

Java 프로그래밍 길잡이

✓ 원리를 알면 IT가 맛있다

Java Programming for Beginners



chapter 08.

자바의 핵심 클래스

- String 및 StringBuffer 클래스 사용법을 학습한다.
- Wrapper 클래스 용도 및 사용법을 학습한다.
- Random 클래스 사용법을 학습한다.
- Date , Calendar 클래스 사용법을 학습한다.
- SimpleDateFormat 클래스 사용법을 학습한다.
- StringTokenizer 클래스 사용법을 학습한다.

문자열 생성 방법

- 가. String 클래스 이용
- 나. StringBuffer 클래스 이용
- 다. + (연결연산자) 이용
- 라. toString 메소드 이용

String 클래스를 이용한 문자열 생성 방법 2가지

가. 리터럴 이용

동일한 문자열인 경우에는 새로 생성하지 않고 재사용한다.
따라서 name과 name2의 위치값을 동일하다.

```
예> String name = "홍길동";  
     String name 2= "홍길동";
```

나. new 이용

동일한 문자열일지라도 매번 새롭게 생성된다.
따라서 s와 s2의 위치값은 서로 다르다.

```
예> String s  = new String("hello");  
     String s2 = new String("hello");
```

String 클래스의 핵심 메소드 정리.

메소드	설명
length	문자열의 길이를 반환한다.
equals	문자열이 일치하는지를 검사한다.
equalsIgnoreCase	대소문자 구별없이 문자열이 일치하는지를 검사한다.
substring	부분열을 구한다.
replace	문자열을 대치한다.
toUpperCase	문자열을 대문자로 바꾼다.
toLowerCase	문자열을 소문자로 바꾼다.
charAt	특정 문자를 반환한다.
trim	공백문자를 제거한다.
concat	문자열을 연결한다.

String 는 한번 생성하면 변경되지 않는다. (철사)

StringBuffer 클래스를 이용한 문자열 생성 방법

가. new 이용

```
StringBuffer buffer = new StringBuffer();
```

StringBuffer 클래스 핵심 메소드

메소드	설명
length	문자열의 길이를 반환한다.
capacity	초기 버퍼크기를 지정한다.
append	버퍼에 문자열을 추가한다.
insert	버퍼에 문자열을 삽입한다.

StringBuffer 는 동적으로 크기가 변경된다. (고무줄)

문자열과 다른 데이터형이 + 를 만나면 문자열로 변경된다.

예>

```
String s = 1 + 2 + 3 + "hello"; // 6hello
```

```
String s2 = "hello" + 1 + 2 + 3; // hello123
```


기본 데이터를 문자열로 변경하는 메소드이다.

```
예> int n = 10;  
     char c = 'A';  
     float f = 3.14F;
```

```
String s = Integer.toString(n);  
String s2 = Character.toString( c );  
String s3 = Float.toString( f );
```

* **String.valueOf(값)** 을 더 많이 사용한다.

```
String s = String.valueOf(n);  
String s2 = String.valueOf(c);  
String s3 = String.valueOf(f);
```

8개의 기본데이터에 해당하는 8개의 클래스 파일들을 부르는 이름이다.

기본데이터	Wrapper
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
char	Character
boolean	Boolean

기본데이터를 이용해서 작업할 때, 유용하게 사용될 수 있는 메소드와 상수, 변수들을 제공한다.

따라서 개발자들은 기본데이터 처리시 반드시 알고 있어야 되는 클래스들이다.

사용방법:

일반적으로 클래스 사용법을 따르면 된다. (객체생성)

문법:

```
Integer n = new Integer(100);
```

```
int n2 = 100;
```

```
System.out.println( n ); // 100
```

```
System.out.println( n2 ); //100
```

출력하면 모두 동일한 값의 100이 출력된다.

차이점은 n은 참조변수이고 , n2는 기본형 변수이다.

결국 , 쓸데없이 구분 지어진 형태이다. (결과값이 동일하니까...)

- 오토박싱(auto boxing) 과 오토언박싱 (auto unboxing)

가. 오토박싱

자동으로 기본형 데이터가 Wrapper 로 형변환 되는 것을 의미한다.

```
예> int n = 10;  
    Integer n2 = n;
```

나. 오토 언박싱

자동으로 Wrapper 클래스가 기본형 데이터로 형변환되는 것을 의미한다.

```
예> Integer n = new Integer( 100 );  
    int n2 = n;
```

- 임의의 값을 발생시키는 유틸리티 클래스이다.
- java.util 패키지에 존재한다. 따라서 반드시 import 해야 된다.

메소드	설명
nextInt	임의의 정수값을 구한다.
nextInt(범위)	0부터 입력범위-1까지의 임의의 정수값을 구한다.
nextBoolean	임의의 불린값을 구한다.
nextFloat	0.0과 1.0 사이의 임의의 float 값을 구한다.
nextDouble	0.0과 1.0 사이의 임의의 double 값을 구한다.

문법:

```
Random ran = new Random();  
int n = ran.nextInt();
```

자바에서 날짜 데이터를 위해서 다음 2가지가 제공된다.

가. java.util.Date

```
사용예 > Date d = new Date();  
          System.out.println( d );
```

나. java.util.Calendar

```
사용예> Calendar cal = Calendar.getInstance();  
  
// 메소드와 상수 이용하여 데이터를 얻는다.
```

- 날짜 데이터를 특정 포맷으로 출력할 때 사용 되는 유틸리티 클래스이다.
- java.text 패키지에 존재한다. 따라서 반드시 import 해야 된다.


문법:

```
SimpleDateFormat sdf = new SimpleDateFormat( “형식문자” );  
String str = sdf.format( new Date() );
```

형식 문자	기능
y	년도 표시
M	월 표시
d	일 표시
H	시간(0~23)표시
m	분 표시
s	초 표시
h	시간(1~12)표시
a	AM/PM 표시

특정 구분자로 구분되어진 통문자열을 , 구분자를 기준으로 문자열을 추출할 수 있는 유틸리티 클래스이다. 구분자의 기본은 공백이다.

예> String str = “홍길동/20/서울”;
String str2 = “홍길동 20 서울”



토큰(token)

문법:

```
StringTokenizer stk = new StringTokenizer( 문자열 , 구분자 );  
String str = stk.nextToken();
```

또는

```
for( stk.hasMoreTokens() ){  
    System.out.println( stk.nextToken() );  
}
```




Thank you
