

Java에서 date, time, Calendar 공부하기

작성일: 20.02.06

작성자: 도원진

1. Time Stamp

1.1 timestamp

- **unix timestamp**
 - 1970년 1월 1일 00:00:00 [협정 세계시](#)(UTC) 부터의 경과 시간을 초로 환산하여 정수로 나타낸 것이다.
- **timezone**
 - 지구의 자전에 따른 지역 사이에 생기는 낮과 밤의 차이를 인위적으로 조정하기 위해 고안된 시간의 구분선을 일컫는다.

1.2 java.util.Date

- **Date**
 - 출력값: Wed Feb 05 09:26:20 KST 2020

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

- SimpleDateFormat 사용, 출력값: 2020년 02월 05일 09시 26분 20초

```
SimpleDateFormat sdf = new SimpleDateFormat("yyyy년 MM월 dd일 hh시 mm분 ss초");  
String strNow2 = sdf.format(now);  
System.out.println(strNow2);
```

- **Calendar**
 - static final int 형 상수
 - YEAR
 - MONTH
 - DAY_OF_MONTH
 - DAY_OF_WEEK
 - AM_PM
 - AM
 - PM
 - HOUR
 - MINUTE
 - SECOND
 - MILLISECOND

```
public static void calendarByTimezone(TimeZone timeZone) {  
  
    Calendar now =(timeZone == null ?
```

```

        Calendar.getInstance()
        : Calendar.getInstance(timeZone)) ;
// static final int형 필드
/*
 * YEAR, MONTH, DAY_OF_MONTH, DAY_OF_WEEK
 */
System.out.println("Calendar: " + now);
int year = now.get(Calendar.YEAR);
int month = now.get(Calendar.MONTH);
int day = now.get(Calendar.DAY_OF_MONTH);
int week = now.get(Calendar.DAY_OF_WEEK); //Calendar 의 필드
SUNDAY(1) MONDAY, TUESDAY, WEDNESDAY, ,SATUDAY(7)
int amPm = now.get(Calendar.AM_PM);
String strAmPm = (amPm == Calendar.AM ? "AM":"PM");
int hour = now.get(Calendar.HOUR);
int minute = now.get(Calendar.MINUTE);
int second = now.get(Calendar.SECOND);
int millisecond = now.get(Calendar.MILLISECOND);

System.out.println(year + "년 " + month + "월 " + day + "일 " +
week + "요일 "
+ strAmPm + " " + hour + "시(24시) " + minute + "분 " +
second + "초 " + millisecond + "밀리초 ");
}

```

◦ TimeZone

```

TimeZone tz = TimeZone.getTimeZone("America/Los_Angeles");
System.out.println(TimeZone.getAvailableIDs()); //TimeZone 목록을 알고
싶을 때

```

- 현재시각, 1초, 1분, 1시간, 1일, 1달, 변경하기
- Date Formatting

```

public static void Formatting() {
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy년 MM월 dd일 E요일 a HH:mm:ss\n"
+ "올해의 D번째 날\n이 달의 d번째 날");
    String strDate = sdf.format(new Date());
    System.out.println(strDate);
}

```

2020년 02월 05일 수요일 오후 20:31:46

올해의 36번째 날

이 달의 5번째 날

1.3 네이버 개발자 블로그글

Date, Calendar 클래스의 문제점

- 불변객체가 아니다
 - 멀티쓰레드에서 다른 한 곳에서 시간을 수정하면 잘못된 정보를 사용하게 된다.
- 상수필드 남용

- int형 상수가 많아 오류를 찾기 어렵다.
- 헷갈리는 월 지정
- 일관성 없는 요일 int 형 상수
- Calendar객체를 만드는 비용을 치워서 Date객체로 결과물을 낸다.

1.3.1 LocalDate, LocalTime, LocalDateTime

```

LocalDate currentDate = LocalDate.now(); // 컴퓨터의 현재 날짜 정보 2018-07-26
LocalDate myDate = LocalDate.of(int year, int month, int dayOfMonth); //년,월,일

getYear();
getMonth(); (Month 열거값나옴 JANUARY, FEBRUARY)
getMonthValue(); 월(1,2,3,...)
getDayOfYear(); 년의 몇 번째 일
getDayOfMonth(); 월의 몇 번째 일
getDayOfWeek(); 요일(MONDAY, TUESDAY,...)
isLeapYear(); 윤년여부
////////////////////////////////////
////////////////////////////////////

LocalTime currentTime = LocalTime.now(); // 컴퓨터의 현재 시간 정보. 결과 :
16:24:02.408
LocalTime targetTime = LocalTime.of(int hour, int minute, int second, int
nanoOfSecond);
int getHour() 시간
int getMinute() 분
int getSecond() 초
int getNano() 나노초

////////////////////////////////////
////////////////////////////////////

LocalDateTime 시간,날짜 동시에 필요할 때 사용.
LocalDateTime currentTime = LocalDateTime.now(); // 컴퓨터의 현재 날짜와 시
간 정보. 결과 : 2018-07-26T16:34:24.757
LocalDateTime targetDateTime = LocalDateTime.of(int year, int month, int
dayOfMonth, int hour, int minute, int second, int nanoOfSecond);
ZonedDateTime utcDateTime = ZonedDateTime.now(ZoneId.of("UTC"));
ZonedDateTime seoulDateTime = ZonedDateTime.now(ZoneId.of("Asia/Seoul"));

```