프로그래밍 언어 활용 강의안
(String class)

String 메소드

문자열의 추출, 비교, 찾기, 분리, 변환등과 같은 다양한 메소드 가짐
사용 빈도 높은 메소드

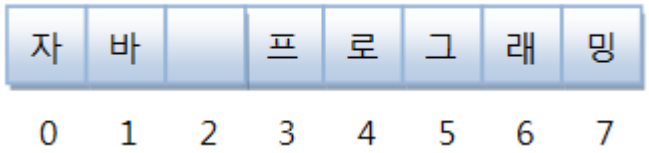
리턴타입	메소드명(매개변수)	설명
char	charAt(int index)	특정 위치의 문자 리턴
boolean	equals(Object anObject)	두 문자열을 비교
byte[]	getBytes()	byte[]로 리턴
byte[]	getBytes(Charset charset)	주어진 문자셋으로 인코딩한 byte[]로 리턴
int	indexOf(String str)	문자열내에서 주어진 문자열의 위치를 리턴
int	length()	총 문자의 수를 리턴
String	replace(CharSequence target, CharSequence replacement)	target 부분을 replacement 로 대치한 새로운 문자열을 리턴
String	substring(int beginIndex)	beginIndex 위치에서 끝까지 짤라낸 새로운 문자열을 리턴
String	substring(int beginIndex, int endIndex)	beginIndex 위치에서 endIndex 전까지 짤라낸 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

String 클래스

문자 추출(charAt())

매개 값으로 주어진 인덱스의 문자 리턴

```
String subject = "자바 프로그래밍";
char charValue = subject.charAt(3);
```



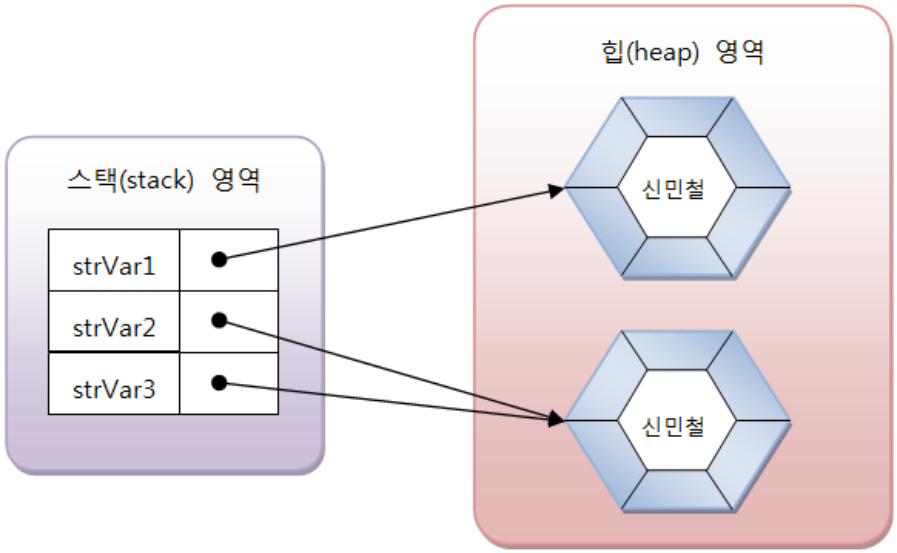
문자열 비교(equals())

문자열 비교할 때 == 연산자 사용하면 원하지 않는 결과 발생!

```
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
```

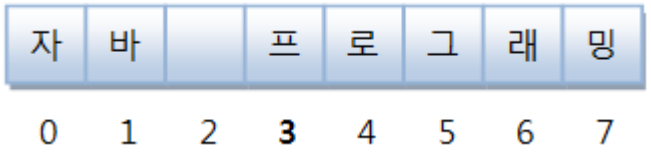


String 클래스

문자열 찾기(indexOf())

매개값으로 주어진 문자열이 시작되는 인덱스 리턴
주어진 문자열이 포함되어 있지 않으면 -1 리턴

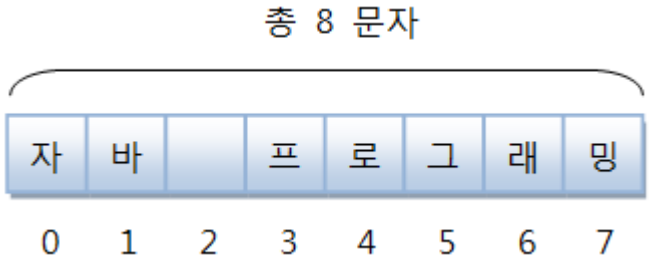
```
String subject = "자바 프로그래밍";
int index = subject.indexOf("프로그래밍");
```



특정 문자열이 포함되어 있는지 여부 따라 실행 코드 달리할 때 사용

문자열 길이(length()) – 공백도 문자에 포함

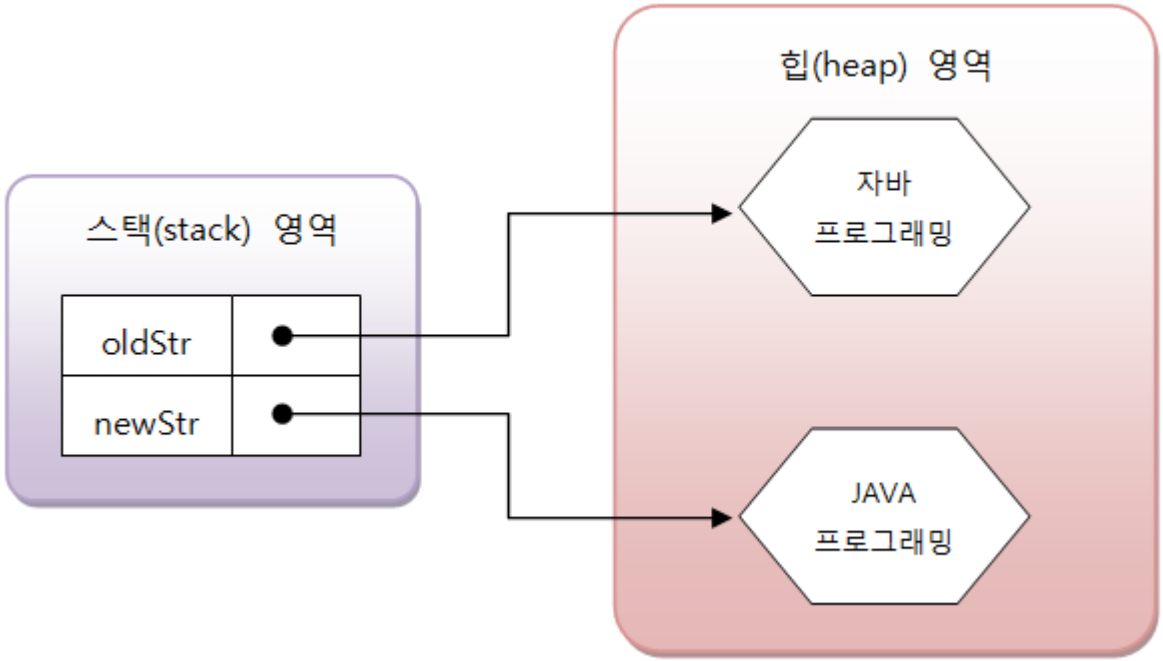
```
String subject = "자바 프로그래밍";
int length = subject.length();
```



String 클래스

문자열 대치(replace())

첫 번째 매개값인 문자열 찾음
두 번째 매개값인 문자열로 대치
새로운 문자열 리턴



String 클래스

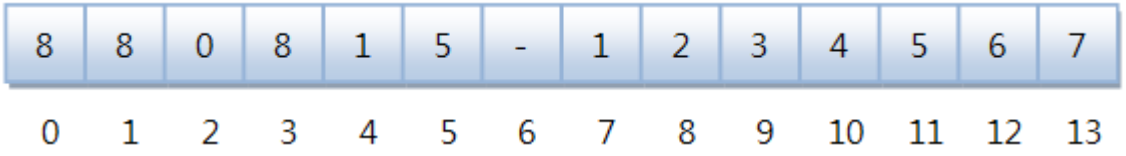
문자열 잘라내기(substring())

substring(int beginIndex, int endIndex)

주어진 시작과 끝 인덱스 사이의 문자열 추출

substring(int beginIndex)

주어진 인덱스 이후부터 끝까지 문자열 추출

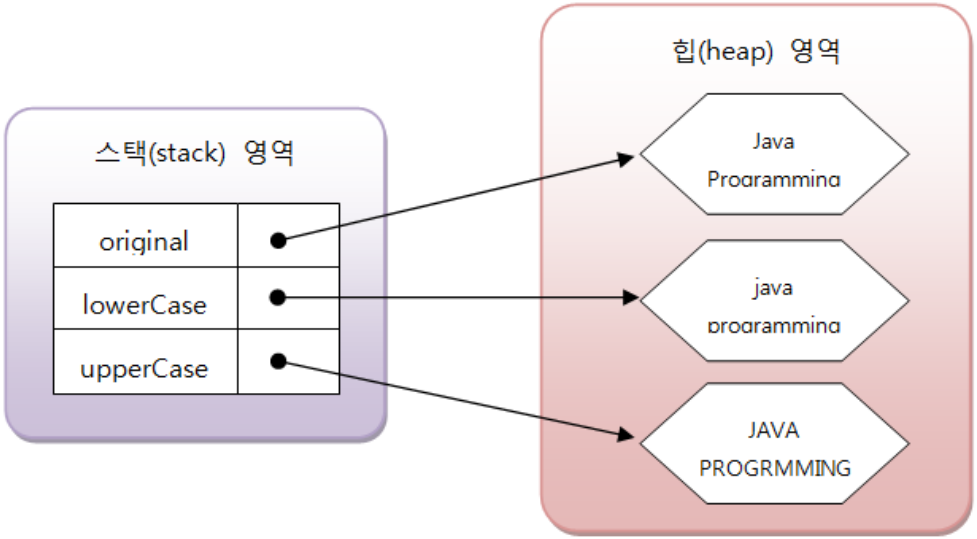



```
String ssn  = "880815-1234567";
String firstNum  = ssn.substring(0,  6);
String secondNum  = ssn.substring(7);
```

String 클래스

알파벳 소·대문자 변경 (toLowerCase(), toUpperCase())

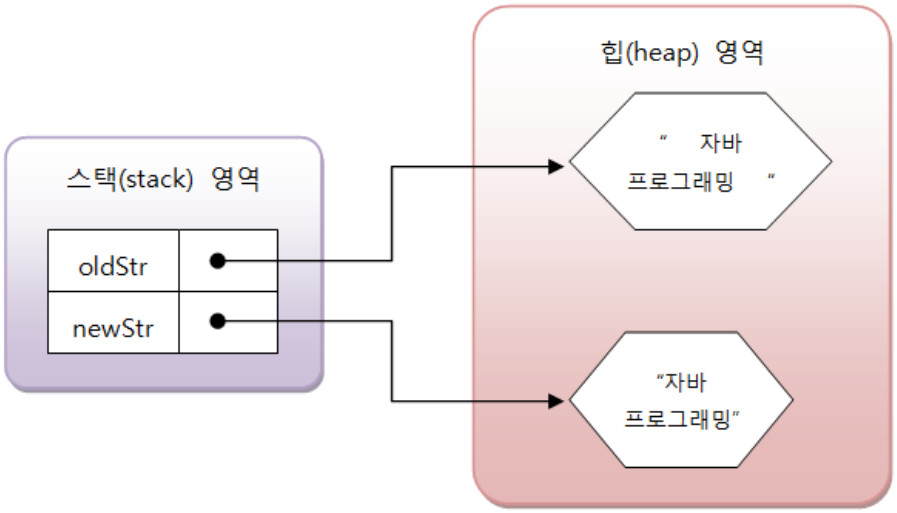
```
String original = "Java Programming";
String lowerCase = original.toLowerCase();
String upperCase = original.toUpperCase();
```



String 클래스

문자열 앞뒤 공백 잘라내기(trim())

```
String oldStr = "    자바 프로그래밍    ";
String newStr = oldStr.trim();
```



String 클래스

문자열 변환(valueOf())

기본 타입의 값을 문자열로 변환

```
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)
```

String 클래스

생성자

byte[] 배열을 문자열로 변환하는 생성자

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

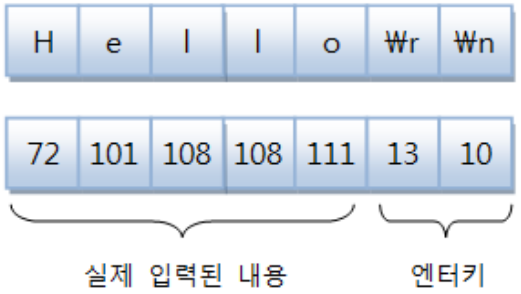
```
//배열 전체를 String 객체 생성
String str = new String(byte[] bytes);
//지정한 문자셋으로 디코딩
String str = new String(byte[] bytes, String charsetName);

//배열의 offset 인덱스 위치부터 length 개 만큼 String 객체 생성
String str = new String(byte[] bytes, int offset, int length);
//지정한 문자셋으로 디코딩
String str = new String(byte[] bytes, int offset, int length, String charsetName)
```

키보드로부터 읽은 바이트 배열을 문자열로 변환

```
byte[] bytes = new byte[100];
int readByteNo = System.in.read(bytes);
String str = new String(bytes, 0, readByteNo-2);
```

입력내용:

바이트 배열 내용 :



문자열 분리 방법

- String의 split() 메소드 이용
- java.util.StringTokenizer 클래스 이용

String의 split()

- 정규표현식을 구분자로 해서 부분 문자열 분리
- 배열에 저장하고 리턴

```
홍길동&이수홍,박연수,김자바-최명호
```

```
String[] names = text.split("&|,|-");
```

StringTokenizer 클래스

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

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

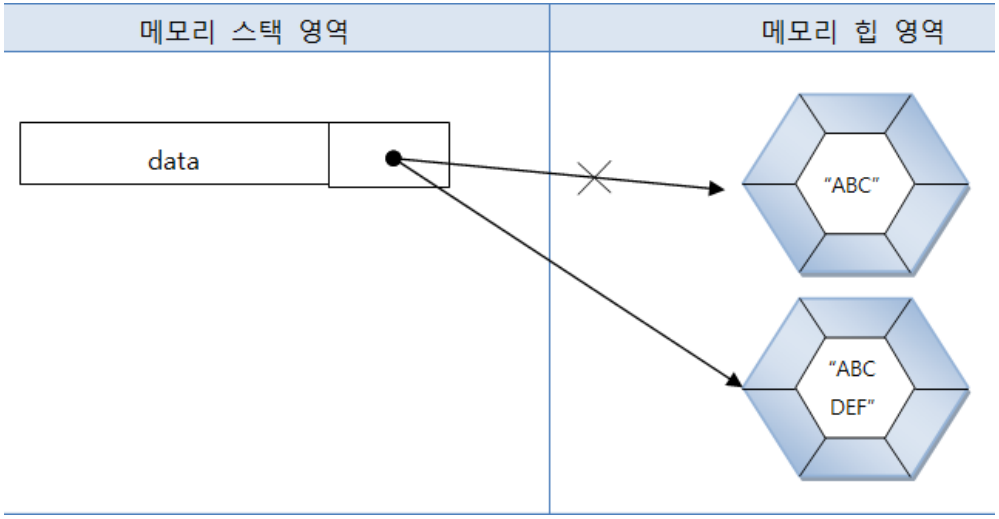
메소드		설명
int	countTokens()	꺼내지 않고 남아있는 토큰의 수
boolean	hasMoreTokens()	남아 있는 토큰이 있는지 여부
String	nextToken()	토큰을 하나씩 꺼내옴

StringBuffer, StringBuilder 클래스

문자열 결합 연산자 +

String 은 내부의 문자열 수정 불가 -> 대치된 새로운 문자열 리턴

```
String data = "ABC";
data += "DEF";
```



StringBuffer, StringBuilder

버퍼(buffer:데이터를 임시로 저장하는 메모리)에 문자열 저장
버퍼 내부에서 추가, 수정, 삭제 작업 가능
멀티 스레드환경: StringBuffer 사용
단일 스레드환경: StringBuilder 사용

```
StringBuilder sb = new StringBuilder();
StringBuilder sb = new StringBuilder(16);
StringBuilder sb = new StringBuilder("Java");
```

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)