

## 제 33 강

# java.util 패키지

교재:p190~198

# 목차

## 1. java.util 패키지

1. java.util 패키지
2. 시간 처리

# 1. java.util 패키지

## java.util 패키지란?

- util (utility)

컴퓨터 분야에서 유틸리티란, 사용자의 편리성을 향상하는 유용하고 실용적인 소프트웨어를 의미

→ java.util 패키지(Package)는 java 프로그래밍에 유용한 클래스들을 모아둔 것

대표적인 클래스로는 날짜와 관련된 Date, Calendar 가 있으며, 자료구조와 관련된 Collection 프레임워크 관련 클래스들이 포함되어 있다.

## 2. 시간 처리

java에서는 시간에 대한 처리를 할 수 있도록  
java.util 패키지 내에서  
Calendar 클래스와 Date 클래스, SimpleDateFormat 클래스를 제공!

### [1] System.currentTimeMillis( )

: 현재 운영체제의 시각을 long 타입으로 반환

- 1970년 1월 1일을 기준으로 현재까지 시간이며 1/1000 초 단위까지 환산

## 2. 시간 처리

### <실습> Exam-69.java

```
public class Date1 {  
    public static void main(String[] args) {  
        long start = System.currentTimeMillis();  
  
        System.out.println("시작 시간: "+start);  
        int a=0;  
        for(int i=1;i<10000000;i++) {  
            a++;  
        }  
        long end=System.currentTimeMillis();  
        System.out.println("종료 시간:"+end);  
        System.out.println("걸린 작업시간:"+end-start);  
    }  
}
```

#### <실행 결과>

시작 시간: 1598256100428
종료 시간: 1598256100431
걸린 작업시간: 3

걸린작업시간이 3 이라는 것은  
milli second 이므로, 3/1000 초가 걸린 것을 의미한다.

## 2. 시간 처리

### [2] java.util.Calendar

: Java에서 제공하는 날짜 관리 클래스

- 추상클래스 이므로 직접 인스턴스 생성 불가: getInstance( ) 메서드를 활용하여 객체 생성
- 싱글 턴 패턴(Singleton Pattern): 객체 사용 시마다 인스턴스 생성하는 것이 아니라, 하나의 인스턴스로 사용하는 방법

## 2. 시간 처리

### <실습> Exam-70.java

```
public class Date2 {  
    public static void main(String[] args) {  
        //      Calendar a = new Calendar(); 불가  
        Calendar a = Calendar.getInstance();  
        Calendar b= new GregorianCalendar();  
  
        System.out.println(a.toString());  
        System.out.println(b.toString());  
    }  
}
```

#### <실행 결과>

```
java.util.GregorianCalendar[time=1598256702897,areFieldsSet=true,areAllFieldsSet=true,  
java.util.GregorianCalendar[time=1598256702916,areFieldsSet=true,areAllFieldsSet=true,  
areFieldsSet=true,areAllFieldsSet=true,areFieldsSet=true,areAllFieldsSet=true]
```

Calendar 클래스 인스턴스 생성

1. GregorianCalendar 클래스 활용
2. getInstance( ) 메서드 활용

두 방법 모두 생성되는 인스턴스 종류는 같다.

## 2. 시간 처리

### <실습> Exam-71.java

Calendar 클래스에서 get( ) 메서드로 정보 가져오기

```
public class Date3 {  
    public static void main(String[] args) {  
        Calendar a = Calendar.getInstance();  
        int year=a.get(Calendar.YEAR);  
        int month = a.get(Calendar.MONTH);  
        int date= a.get(Calendar.DATE);  
  
        System.out.println(year+"년"+month+"월"+date+"일");  
        System.out.print("오늘은 이번주에서 몇 일째 인가?:");  
        System.out.println(a.get(Calendar.DAY_OF_WEEK));  
        //1=일요일 2=월요일 .. 7=토요일  
        System.out.print("이번 년도에서 오늘이 몇 일째 인가?:");  
        System.out.println(a.get(Calendar.DAY_OF_YEAR));  
  
        System.out.print("이번 달은 며칠까지 있는가?:");  
        System.out.println(a.getActualMaximum(Calendar.DATE));  
    }  
}
```

<실행 결과>

```
2020년7월24일  
오늘은 이번주에서 몇 일째 인가?:2  
이번 년도에서 오늘이 몇 일째 인가?:237  
이번 달은 며칠까지 있는가?:31
```



## 2. 시간 처리

### <실습> Exam-72.java

Calendar 클래스에서 set( ) 메서드로 날짜 지정하기

```
public class Date4 {  
    public static void main(String[] args) {  
        Calendar today = Calendar.getInstance();  
        Calendar endOfYear = Calendar.getInstance();  
        Calendar Christmas = Calendar.getInstance();  
  
        endOfYear.set(Calendar.MONTH, 11);  
        endOfYear.set(Calendar.DATE, 31);  
        long diff=endOfYear.getTimeInMillis()-today.getTimeInMillis();  
        System.out.println("연말 까지 남은 날:"+diff/(24*60*60*1000)+"일");  
  
        Christmas.set(2020, 11, 25);  
        diff=Christmas.getTimeInMillis()-today.getTimeInMillis();  
        System.out.println("크리스마스까지 남은 날:"+diff/(24*60*60*1000)+"일");  
    }  
}
```

<실행 결과>

연말 까지 남은 날:129일  
크리스마스까지 남은 날:123일

Calendar 클래스 set( ) 주의사항!

- month는 0부터 11까지!

## 2. 시간 처리

### [3] java.util.Date

: Java에서 제공하는 날짜 관리 클래스

- JDK 1.0 부터 제공되는 클래스이나 여러 문제로 대부분의 메서드와 생성자 사용x

## 2. 시간 처리

### <실습> Exam-73.java

Date 객체 생성하기

```
public class Date5 {  
    public static void main(String[] args) {  
        Date today = new Date();  
        System.out.println(today);  
        long a=System.currentTimeMillis();  
        Date today2=new Date(a);  
        System.out.println(today2);  
    }  
}
```

<실행 결과>

```
Mon Aug 24 17:49:57 KST 2020  
Mon Aug 24 17:49:57 KST 2020
```

## 2. 시간 처리

### <실습> Exam-74.java

long, Calendar, Date 간의 변환

```
public class Date6 {  
    public static void main(String[] args) {  
        //Calendar -> Date  
        Calendar calendar = Calendar.getInstance();  
        Date date=calendar.getTime();  
        System.out.println(date);  
  
        //Date->Calendar  
        calendar.setTime(date);  
  
        //Calendar->long  
        long time=calendar.getTimeInMillis();  
        System.out.println(time);  
  
        //long->Calendar  
        calendar.setTimeInMillis(time);  
    }  
}
```

<실행 결과>

```
Mon Aug 24 17:54:22 KST 2020  
1598259262309
```

## 2. 시간 처리

### [4] SimpleDateFormat

: 날짜를 형식화 하는 클래스

- 날짜 형식을 원하는 형태로 출력하기 위한 클래스

기호	의미	기호	의미
Y	년	a	오전/오후
M	월	H	시간
d	일	m	분
E	요일	s	초

## 2. 시간 처리

### <실습> Exam-75.java

SimpleDateFormat를 사용해 원하는 형식으로 출력

```
public class Format2 {  
    public static void main(String[] args) {  
        Date day=new Date();  
        String patternKorea="yyyy-MM-dd";  
        String patternUSA="MM-dd-yyyy";  
        String patternUK="dd-MM-yyyy";  
        String pattern1 ="E요 HH시 mm분 ss초";  
  
        SimpleDateFormat p1=new SimpleDateFormat(patternKorea);  
        SimpleDateFormat p2=new SimpleDateFormat(patternUSA);  
        SimpleDateFormat p3=new SimpleDateFormat(patternUK);  
        SimpleDateFormat p4=new SimpleDateFormat(pattern1);  
  
        System.out.println("현재 날짜:"+day);  
        System.out.println("한국 형식(년,월,일):"+p1.format(day));  
        System.out.println("미국 형식(월,일,년):"+p2.format(day));  
        System.out.println("영국 형식(일,월,년):"+p3.format(day));  
        System.out.println(p4.format(day));  
    }  
}
```

<실행 결과>

```
현재 날짜:Mon Aug 24 18:02:38 KST 2020  
한국 형식(년,월,일):2020-08-24  
미국 형식(월,일,년):08-24-2020  
영국 형식(일,월,년):24-08-02020  
월요일 18시 02분 38초
```