase());
```



Console

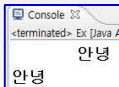
<terminated> Ex [Java Ap

abcde
ABCDE

■ 문자열 앞뒤 공백 잘라내는 메서드 (**trim()**)

– 주어진 문자열의 앞, 뒤에 있는 공백을 전부 없앤다.

```
String str = "    안녕    ";  
System.out.println(str);  
String trim = str.trim();  
System.out.println(trim);
```



Console

<terminated> Ex [Java A

안녕

안녕

2. String 클래스의 메서드 - 8

▣ 문자열로 변환해주는 메서드 (**valueOf()**)

– 기본 타입의 값을 문자열로 변환시켜준다.(GUI프로그래밍에서 자주 등장한다.)

```
static String valueOf(boolean b)
static String valueOf(char c)
static String valueOf(int i)
static String valueOf(long l)
static String valueOf(double d)
static String valueOf(float f)
```

감사합니다.

