

# 날짜와 시간 클래스

## ▣ 날짜와 시간 클래스

- 자바는 컴퓨터의 날짜 및 시각을 **읽을** 수 있도록 java.util 패키지에서 Date 와 Calendar 클래스를 제공하고 있다.
- 또한 날짜와 시간을 **조작**할 수 있도록 java.time 패키지에서 LocalDateTime 등의 클래스를 제공한다.

클래스	설명
Date	날짜 정보를 전달하기 위해 사용
Calendar	다양한 시간대별로 날짜와 시간을 얻을 때 사용
LocalDateTime	날짜와 시간을 조작할 때 사용

## ◎ Date 클래스

- Date는 날짜를 표현하는 클래스로 객체 간에 날짜 정보를 주고받을 때 사용한다.
- Date 클래스에는 여러 개의 생성자가 선언되어 있지만 대부분 Deprecated되어 Date() 생성자만 주로 사용된다.
- Date() 생성자는 컴퓨터의 현재 날짜를 읽어 Date 객체로 만든다.
- 현재 날짜를 문자열로 얻고 싶다면 toString() 메소드를 사용할 수 있지만 영문으로 출력되기 때문에 우리가 원하는 형식이 아니다. 원하는 문자열로 얻고 싶다면 SimpleDateFormat 클래스와 함께 사용해야 한다.

```
Date now = new Date();
```

## ◎ Calendar 클래스

- Calendar 클래스는 달력을 표현하는 추상 클래스이다.
- 날짜와 시간을 계산하는 방법이 지역과 문화에 따라 다르기 때문에  
특정 역법(날짜와 시간을 매기는 방법)에 따르는 달력은 자식 클래스에서 구현하도록 되어 있다.
- 특별한 역법을 사용하는 경우가 아니라면 직접 하위 클래스를 만들 필요는 없고,  
Calendar 클래스의 정적 메소드인 getInstance() 메소드를 이용하면 컴퓨터에 설정되어 있는 시간대(TimeZone)를  
기준으로 Calendar 하위 객체를 얻을 수 있다.
- Calendar가 제공하는 날짜와 시간에 대한 정보를 얻기 위해서는 get() 메소드를 이용한다.  
get() 메소드의 매개값으로 Calendar에 정의된 상수를 주면, 상수가 의미하는 값을 리턴한다.

```
Calendar now = Calendar.getInstance();
```

```
int year    = now.get(Calendar.YEAR);      //년도를 리턴
int month   = now.get(Calendar.MONTH) + 1; //월을 리턴
int day     = now.get(Calendar.DAY_OF_MONTH); //일을 리턴
int week    = now.get(Calendar.DAY_OF_WEEK); //요일을 리턴
int amPm    = now.get(Calendar.AM_PM);     //오전/오후를 리턴
int hour    = now.get(Calendar.HOUR);      //시를 리턴
int minute  = now.get(Calendar.MINUTE);    //분을 리턴
int second  = now.get(Calendar.SECOND);    //초를 리턴
```

- Calendar 클래스의 오버로딩된 다른 getInstance() 메소드를 이용하면 미국/로스앤젤레스와 같은 다른 시간대의 Calendar를 얻을 수 있다.
- 알고 싶은 시간대의 TimeZone 객체를 얻어, getInstance() 메소드의 매개값으로 넘겨주면 된다.

```
TimeZone timeZone = TimeZone.getTimeZone("America/Los_Angeles");  
Calendar now = Calendar.getInstance( timeZone );
```

- America/Los\_Angeles와 같은 시간대 ID는 TimeZone.getAvailableIDs() 메소드가 리턴하는 값 중 하나를 사용하면 된다.

```
public class PrintTimeZoneID {  
    public static void main(String[] args) {  
        String[] availableIDs = TimeZone.getAvailableIDs();  
        for(String id : availableIDs) {  
            System.out.println(id);  
        }  
    }  
}
```

실행 결과

```
Africa/Abidjan  
Africa/Accra  
Africa/Addis_Ababa  
...
```

## ◎ LocalDateTime 클래스 - 날짜와 시간 조작

- Date 와 Calendar는 날짜와 시간 정보를 얻기에는 충분하지만, 날짜와 시간을 조작할 수는 없다.
- 이때는 java.time 패키지의 LocalDateTime 클래스가 제공하는 메소드를 이용하면 매우 쉽게 날짜와 시간을 조작할 수 있다.
- LocalDateTime 클래스를 이용해서 현재 컴퓨터의 날짜와 시간을 얻는 방법은 아래와 같다.

```
LocalDateTime now = LocalDateTime.now();
```

메소드(매개변수)	설명
minusYears(long)	년 빼기
minusMonths(long)	월 빼기
minusDays(long)	일 빼기
minusWeeks(long)	주 빼기
plusYears(long)	년 더하기
plusMonths(long)	월 더하기
plusWeeks(long)	주 더하기
plusDays(long)	일 더하기
minusHours(long)	시간 빼기
minusMinutes(long)	분 빼기
minusSeconds(long)	초 빼기
minusNanos(long)	나노초 빼기
plusHours(long)	시간 더하기
plusMinutes(long)	분 더하기
plusSeconds(long)	초 더하기

◎ LocalDateTime 클래스 - 날짜와 시간 비교

- LocalDateTime 클래스는 날짜와 시간을 비교할 수 있는 메소드를 제공한다.
- 비교를 위해 특정 날짜와 시간으로 LocalDateTime 객체를 얻는 방법은 year부터 second까지 매개값을 모두 int 타입 값으로 제공하면 된다.

```
LocalDateTime target = LocalDateTime.of(year, month, dayOfMonth, hour, minute, second);
```

리턴 타입	메소드(매개변수)	설명
boolean	isAfter(other)	이후 날짜인지?
	isBefore(other)	이전 날짜인지?
	isEqual(other)	동일 날짜인지?
long	until(other, unit)	주어진 단위(unit) 차이를 리턴

## ※ ChronoUnit 클래스

클래스	설명
ChronoUnit.YEARS	전체 년 차이
ChronoUnit.MONTHS	전체 월 차이
ChronoUnit.WEEKS	전체 주 차이
ChronoUnit.DAYS	전체 일 차이
ChronoUnit.HOURS	전체 시간 차이
ChronoUnit.SECONDS	전체 초 차이
ChronoUnit.MILLIS	전체 밀리초 차이
ChronoUnit.NANOS	전체 나노초 차이