

# 제19장

## 유용한 클래스들



# 1. 정규 표현식과 Pattern클래스

## ■ 정규 표현식(Regular Expression)을 작성하는 방법

- 문자열이 정해져 있는 형식으로 구성되어 있는지 확인할 때 사용한다.

ex) 이메일, 전화번호, 우편번호, 비밀번호 등

- 문자 또는 숫자 기호와 반복적인 기호와 서로 결합된 문자열 형태로 이루어진다.

## ■ 정규 표현식 작성 방법

- 우측표는 기본적으로 알아야 된다.

- 전화번호

- (02|010)-\d{3,4}-\d{4}
- ex) 02-123-4678
- 010-2989-1038
- - 이메일
- \w+@\w+\.\w+(\.\w+)?
- ex) perpear@naver.com
- perpear@naver.co.kr

기호	설명	
[ ]	한 개의 문자	[abc] a, b, c 중 하나의 문자
		[^abc] a, b, c 이외의 하나의 문자
		[a-zA-Z] a~z, A~Z 중 하나의 문자
\d	한 개의 숫자, [0-9]와 동일	
\s	공백 <b>s : space</b> 약자	
\w	한 개의 알파벳 또는 한 개의 숫자, [a-zA-Z_0-9]와 동일	
?	없음 또는 한 개	
*	없음 또는 한 개 이상	
+	한 개 이상	
{n}	정확히 n 개	
{n,}	최소한 n 개	
{n, m}	n 개에서부터 m 개까지	
()	그룹핑	

반도  
기호

# 1. 정규 표현식과 Pattern클래스

## ■ Pattern 클래스

– 정규 표현식으로 문자열을 검증하는 역할을 한다.

```
//010으로 시작하는 휴대폰 전화번호 패턴 정의
String regExp = "(010)-\\d{3,4}-\\d{4}";
String data = "010-9977-8823";

//패턴에 일치하는지 확인하는 작업
boolean result = Pattern.matches(regExp, data);
if(result)
    System.out.println("전화번호 양식이 맞습니다.!");
else
    System.out.println("똑바로 입력하세요.");
```

Console

```
<terminated> ReflectionEx [Java Application] C:\Program F
전화번호 양식이 맞습니다.!
```

## 2. Format클래스

### ■ 형식(Format) 클래스

– 숫자와 날짜 등을 원하는 형식으로 문자열을 변환하는데 사용한다.

– 종류

\* 숫자형식 : DecimalFormat클래스

\* 날짜형식 : SimpleDateFormat클래스

○ \* 0 : 자리수를 의미 함. 0은 해당 자리의 값이

○       없더라도 0을 표시한다.

○ # : 자리수를 의미 함. 0과 달리 해당 자리의

○       값이 없으면 표시하지 않는다.

패턴 예	1234567.89 - > 변환 결과
0	1234568
0.0	1234567.9
0000000000.000 0	0001234567.8900
#	1234568
##	1234567.9
#####.### #	1234567.89
#.0	1234567.9

## 2. Format클래스

### ■ 숫자 형식 클래스(DecimalFormat)

– 적용할 패턴을 선택하여 생성자의 매개 값으로 주면서 객체를 생성한다.

```
DecimalFormat dFormat = new DecimalFormat("#,###");  
String str = dFormat.format(7798.634);  
System.out.println(str);
```

Console   
<terminated> Ref  
7,799

### ■ 날짜 형식 클래스(SimpleDateFormat)

```
SimpleDateFormat sFormat = new SimpleDateFormat("yyyy-MM-dd");  
String strDate = sFormat.format(new Date());  
System.out.println(strDate);
```

Console   
<terminated> ReflectionEx  
2019-09-08

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 구분자 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		

### 3. StringTokenizer클래스

■ 문자열을 분리하는 방법은 2가지가 있다.

- String클래스의 split()메서드를 이용하는 방법
- java.util.StringTokenizer의 메서드를 조합하여 분리하는 방법

■ String의 **split()**

- 정규 표현식을 구분자로 하여 문자열을 토큰으로 분리 후, String[]을 리턴한다.
- 만약, **구분자 사이에 데이터가 없다면 공백을 리턴**한다.

```
String str = "돼지고기/어묵,소고기-오징어&라면%곱배기";
String[] tokens = str.split("/|,|-|&|%");
for(String token : tokens) {
    System.out.println(token);
}
```

```
Console 23
<terminated> Reflect
돼지고기
어묵
소고기
오징어
라면
곱배기
```

### 3. StringTokenizer클래스

#### ■ StringTokenizer의 메서드 사용

— 구분자 사이에 단어가 없어도 공백을 리턴하지 않고 다음 단어를 리턴한다.

```
String str = "돼지고기/어묵,소고기&오징어&라면,곰배기";
StringTokenizer sTokenizer = new StringTokenizer(str,"/,|/|&|%");
while(sTokenizer.hasMoreTokens()) {
    String token = sTokenizer.nextToken();
    System.out.println(token);
}
```

Console <terminated> Re  
돼지고기  
어묵  
소고기  
오징어  
라면  
곰배기

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

String클래스 : split()의 경우는 정규표현식을 사용하므로 가변적인 요소가 많은 문자열이나 정확한 분리가 필요할 때 사용하기에 범용적이며, 유용하다.

StringTokenizer클래스 : 속도적인 측면에서는 빠르다.

## 4. Date클래스

- 날짜를 표현하기 위한 클래스이지만, **TimeStamp와 같은 역할을 하는 용도로** 많이 사용되는 클래스이다.

날짜 정보를 객체 간에 서로 주고 받을 때, 매개값으로 주로 사용한다.

```
Date date = new Date();  
String str1 = date.toString();  
System.out.println(str1);
```

```
SimpleDateFormat sFormat = new SimpleDateFormat("yyyy년 MM월 dd일 a hh시 mm분");  
String str2 = sFormat.format(date);  
System.out.println(str2);
```

Console

<terminated> ReflectionEx [Java Application] C:\Program Files\Java\jre1.8

Sun Sep 08 17:52:13 KST 2019

2019년 09월 08일 오후 05시 52분



## 5. Calendar클래스

- 달력을 표현한 추상 클래스이다.
- OS에 설정되어진 시간대(TimeZone) 기준의 Calendar객체를 얻는 방법(싱글톤)

```
Calendar calendar = Calendar.getInstance(); //싱글톤 패턴
```

- 다른 시간대에 있는 Calendar객체를 얻는 방법(TimeZone클래스 이용)

```
TimeZone tZone = TimeZone.getTimeZone("America/Los_Angeles");  
Calendar calendar = Calendar.getInstance(tZone);
```

- 날짜 및 시간 정보 얻는 방법

```
int year = calendar.get(Calendar.YEAR);  
int month = calendar.get(Calendar.MONTH) + 1;  
int day = calendar.get(Calendar.DAY_OF_MONTH);  
int hour = calendar.get(Calendar.HOUR_OF_DAY);  
int minute = calendar.get(Calendar.MINUTE);  
int second = calendar.get(Calendar.SECOND);
```

감사합니다.

