

14.API II

1) 날짜(Calendar, Gregorian Calendar) API

자바에서 날짜와 시간을 표현할 때 많이 쓰는 클래스인 Calendar는 싱글톤 패턴을 적용한 클래스 이다. 때문에 New 객체를 생성하여 클래스를 사용할 수 없고 대신 getInstance메소드를 이용해 객체 생성한 뒤 원하는 정보를 get()함수로 가져온다.

Ex) 1. Calendar형

```
1  //Calendar cal = new Calendar();           //싱글톤이라 에러
2  Calendar cal = Calendar.getInstance();      //getter를 이용해 객체생성
3  System.out.println(cal);                   // 가독성 나쁜 장황한 데이터 출력
4  int year = cal.get(Calendar.YEAR);          // 연도
5  int month = cal.get(Calendar.MONTH) + 1;    // 월 0,1,2 ... 0월부터 세기때문에 +1
6  int day = cal.get(Calendar.DAY_OF_MONTH);   // 일
7  int week = cal.get(Calendar.DAY_OF_WEEK);   // 이번주의 몇일째 일1 월2 화3 수4 ..토7
8  int hour24 = cal.get(Calendar.HOUR_OF_DAY); // 24시간 단위
9  int ampm = cal.get(Calendar.AM_PM);        // 0(오전) 1(오후)
10 int hour = cal.get(Calendar.HOUR);          // 12시간 단위
11 int minute = cal.get(Calendar.MINUTE);      // 분
12 int second = cal.get(Calendar.SECOND);      // 초
13 int millisec = cal.get(Calendar.MILLISECOND); // 밀리세컨
14
15 System.out.printf("%d년 %d월 %d일 ", year, month, day); // 2022년 3월 31일 목요일
16
17 switch (week) {
18     case 1:
19         System.out.println("일요일");
20         break;
21     case 2:
22         System.out.println("월요일");
23         break;
24     case 3:
25         System.out.println("화요일");
26         break;
27     case 4:
28         System.out.println("수요일");
29         break;
30     case 5:
31         System.out.println("목요일");
32         break;
33     case 6:
34         System.out.println("금요일");
35         break;
36     case 7:
37         System.out.println("토요일");
38         break;
39 }
40 System.out.print(ampm == 0 ? "오전" : "오후");
41 System.out.printf("%d시 %d분 %d초 %d\n", hour, minute, second, millisec);
42 //오후4시 3분 25초 474
43 System.out.printf("%d시 %d분 %d초 %d", hour24, minute, second, millisec);
44 //16시 3분 25초 474
45
46 // %d(정수) %f(실수) %s(문자열) %c(문자)
47 // %tY(년) %tm(월) %td(일) %ta(요일) %tH(24시간) %tp(오전 /오후) %tl(12시) %tM(분) %tS(초)
48 // Calendar형 cal객체에 담겨진 여러 데이터를 %tY, %tH 형식으로 데이터 추출하여 사용
49
50 System.out.printf("%tY년 %tm월 %td일 %ta요일 %tp %tl시 %tM분 %tS초 \n",
51     cal, cal, cal, cal, cal, cal, cal, cal, cal);
52 System.out.printf("%1$tY년 %1$tm월 %1$td일 %1$ta요일 %1$tp %1$tl시 %1$tM분 %1$tS초", cal);
53 //t앞에 1$영령어는 뒤에 서술한 자료받을 객체중 첫번째 객체 자료 받는다는 뜻
54 //2022년 03월 31일 Thu요일 pm 4시 03분 25초
```

2. Gregorian Calendar형

Calendar를 상속 받고 있기 때문에 Calendar형과 사용법이 동일하다.

객체를 생성할 수 있다는 점에서 차이점이 있다.

Ex)

```

1 GregorianCalendar cal = new GregorianCalendar();
2 GregorianCalendar gcThat = new GregorianCalendar(2022, 2, 11, 9, 30); //
3 //Calendar형과 사용 방법이 동일하나 객체를 생성할수있고
4 //매개변수에 특정날짜 명시 설정할수있고 입력하지 않으면 현재시각대를 의미한다.
5
6 System.out.println(cal); // 가독성 나쁜 장황한 데이터 출력
7 int year = cal.get(Calendar.YEAR); // 연도
8 int month = cal.get(Calendar.MONTH) + 1; // 월 0,1,2 ... 0월부터 세기때문에 +1
9 int day = cal.get(Calendar.DAY_OF_MONTH); // 일
10 int week = cal.get(Calendar.DAY_OF_WEEK); // 이번주의 몇일째 일1 월2 화3 수4 ..토7
11 int hour24 = cal.get(Calendar.HOUR_OF_DAY); // 24시간 단위
12 int ampm = cal.get(Calendar.AM_PM); // 0(오전) 1(오후)
13 int hour = cal.get(Calendar.HOUR); // 12시간 단위
14 int minute = cal.get(Calendar.MINUTE); // 분
15 int second = cal.get(Calendar.SECOND); // 초
16 int millisec = cal.get(Calendar.MILLISECOND); // 밀리세컨
17

```

3. Data

자바에서 시간과 날짜를 구하기 위한 또 하나의 클래스인데 API문서에서 찾아보면 대부분의 메소드들이

Deprecated(더 이상 사용되지 않는다) 라고 규정되어 있다. 사용은 가능하지만 권고하지 않는것이다.

Java™ Platform Standard Ed. 7		Compares two dates for equality.
All Classes	int	getDate() Deprecated. As of JDK version 1.1, replaced by <code>Calendar.getDate(Calendar.DAY_OF_MONTH)</code> .
Packages	int	getDay() Deprecated. As of JDK version 1.1, replaced by <code>Calendar.getDate(Calendar.DAY_OF_WEEK)</code> .
java.applet	int	getHours() Deprecated. As of JDK version 1.1, replaced by <code>Calendar.getTime(Calendar.HOUR_OF_DAY)</code> .
java.awt	int	getMinutes() Deprecated. As of JDK version 1.1, replaced by <code>Calendar.getTime(Calendar.MINUTE)</code> .
java.awt.color	int	getMonth() Deprecated. As of JDK version 1.1, replaced by <code>Calendar.getTime(Calendar.MONTH)</code> .
java.awt.datatransfer	int	getSeconds() Deprecated. As of JDK version 1.1, replaced by <code>Calendar.getTime(Calendar.SECOND)</code> .
java.awt.dnd	long	getTime() Returns the number of milliseconds since January 1, 1970, 00:00:00 GMT represented by this Date object.
java.awt.event	int	getTimeZoneOffset() Deprecated. As of JDK version 1.1, replaced by <code>-Calendar.getTime(Calendar.ZONE_OFFSET) + Calendar.getTime(Calendar.DST_OFFSET) / (60 * 1000)</code> .
java.awt.font	int	getYear() Deprecated. As of JDK version 1.1, replaced by <code>Calendar.getTime(Calendar.YEAR) - 1900</code> .
java.awt.geom	int	hashCode() Returns a hash code value for this object.
java.awt.im	static long	parse(String s) Deprecated. As of JDK version 1.1, replaced by <code>DateFormat.parse(String s)</code> .
java.awt.im.snt	void	setDate(int date) Deprecated. As of JDK version 1.1, replaced by <code>Calendar.set(Calendar.DAY_OF_MONTH, int date)</code> .
Package	void	setHours(int hours) Deprecated. As of JDK version 1.1, replaced by <code>Calendar.set(Calendar.HOUR_OF_DAY, int hours)</code> .
DataLineInfo	void	setMinutes(int minutes) Deprecated. As of JDK version 1.1, replaced by <code>Calendar.set(Calendar.MINUTE, int minutes)</code> .
DataOutput	void	setMonth(int month) Deprecated. As of JDK version 1.1, replaced by <code>Calendar.set(Calendar.MONTH, int month)</code> .
DataOutputStream	void	setSeconds(int seconds) Deprecated. As of JDK version 1.1, replaced by <code>Calendar.set(Calendar.SECOND, int seconds)</code> .
DataOutputStream		
DataSource		
DataTruncation		
DatatypeConfigurationException		
DatatypeConstants		
DatatypeConstants.Field		
DatatypeConverter		
DatatypeConverterInterface		
DatatypeFactory		
Date		
Date		
DateFormat		
DateFormat.Field		
DateFormatProvider		
DateFormatsymbols		
DateFormatsymbolsProvider		
DateFormatter		
DateTimeAtCompleted		
DateTimeCreation		
DateTimeProcessing		
DeltaTimeSyntax		
DebugGraphics		
DecimalFormat		
DecimalFormatSymbols		
DecimalFormatSymbolsProvider		
DeclaredType		
DeclHandler		
DefaultBoundedRangeModel		
DefaultButtonModel		
DefaultCaret		
DefaultCellEditor		
DefaultColorSelectionModel		
DefaultComboBoxModel		
DefaultDesktopManager		
DefaultEditorKit		
DefaultEditorKit.BEEPAction		
DefaultEditorKit.CopyAction		
DefaultEditorKit.CutAction		

Data 클래스뿐만 아니라 Calendar 클래스 역시 모호한 문제가 있어 사용을 지양하고 있다. 그 중 사용하면서 느끼는 문제점은 Calendar와 Date는 월 지정이 배열 인덱스같이(0부터 1월)되어있고, 두 클래스의 요일 상수가 다르다는 점이다. Calendar 객체는 일요일이 1부터 시작, Date 객체는 일요일이 0부터 시작한다.

4. SimpleDateFormat

위에서 보았듯 날짜와 시간관련 클래스들을 사용하기가 까다로운 편이다. 장황하게 코드가 길어지는걸

피할수없다. 때문에 특정 날짜와 시간들을 사용자의 필요에 맞게 형식을 만들어 출력할 수 있게 도와주는 것이

SimpleDateFormat클래스이다.

Ex)

```
1 public static void main(String[] args) {
2     Date now1 = new Date();
3     Calendar now2 = Calendar.getInstance();
4     GregorianCalendar now3 = new GregorianCalendar();
5
6     SimpleDateFormat sdf = new SimpleDateFormat("yyyy년 MM월 dd일 E요일 a hh시 mm분 ss초");
7     SimpleDateFormat sdf1 = new SimpleDateFormat("올해 몇번째 날 D 올해 몇번째 주 W");
8
9     //날짜타입을 원하는 포맷으로
10    //yyyy(년도4자리) yy(년도 2자리) M(3) MM(03) d(9) dd(09)
11    //E(요일) a(오전/오후) H(24시간) h(12시간) m(분) s(초)
12    //w(이번주가 올해 시작부터 몇번째 주인지) D(오늘이 올해 시작부터 며칠째인지 )
13
14    System.out.println(sdf.format(now1));//format() 매개변수에 Date형이 들어가야함
15    // 2022년 03월 31일 Thu요일 PM 10시 52분 27초
16    System.out.println(sdf.format(now2.getTime()));//getTime의 리턴타입이 Date형임
17    // 2022년 03월 31일 Thu요일 PM 10시 52분 27초
18    System.out.println(sdf.format(now3.getTime()));
19    // 2022년 03월 31일 Thu요일 PM 10시 52분 27초
20    System.out.println(sdf1.format(now3.getTime()));
21    //올해 몇번째 날 90 올해 몇번째 주 5
22 }
```