

Java DateTime

- `Date`는 JDK1.0, `Calendar`는 JDK1.1 부터 제공되었다.
- JDK1.8부터 `java.time` 패키지 로 `Calendar`와 `Date`의 단점을 개선한 클래스들이 추가되었다.

Java 1.8 이전 버전의 SDK에서 날짜와 시간을 다루는 `java.util.Date` 클래스와 `java.util.Calendar` 클래스는 사용하기가 불편하고 직관적이지 않으며 또한 `java.util.Date`와 `SimpleDateFormat`는 Thread-Safe 하지 않아서 잠재적인 동시성 문제를 가지고 있다. 많이 늦었지만 JDK 8에서는 개선된 날짜와 시간 API가 제공된다.

기존 API

- `java.util.Date` - 날짜와 시간, 기본 시간대를 사용하여 출력.
- `java.util.Calendar` - 날짜와 시간, 날짜를 조작하는데 더 많은 메소드 제공.
- `java.text.SimpleDateFormat` - 날짜와 달력을위한 형식 (날짜 -> 텍스트), 변환 (텍스트 -> 날짜).

JAVA8 에서 추가된 API

- `java.time.LocalDate` - 날짜(시간 포함하지 않음), 타임존 사용하지 않음.
- `java.time.LocalDateTime` - 시간(날짜 포함하지 않음), 타임존 사용하지 않음.
- `java.time.LocalDateTime` - 날짜 및 시간, 타임존 사용하지 않음.
- `java.time.ZonedDateTime` - 날짜 및 시간, 타임존 사용.
- `java.time.DateTimeFormatter` - `java.time`에 대한 형식 (날짜 -> 텍스트), 변환 (텍스트 -> 날짜)
- `java.time.Duration` - 시간을 초 단위 및 나노초 단위로 측정한다.
- `java.time.Period` - 시간을 년, 월, 일로 측정한다.
- `java.time.TemporalAdjuster` - 날짜를 조정한다.

1. java.util.Date

1-1. 생성자

```
package kr.green.utildate;

import java.util.Date;

public class Ex01_Date {
    public static void main(String[] args) {
        // 생성자 1
        Date date1 = new Date(); // 오늘 날짜

        long msec1 = date1.getTime();
        long msec2 = System.currentTimeMillis();
        System.out.println("1970년 1월 1일 9:00:00 이후 밀리초 : " + msec1);
        System.out.println("1970년 1월 1일 9:00:00 이후 밀리초 : " + msec2);

        // 생성자 2
        Date date2 = new Date(msec2); // 오늘 날짜
        System.out.println("날짜 1 : " + date1);
```

```

        System.out.println("날짜 2 : " + date2);
    }
}

```

결과

```

1970년 1월 1일 9:00:00 이후 밀리초 : 1633402219749
1970년 1월 1일 9:00:00 이후 밀리초 : 1633402219749
날짜 1 : Tue Oct 05 11:50:19 KST 2021
날짜 2 : Tue Oct 05 11:50:19 KST 2021

```

```

package kr.green.utildate;

import java.text.SimpleDateFormat;
import java.util.Date;
// @Deprecated 태그는 클래스나 메소드 등을 더 이상 사용이 권장하지 않는 경우에 사용한다.
// 사용이 권장되지 않는다는 것은 사용을 불가능하다는 것은 아니다. 다만 권장을 하지 않고 차후
// 에 없어질 수도 있다는 것을 의미한다.

public class Ex02_Date {
    @SuppressWarnings("deprecation")
    public static void main(String[] args) {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss");
        Date date1 = new Date(1, 0, 1);
        // year : 1900년 이후의 년 수
        // month : 0 ~ 11
        // date : 1-31.

        Date date2 = new Date(1, 0, 1, 1, 1);
        // hrs : 0~23
        // min : 0~59

        Date date3 = new Date(1, 0, 1, 1, 1, 1);
        // sec : 0~59

        Date date4 = new Date(1, 1, 31); // 2월 31일

        System.out.println("날짜 1 : " + date1);
        System.out.println("날짜 2 : " + date2);
        System.out.println("날짜 3 : " + date3);
        System.out.println("날짜 4 : " + date4);
        System.out.println();
        System.out.println("날짜 1 : " + sdf.format(date1));
        System.out.println("날짜 2 : " + sdf.format(date2));
        System.out.println("날짜 3 : " + sdf.format(date3));
        System.out.println("날짜 4 : " + sdf.format(date4));
    }
}

```

결과

날짜 1 : Tue Jan 01 00:00:00 KST 1901
날짜 2 : Tue Jan 01 01:01:00 KST 1901
날짜 3 : Tue Jan 01 01:01:01 KST 1901
날짜 4 : Sun Mar 03 00:00:00 KST 1901

날짜 1 : 1901-01-01 12:00:00
날짜 2 : 1901-01-01 01:01:00
날짜 3 : 1901-01-01 01:01:01
날짜 4 : 1901-03-03 12:00:00

1-2. Getter & setter

```
package kr.green.utildate;

import java.lang.reflect.Method;
import java.text.SimpleDateFormat;
import java.util.Date;

public class Ex03_Date {
    @SuppressWarnings("deprecation")
    public static void main(String[] args) {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd(E) hh:mm:ss");
        Date date = new Date();

        Method[] methods = date.getClass().getMethods();
        for(Method m : methods) {
            if(m.getName().startsWith("get"))
                System.out.println(m.getName() + " : " + m);
        }
        System.out.println();
        for(Method m : methods) {
            if(m.getName().startsWith("set"))
                System.out.println(m.getName() + " : " + m);
        }
        System.out.println();

        System.out.println("getYear() : " + date.getYear());
        System.out.println("getMonth() : " + date.getMonth());
        System.out.println("getDate() : " + date.getDate());
        System.out.println("getDay() : " + date.getDay());
        System.out.println("getHours() : " + date.getHours());
        System.out.println("getMinutes() : " + date.getMinutes());
        System.out.println("getSeconds() : " + date.getSeconds());
        System.out.println("getTime() : " + date.getTime());
        System.out.println("getTimezoneOffset() : " + date.getTimezoneOffset());
        System.out.println();
        System.out.println(sdf.format(date));
        date.setYear(100);
        System.out.println(sdf.format(date));
        date.setMonth(0);
        System.out.println(sdf.format(date));
        date.setDate(1);
        System.out.println(sdf.format(date));
        date.setHours(1);
        System.out.println(sdf.format(date));
        date.setMinutes(1);
```

```

        System.out.println(sdf.format(date));
        date.setSeconds(1);
        System.out.println(sdf.format(date));
        date.setTime(1000); // 1초
        System.out.println(sdf.format(date));
        date.setTime(1000 * 60 * 60); // 1시간
        System.out.println(sdf.format(date));
        date.setTime(1000 * 60 * 60 * 24L); // 1일
        System.out.println(sdf.format(date));
        date.setTime(1000 * 60 * 60 * 24L * 365); // 1년
        System.out.println(sdf.format(date));
    }
}

```

결과

```

getDay : public int java.util.Date.getDay()
getMonth : public int java.util.Date.getMonth()
getDate : public int java.util.Date.getDate()
getHours : public int java.util.Date.getHours()
getYear : public int java.util.Date.getYear()
getSeconds : public int java.util.Date.getSeconds()
getTime : public long java.util.Date.getTime()
getTimezoneOffset : public int java.util.Date.getTimezoneOffset()
getMinutes : public int java.util.Date.getMinutes()
getClass : public final native java.lang.Class java.lang.Object.getClass()

setHours : public void java.util.Date.setHours(int)
setSeconds : public void java.util.Date.setSeconds(int)
setTime : public void java.util.Date.setTime(long)
setMonth : public void java.util.Date.setMonth(int)
setMinutes : public void java.util.Date.setMinutes(int)
setYear : public void java.util.Date.setYear(int)
setDate : public void java.util.Date.setDate(int)

getYear() : 121
getMonth() : 9
getDate() : 6
getDay() : 3
getHours() : 9
getMinutes() : 41
getSeconds() : 1
getTime() : 1633480861488
getTimezoneOffset() : -540

2021-10-06(수) 09:41:01
2000-10-06(금) 09:41:01
2000-01-06(목) 09:41:01
2000-01-01(토) 09:41:01
2000-01-01(토) 01:41:01
2000-01-01(토) 01:01:01
2000-01-01(토) 01:01:01
1970-01-01(목) 09:00:01
1970-01-01(목) 10:00:00
1970-01-02(금) 09:00:00
1971-01-01(금) 09:00:00

```

1.3 Method

```
package kr.green.utildate;

import java.text.SimpleDateFormat;
import java.util.Date;
/*
 * public boolean after(Date when)
 * public boolean before(Date when)
 */
public class Ex04_Date {
    @SuppressWarnings("deprecation")
    public static void main(String[] args) {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd(E) hh:mm:ss");
        Date date1 = new Date();
        Date date2 = new Date();
        date2.setDate(1);
        System.out.println("날짜 1 : " + sdf.format(date1));
        System.out.println("날짜 2 : " + sdf.format(date2));

        System.out.println("date1.after(date2) : " + date1.after(date2));
        System.out.println("date1.before(date2) : " + date1.before(date2));
        System.out.println("date2.after(date1) : " + date2.after(date1));
        System.out.println("date2.before(date1) : " + date2.before(date1));

    }
}
```

결과

```
날짜 1 : 2021-10-05(화) 01:01:43
날짜 2 : 2021-10-01(금) 01:01:43
date1.after(date2) : true
date1.before(date2) : false
date2.after(date1) : false
date2.before(date1) : true
```

```
package kr.green.utildate;

import java.text.SimpleDateFormat;
import java.util.Date;
/*
 * public Object clone()
 */
public class Ex05_Date {
    @SuppressWarnings("deprecation")
    public static void main(String[] args) {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd(E) hh:mm:ss");
        Date date1 = new Date();
        Date date2 = (Date) date1.clone();
        Date date3 = date1;

        System.out.println(sdf.format(date1) + " : " + date1.hashCode());
        System.out.println(sdf.format(date2) + " : " + date2.hashCode());
    }
}
```

```

        System.out.println(sdf.format(date3) + " : " + date3.hashCode());
        System.out.println();

        date1.setDate(1);
        System.out.println(sdf.format(date1) + " : " + date1.hashCode());
        System.out.println(sdf.format(date2) + " : " + date2.hashCode());
        System.out.println(sdf.format(date3) + " : " + date3.hashCode());
        System.out.println();

        date2.setDate(1);
        System.out.println(sdf.format(date1) + " : " + date1.hashCode());
        System.out.println(sdf.format(date2) + " : " + date2.hashCode());
        System.out.println(sdf.format(date3) + " : " + date3.hashCode());
        System.out.println();

        System.out.println("compareTo");
        System.out.println(sdf.format(date1));
        System.out.println(sdf.format(date2));
        System.out.println(sdf.format(date3));
        // 0, -1, 1
        System.out.println(date1.compareTo(date2));
        System.out.println(date1.compareTo(date3));
        System.out.println(date3.compareTo(date2));
        System.out.println();

        date1.setDate(2);
        System.out.println(sdf.format(date1));
        System.out.println(sdf.format(date2));
        System.out.println(date1.compareTo(date2));
        System.out.println(date2.compareTo(date1));
        System.out.println();

        // equals
        System.out.println("equals");
        System.out.println(sdf.format(date1));
        System.out.println(sdf.format(date2));
        System.out.println(sdf.format(date3));
        System.out.println(date1.equals(date2));
        System.out.println(date2.equals(date1));
        System.out.println(date1.equals(date3));

    }
}

```

결과

```

2021-10-05(화) 02:16:36 : 1323423375
2021-10-05(화) 02:16:36 : 1323423375
2021-10-05(화) 02:16:36 : 1323423375

2021-10-01(금) 02:16:36 : 977823375
2021-10-05(화) 02:16:36 : 1323423375
2021-10-01(금) 02:16:36 : 977823375

2021-10-01(금) 02:16:36 : 977823375
2021-10-01(금) 02:16:36 : 977823375
2021-10-01(금) 02:16:36 : 977823375

```

```

compareTo
2021-10-01(금) 02:16:36
2021-10-01(금) 02:16:36
2021-10-01(금) 02:16:36
0
0
0

2021-10-02(토) 02:16:36
2021-10-01(금) 02:16:36
1
-1

equals
2021-10-02(토) 02:16:36
2021-10-01(금) 02:16:36
2021-10-02(토) 02:16:36
false
false
true

```

2. java.util.Calendar

2-1. 생성자

`Calendar` 는 추상클래스라서 객체를 직접 생성할 수 없고, 메소드를 통해서 완전히 구현된 클래스의 인스턴스를 얻어야 한다.

```

package kr.green.utilcalendar;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;

public class Ex01_Calendar {
    public static void main(String[] args) {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss");
        Calendar calendar = Calendar.getInstance();
        // Calendar를 Date로 변환
        Date date = calendar.getTime();

        System.out.println(calendar);
        System.out.println(date);
        System.out.println(sdf.format(date));

        Calendar calendar2 = Calendar.getInstance();
        // Date를 Calendar로 변환
        calendar2.setTime(new Date(System.currentTimeMillis()));
        System.out.println(sdf.format(calendar2.getTime()));
    }
}

```

결과

```

java.util.GregorianCalendar[time=1633420120403,areFieldsSet=true,areAllFieldsSet
=true,lenient=true,zone=sun.util.calendar.ZoneInfo[id="Asia/Seoul",offset=324000
00,dstSavings=0,useDaylight=false,transitions=30,lastRule=null],firstDayOfWeek=1
,minimalDaysInFirstWeek=1,ERA=1,YEAR=2021,MONTH=9,WEEK_OF_YEAR=41,WEEK_OF_MONTH=
2,DAY_OF_MONTH=5,DAY_OF_YEAR=278,DAY_OF_WEEK=3,DAY_OF_WEEK_IN_MONTH=1,AM_PM=1,HO
UR=4, HOUR_OF_DAY=16,MINUTE=48,SECOND=40,MILLISECOND=403,ZONE_OFFSET=32400000,DST
_OFFSET=0]
Tue Oct 05 16:48:40 KST 2021
2021-10-05 04:48:40
2021-10-05 04:48:40

```

```

package kr.green.utilcalendar;

import java.util.Calendar;
import java.util.Locale;
import java.util.TimeZone;

/*
Java Calendar.getInstance()
Java Calendar.getInstance(Locale aLocale)
Java Calendar.getInstance(TimeZone zone)
Java Calendar.getInstance(TimeZone zone, Locale aLocale)
*/
public class Ex02_Calendar {
    public static void main(String[] args) {
        // 추상클래스이므로 new 명령어로 객체를 생성할 수 없다.
        // Calendar calendar = new Calendar();
        Calendar calendar1 = Calendar.getInstance();
        viewCalendar(calendar1);

        Locale locale = Locale.getDefault();
        System.out.println("Locale : " + locale);
        Calendar calendar2 = Calendar.getInstance(locale);
        viewCalendar(calendar2);

        TimeZone timeZone = TimeZone.getDefault();
        System.out.println("TimeZone : " + timeZone);
        Calendar calendar3 = Calendar.getInstance(timeZone);
        viewCalendar(calendar3);
        System.out.println();

        Calendar calendar4 = Calendar.getInstance(TimeZone.getTimeZone("GMT"),
        Locale.CANADA);
        System.out.println("TimeZone : " + TimeZone.getTimeZone("GMT"));
        System.out.println("Locale : " + Locale.CANADA);
        viewCalendar(calendar4);

    }

    private static void viewCalendar(Calendar calendar) {
        System.out.print(calendar.get(Calendar.YEAR));
        System.out.print("-" + (calendar.get(Calendar.MONTH)+1));
        System.out.print("-" + calendar.get(Calendar.DATE));
        System.out.print("(" + calendar.get(Calendar.DAY_OF_WEEK) + ") ");
        System.out.print(calendar.get(Calendar.HOUR));
        System.out.print(": " + calendar.get(Calendar.MINUTE));
    }
}

```



```

        System.out.print(": " + calendar.get(Calendar.SECOND));
        System.out.println("." + calendar.get(Calendar.MILLISECOND));
        System.out.println();
    }
}

```

결과

```
2021-10-5(3) 5:6:4.795
```

```
Locale : ko_KR
```

```
2021-10-5(3) 5:6:4.820
```

```
TimeZone :
```

```
sun.util.calendar.ZoneInfo[id="Asia/Seoul",offset=3240000,dstSavings=0,useDayli
ght=false,transitions=30,lastRule=null]
```

```
2021-10-5(3) 5:6:4.821
```

```
TimeZone :
```

```
sun.util.calendar.ZoneInfo[id="GMT",offset=0,dstSavings=0,useDaylight=false,tran
sitions=0,lastRule=null]
```

```
Locale : en_CA
```

```
2021-10-5(3) 8:6:4.821
```

타임존이란?

타임존은 동일한 로컬 시간을 따르는 지역을 의미하며, 주로 해당 국가에 의해 법적으로 지정된다. 보통 국가별로 각자의 고유한 타임존을 사용하고 있으며, 미국이나 캐나다처럼 면적이 넓은 나라인 경우 지역별로 각기 다른 타임존을 사용하기도 한다. (반면 중국은 그 넓은 면적에도 불구하고 하나의 타임존을 이용하고 있는데, 이로 인해 중국 서쪽 지역에서는 오전 10시는 되어야 해를 볼 수 있다)

<https://meetup.toast.com/posts/125> 참조

```

package kr.green.utilcalendar;

import java.util.Calendar;
import java.util.TimeZone;

public class Ex03_Calendar {
    public static void main(String[] args) {
        String[] timeZones = TimeZone.getAvailableIDs();
        Calendar calendar = Calendar.getInstance();
        for(String tz : timeZones) {
            calendar.setTimeZone(TimeZone.getTimeZone(tz));
            System.out.print(calendar.getTimeZone().getDisplayName() + " : ");
            viewCalendar(calendar);
        }
    }

    private static void viewCalendar(Calendar calendar) {
        System.out.print(calendar.get(Calendar.YEAR));
        System.out.print("-" + (calendar.get(Calendar.MONTH)+1));
        System.out.print("-" + calendar.get(Calendar.DATE));
    }
}

```

```

        System.out.print("(" + calendar.get(Calendar.DAY_OF_WEEK) + ") ");
        System.out.print(calendar.get(Calendar.HOUR));
        System.out.print(": " + calendar.get(Calendar.MINUTE));
        System.out.print(": " + calendar.get(Calendar.SECOND));
        System.out.println("." + calendar.get(Calendar.MILLISECOND));
    }
}

```

결과

```

그리니치 표준시 : 2021-10-5(3) 8:25:44.116
그리니치 표준시 : 2021-10-5(3) 8:25:44.116
동아프리카 시간 : 2021-10-5(3) 11:25:44.116
중부 유럽 표준시 : 2021-10-5(3) 9:25:44.116
.
.
.
대한민국 표준시 : 2021-10-5(3) 5:25:44.116
중국 표준시 : 2021-10-5(3) 4:25:44.116
싱가포르 표준시 : 2021-10-5(3) 4:25:44.116
.
.
.
미 태평양 표준시 : 2021-10-5(3) 1:25:44.116
솔로몬 제도 시간 : 2021-10-5(3) 7:25:44.116
인도차이나 반도 시간 : 2021-10-5(3) 3:25:44.116

```

로케일(Locale)이란?

로케일은 사용자 인터페이스에서 사용되는 언어, 지역 설정, 출력 형식 등을 정의하는 문자열입니다. 유닉스 Unix와 리눅스 Linux와 같은 POSIX 기반의 시스템에서는 같은 형식을 공유하고 있습니다. 애플의 맥 OS를 비롯한 Darwin 계열 운영체제들도 POSIX 호환이 되기 때문에 같은 형식의 로케일 문자열을 사용합니다.

로케일과 관련된 환경변수들을 설정함으로써 로케일 설정을 변경할 수 있으며, 이 설정값에 따라서 프로그램이 지원하는 경우 메시지, 시간 형식, 통화 형식 등의 출력이나 동작을 변경할 수 있습니다. 로케일 설정은 특정 언어의 입출력에 영향을 주는 경우가 있어서, 영어 이외의 언어를 사용하는 경우 적절한 값을 지정해야 합니다.

```

package kr.green.utilcalendar;

import java.text.NumberFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Locale;

public class Ex04_Calendar {
    public static void main(String[] args) {
        Locale[] locales = Locale.getAvailableLocales();
        Calendar calendar = null;
        double num = 123456789.123456;
        NumberFormat numberFormat = null;
        SimpleDateFormat dateFormat = new SimpleDateFormat();
        for(Locale locale : locales) {

```

```

        dateFormat = new SimpleDateFormat("MMM", locale);
        numberFormat = NumberFormat.getInstance(locale);
        calendar = Calendar.getInstance(locale);
        System.out.println(locale + "(" + locale.getDisplayCountry() + ")" +
" : "+ numberFormat.format(num) + " : " +
        dateFormat.format(calendar.getTime()));
    }
}
}

```

결과

```

() : 123,456,789.123 : Oct
nn() : 123 456 789,123 : okt
ar_JO(요르단) : تشرين الأول : ١٢٣,٤٥٦,٧٨٩,١٢٣
bg() : 123 456 789,123 : окт
kea() : 123 456 789,123 : Otu
.
.
.
ko_KR(대한민국) : 123,456,789.123 : 10월
el_CY(키프로스) : 123.456.789,123 : ΟΚΤ
mr_IN(인도) : १२३,४५६,७८९.१२३ : ऑक्टो
.
.
.
sd_PK(파키스탄) : آڪٽوبر : ١٢٣,٤٥٦,٧٨٩,١٢٣
mas_TZ(탄자니아) : 123,456,789.123 : Gís
ti() : 123,456,789.123 : ጥቅ
kok() : 123,456,789.123 : ऑक्टोबर
ewo() : 123 456 789,123 : nga
ms_BN(브루나이) : 123.456.789,123 : Okt
ccp_IN(인도) : 123,456,789.123 : ᱠᱚᱰᱚᱨ
br_FR(프랑스) : 123 456 789,123 : Here

```

2-2. field

```

package kr.green.utilcalendar;

import java.lang.reflect.Field;
import java.util.Calendar;

public class Ex05_Calendar {
    public static void main(String[] args) {
        Calendar calendar = Calendar.getInstance();

        Field[] fields = calendar.getClass().getFields();
        for(Field field : fields) {
            System.out.println(field);
        }
        System.out.println();
    }
}

```

결과

```
public static final int java.util.GregorianCalendar.BC
public static final int java.util.GregorianCalendar.AD
public static final int java.util.Calendar.ERA
public static final int java.util.Calendar.YEAR
public static final int java.util.Calendar.MONTH
public static final int java.util.Calendar.WEEK_OF_YEAR
public static final int java.util.Calendar.WEEK_OF_MONTH
public static final int java.util.Calendar.DATE
public static final int java.util.Calendar.DAY_OF_MONTH
public static final int java.util.Calendar.DAY_OF_YEAR
public static final int java.util.Calendar.DAY_OF_WEEK
public static final int java.util.Calendar.DAY_OF_WEEK_IN_MONTH
public static final int java.util.Calendar.AM_PM
public static final int java.util.Calendar.HOUR
public static final int java.util.Calendar.HOUR_OF_DAY
public static final int java.util.Calendar.MINUTE
public static final int java.util.Calendar.SECOND
public static final int java.util.Calendar.MILLISECOND
public static final int java.util.Calendar.ZONE_OFFSET
public static final int java.util.Calendar.DST_OFFSET
public static final int java.util.Calendar.FIELD_COUNT
public static final int java.util.Calendar.SUNDAY
public static final int java.util.Calendar.MONDAY
public static final int java.util.Calendar.TUESDAY
public static final int java.util.Calendar.WEDNESDAY
public static final int java.util.Calendar.THURSDAY
public static final int java.util.Calendar.FRIDAY
public static final int java.util.Calendar.SATURDAY
public static final int java.util.Calendar.JANUARY
public static final int java.util.Calendar.FEBRUARY
public static final int java.util.Calendar.MARCH
public static final int java.util.Calendar.APRIL
public static final int java.util.Calendar.MAY
public static final int java.util.Calendar.JUNE
public static final int java.util.Calendar.JULY
public static final int java.util.Calendar.AUGUST
public static final int java.util.Calendar.SEPTEMBER
public static final int java.util.Calendar.OCTOBER
public static final int java.util.Calendar.NOVEMBER
public static final int java.util.Calendar.DECEMBER
public static final int java.util.Calendar.UNDECIMBER
public static final int java.util.Calendar.AM
public static final int java.util.Calendar.PM
public static final int java.util.Calendar.ALL_STYLES
public static final int java.util.Calendar.SHORT
public static final int java.util.Calendar.LONG
public static final int java.util.Calendar.NARROW_FORMAT
public static final int java.util.Calendar.NARROW_STANDALONE
public static final int java.util.Calendar.SHORT_FORMAT
public static final int java.util.Calendar.LONG_FORMAT
public static final int java.util.Calendar.SHORT_STANDALONE
public static final int java.util.Calendar.LONG_STANDALONE
```

```
package kr.green.utilcalendar;
```

```

import java.util.Calendar;
import java.util.Locale;
import java.util.Map;

// public static final int java.util.Calendar.ALL_STYLES : 0
public class Ex06_Calendar {
    public static void main(String[] args) {
        Calendar calendar = Calendar.getInstance();
        Locale locale = Locale.getDefault();

        Map<String, Integer> map =
calendar.getDisplayNames(Calendar.DAY_OF_WEEK, Calendar.ALL_STYLES, locale);
        for(String key : map.keySet()) {
            System.out.println(key + " : " + map.get(key));
        }
        System.out.println();

        map = calendar.getDisplayNames(Calendar.MONTH, Calendar.ALL_STYLES,
locale);
        for(String key : map.keySet()) {
            System.out.println(key + " : " + map.get(key));
        }
        System.out.println();
    }
}

```

결과

```

금요일 : 6
목요일 : 5
수요일 : 4
월요일 : 2
일요일 : 1
토요일 : 7
화요일 : 3

금 : 6
목 : 5
수 : 4
월 : 2
일 : 1
토 : 7
화 : 3

10월 : 9
11월 : 10
12월 : 11
1월 : 0
2월 : 1
3월 : 2
4월 : 3
5월 : 4
6월 : 5
7월 : 6
8월 : 7
9월 : 8

```

```

package kr.green.utilcalendar;

import java.util.Calendar;
import java.util.Locale;

// public static final int AM_PM = 9;
// public static final int AM = 0;
// public static final int PM = 1;
public class Ex07_Calendar {
    public static void main(String[] args) {
        Locale locale = Locale.getDefault();
        Calendar calendar = Calendar.getInstance(locale);
        System.out.println(calendar.getTime());

        calendar.set(Calendar.AM_PM, Calendar.AM );
        System.out.println(calendar.getTime());

        calendar.set(Calendar.AM_PM, Calendar.PM );
        System.out.println(calendar.getTime());

        System.out.println();
    }
}

```

결과

```

wed Oct 06 10:35:50 KST 2021
wed Oct 06 10:35:50 KST 2021
wed Oct 06 22:35:50 KST 2021

```

```

package kr.green.utilcalendar;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Locale;

// public static final int java.util.Calendar.SUNDAY : 1
// public static final int java.util.Calendar.MONDAY : 2
// public static final int java.util.Calendar.TUESDAY : 3
// public static final int java.util.Calendar.WEDNESDAY : 4
// public static final int java.util.Calendar.THURSDAY : 5
// public static final int java.util.Calendar.FRIDAY : 6
// public static final int java.util.Calendar.SATURDAY : 7
public class Ex08_Calendar {
    public static void main(String[] args) {
        SimpleDateFormat sdf = new SimpleDateFormat("E(EEEE)");
        Locale locale = Locale.getDefault();
        Calendar calendar = Calendar.getInstance(locale);
        System.out.println(calendar.getTime());
        System.out.println(calendar.get(Calendar.DAY_OF_WEEK) + " : " +
sdf.format(calendar.getTime()));

        calendar.set(Calendar.DAY_OF_WEEK, Calendar.SUNDAY);
        System.out.println(calendar.getTime());
        System.out.println(calendar.get(Calendar.DAY_OF_WEEK) + " : " +
sdf.format(calendar.getTime()));
    }
}

```

```

        System.out.println();
    }
}

```

결과

```

wed Oct 06 10:44:33 KST 2021
4 : 수(수요일)
Sun Oct 03 10:44:33 KST 2021
1 : 일(일요일)

```

```

package kr.green.utilcalendar;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Locale;

// public static final int java.util.Calendar.JANUARY : 0
// public static final int java.util.Calendar.FEBRUARY
// public static final int java.util.Calendar.MARCH
// public static final int java.util.Calendar.APRIL
// public static final int java.util.Calendar.MAY
// public static final int java.util.Calendar.JUNE
// public static final int java.util.Calendar.JULY
// public static final int java.util.Calendar.AUGUST
// public static final int java.util.Calendar.SEPTEMBER
// public static final int java.util.Calendar.OCTOBER
// public static final int java.util.Calendar.NOVEMBER
// public static final int java.util.Calendar.DECEMBER : 11
public class Ex09_Calendar {
    public static void main(String[] args) {
        SimpleDateFormat sdf = new SimpleDateFormat("MM(MMM)");
        Locale locale = Locale.getDefault();
        Calendar calendar = Calendar.getInstance(locale);
        System.out.println(calendar.getTime());
        System.out.println(calendar.get(Calendar.MONTH) + " : " +
sdf.format(calendar.getTime()));

        calendar.set(Calendar.MONTH, Calendar.DECEMBER);
        System.out.println(calendar.getTime());
        System.out.println(calendar.get(Calendar.MONTH) + " : " +
sdf.format(calendar.getTime()));
        System.out.println();
    }
}

```

결과

```

wed Oct 06 10:53:11 KST 2021
9 : 10(10월)
Mon Dec 06 10:53:11 KST 2021
11 : 12(12월)

```

```

package kr.green.utilcalendar;

```

```

import java.util.Calendar;
import java.util.Locale;

// public static final int java.util.Calendar.SHORT : 1
// public static final int java.util.Calendar.LONG : 2
// public static final int java.util.Calendar.NARROW_FORMAT : 4
// public static final int java.util.Calendar.NARROW_STANDALONE
// public static final int java.util.Calendar.SHORT_FORMAT
// public static final int java.util.Calendar.LONG_FORMAT
// public static final int java.util.Calendar.SHORT_STANDALONE
// public static final int java.util.Calendar.LONG_STANDALONE
public class Ex10_Calendar {
    public static void main(String[] args) {
        Locale locale = Locale.getDefault();
        Calendar calendar = Calendar.getInstance(locale);

        String displayName = calendar.getDisplayName(Calendar.DAY_OF_WEEK,
Calendar.SHORT, locale);
        System.out.println(displayName);
        displayName = calendar.getDisplayName(Calendar.DAY_OF_WEEK,
Calendar.SHORT_FORMAT, locale);
        System.out.println(displayName);
        displayName = calendar.getDisplayName(Calendar.DAY_OF_WEEK,
Calendar.SHORT_STANDALONE, locale);
        System.out.println(displayName);
        System.out.println();

        displayName = calendar.getDisplayName(Calendar.DAY_OF_WEEK,
Calendar.LONG, locale);
        System.out.println(displayName);
        displayName = calendar.getDisplayName(Calendar.DAY_OF_WEEK,
Calendar.LONG_FORMAT, locale);
        System.out.println(displayName);
        displayName = calendar.getDisplayName(Calendar.DAY_OF_WEEK,
Calendar.LONG_STANDALONE, locale);
        System.out.println(displayName);
        System.out.println();

        displayName = calendar.getDisplayName(Calendar.DAY_OF_WEEK,
Calendar.NARROW_FORMAT, locale);
        System.out.println(displayName);
        displayName = calendar.getDisplayName(Calendar.DAY_OF_WEEK,
Calendar.NARROW_STANDALONE, locale);
        System.out.println(displayName);
        System.out.println();

        displayName = calendar.getDisplayName(Calendar.MONTH, Calendar.SHORT,
locale);
        System.out.println(displayName);
        displayName = calendar.getDisplayName(Calendar.MONTH, Calendar.LONG,
locale);
        System.out.println(displayName);
        displayName = calendar.getDisplayName(Calendar.MONTH,
Calendar.NARROW_FORMAT, locale);
        System.out.println(displayName);
        System.out.println();

        locale = Locale.GERMANY;
    }
}

```



```

        displayName = calendar.getDisplayName(Calendar.MONTH, Calendar.SHORT,
        locale);
        System.out.println(displayName);
        displayName = calendar.getDisplayName(Calendar.MONTH, Calendar.LONG,
        locale);
        System.out.println(displayName);
        displayName = calendar.getDisplayName(Calendar.MONTH,
        Calendar.NARROW_FORMAT, locale);
        System.out.println(displayName);
        System.out.println();
    }
}

```

결과

```

수
수
수

수요일
수요일
수요일

수
수

10월
10월
10월

Okt.
Oktober
0

```

2-3. Getter & Setter

```

package kr.green.utilcalendar;

import java.lang.reflect.Method;
import java.util.Calendar;
import java.util.Locale;

public class Ex11_Calendar {
    public static void main(String[] args) {
        Locale locale = Locale.getDefault();
        Calendar calendar = Calendar.getInstance(locale);

        Method[] methods = calendar.getClass().getMethods();
        for(Method m : methods) {
            if(m.getName().startsWith("get"))
                System.out.println(m.getName() + " : " + m);
        }
        System.out.println();
        for(Method m : methods) {

```

```

        if(m.getName().startsWith("set"))
            System.out.println(m.getName() + " : " + m);
    }
    System.out.println();
}
}

```

결과

```

getGregorianChange : public final java.util.Date
java.util.GregorianCalendar.getGregorianChange()
getCalendarType : public java.lang.String
java.util.GregorianCalendar.getCalendarType()
getTimeZone : public java.util.TimeZone
java.util.GregorianCalendar.getTimeZone()
getWeekYear : public int java.util.GregorianCalendar.getWeekYear()
getWeeksInWeekYear : public int java.util.GregorianCalendar.getWeeksInWeekYear()
getMinimum : public int java.util.GregorianCalendar.getMinimum(int)
getMaximum : public int java.util.GregorianCalendar.getMaximum(int)
getGreatestMinimum : public int
java.util.GregorianCalendar.getGreatestMinimum(int)
getLeastMaximum : public int java.util.GregorianCalendar.getLeastMaximum(int)
getActualMinimum : public int java.util.GregorianCalendar.getActualMinimum(int)
getActualMaximum : public int java.util.GregorianCalendar.getActualMaximum(int)
get : public int java.util.Calendar.get(int)
getInstance : public static java.util.Calendar
java.util.Calendar.getInstance(java.util.TimeZone)
getInstance : public static java.util.Calendar java.util.Calendar.getInstance()
getInstance : public static java.util.Calendar
java.util.Calendar.getInstance(java.util.TimeZone,java.util.Locale)
getInstance : public static java.util.Calendar
java.util.Calendar.getInstance(java.util.Locale)
getDisplayName : public java.lang.String
java.util.Calendar.getDisplayName(int,int,java.util.Locale)
getAvailableLocales : public static synchronized java.util.Locale[]
java.util.Calendar.getAvailableLocales()
getTime : public final java.util.Date java.util.Calendar.getTime()
getTimeInMillis : public long java.util.Calendar.getTimeInMillis()
getDisplayNames : public java.util.Map
java.util.Calendar.getDisplayNames(int,int,java.util.Locale)
getAvailableCalendarTypes : public static java.util.Set
java.util.Calendar.getAvailableCalendarTypes()
getFirstDayOfWeek : public int java.util.Calendar.getFirstDayOfWeek()
getMinimalDaysInFirstWeek : public int
java.util.Calendar.getMinimalDaysInFirstWeek()
getClass : public final native java.lang.Class java.lang.Object.getClass()

setGregorianChange : public void
java.util.GregorianCalendar.setGregorianChange(java.util.Date)
setTimeZone : public void
java.util.GregorianCalendar.setTimeZone(java.util.TimeZone)
setWeekDate : public void java.util.GregorianCalendar.setWeekDate(int,int,int)
set : public final void java.util.Calendar.set(int,int,int,int,int)
set : public final void java.util.Calendar.set(int,int,int,int,int,int)
set : public final void java.util.Calendar.set(int,int,int)
set : public void java.util.Calendar.set(int,int)
setTime : public final void java.util.Calendar.setTime(java.util.Date)

```

```
setTimeInMillis : public void java.util.Calendar.setTimeInMillis(long)
setLenient : public void java.util.Calendar.setLenient(boolean)
setFirstDayOfWeek : public void java.util.Calendar.setFirstDayOfWeek(int)
setMinimalDaysInFirstWeek : public void
java.util.Calendar.setMinimalDaysInFirstWeek(int)
```

```
package kr.green.utilcalendar;

import java.util.Calendar;
import java.util.Locale;

//get : public int java.util.Calendar.get(int)
public class Ex12_Calendar {
    public static void main(String[] args) {
        Locale locale = Locale.getDefault();
        Calendar calendar = Calendar.getInstance(locale);

        System.out.println("년 : " + calendar.get(Calendar.YEAR));
        System.out.println("월 : " + (calendar.get(Calendar.MONTH) + 1));
        System.out.println("일 : " + calendar.get(Calendar.DATE));
        System.out.println("요일 : " + calendar.get(Calendar.DAY_OF_WEEK));
        System.out.println("시 : " + calendar.get(Calendar.HOUR));
        System.out.println("분 : " + calendar.get(Calendar.MINUTE));
        System.out.println("초 : " + calendar.get(Calendar.SECOND));
        System.out.println("1/1000초 : " + calendar.get(Calendar.MILLISECOND));
        System.out.println();

        System.out.println("월의 일 : " + calendar.get(Calendar.DAY_OF_MONTH));
        System.out.println("년의 일 : " + calendar.get(Calendar.DAY_OF_YEAR));
        System.out.println("시 : " + calendar.get(Calendar.HOUR_OF_DAY));
        String displayName = calendar.getDisplayName(Calendar.AM_PM,
        Calendar.SHORT, locale);
        System.out.println("오전/오후 : " + calendar.get(Calendar.AM_PM) + ", " +
        displayName);
        System.out.println(displayName);
        System.out.println();

        calendar.set(Calendar.HOUR_OF_DAY, 18);
        displayName = calendar.getDisplayName(Calendar.AM_PM, Calendar.SHORT,
        locale);
        System.out.println("오전/오후 : " + calendar.get(Calendar.AM_PM) + ", " +
        displayName);
        System.out.println("12 시 : " + calendar.get(Calendar.HOUR));
        System.out.println("24 시 : " + calendar.get(Calendar.HOUR_OF_DAY));
        System.out.println();
    }
}
```

결과

```
년 : 2021
월 : 10
일 : 6
요일 : 4
시 : 11
분 : 34
```

초 : 40
1/1000초 : 703

월의 일 : 6
년의 일: 279
시 : 11
오전/오후 : 0, 오전
오전

오전/오후 : 1, 오후
12 시 : 6
24 시 : 18

```
package kr.green.utilcalendar;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Locale;

// public final void java.util.Calendar.set(int,int,int,int,int)
// public final void java.util.Calendar.set(int,int,int,int,int,int)
// public final void java.util.Calendar.set(int,int,int)
// public void java.util.Calendar.set(int,int)
public class Ex13_Calendar {
    public static void main(String[] args) {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MMM-dd(EEEE)
hh:mm:ss.S");
        Locale locale = Locale.getDefault();
        Calendar calendar = Calendar.getInstance(locale);
        System.out.println(sdf.format(calendar.getTime()));

        calendar.set(2000, 1, 1);
        System.out.println(sdf.format(calendar.getTime()));

        calendar.set(2002, 4, 3, 11, 23);
        System.out.println(sdf.format(calendar.getTime()));

        calendar.set(2002, 0, 1, 2, 3, 4);
        System.out.println(sdf.format(calendar.getTime()));

        calendar.set(Calendar.DATE, 33);
        System.out.println(sdf.format(calendar.getTime()));

        calendar.set(Calendar.MONTH, 13);
        System.out.println(sdf.format(calendar.getTime()));
    }
}
```

결과

2021-10월-06(수요일) 11:40:15.482
2000-2월-01(화요일) 11:40:15.482
2002-5월-03(금요일) 11:23:15.482
2002-1월-01(화요일) 02:03:04.482
2002-2월-02(토요일) 02:03:04.482
2003-2월-02(일요일) 02:03:04.482

```

package kr.green.utilcalendar;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Locale;

// public abstract void add(int field, int amount)
public class Ex14_Calendar {
    public static void main(String[] args) {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd(EEEE)
hh:mm:ss.S");
        Locale locale = Locale.getDefault();
        Calendar calendar = Calendar.getInstance(locale);
        System.out.println("오늘 : " + sdf.format(calendar.getTime()));

        calendar.add(Calendar.DATE, 20);
        System.out.println("20일 후 : " + sdf.format(calendar.getTime()));

        calendar.add(Calendar.MONTH, -2);
        System.out.println("2개월 전 : " + sdf.format(calendar.getTime()));

        calendar.add(Calendar.YEAR, -5);
        System.out.println("5년 전 : " + sdf.format(calendar.getTime()));
    }
}

```

결과

```

오늘 : 2021-10월-06(수요일) 11:51:13.238
20일 후 : 2021-10월-26(화요일) 11:51:13.238
2개월 전 : 2021-8월-26(목요일) 11:51:13.238
5년 전 : 2016-8월-26(금요일) 11:51:13.238

```

```

package kr.green.utilcalendar;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Locale;

// public abstract void roll(int field, boolean up)
// public void roll(int field, int amount)
public class Ex15_Calendar {
    public static void main(String[] args) {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd(EEEE)");
        Locale locale = Locale.getDefault();
        Calendar calendar = Calendar.getInstance(locale);
        System.out.println("오늘 : " + sdf.format(calendar.getTime()));

        calendar.roll(Calendar.MONTH, true);
        System.out.println("1개월 후 : " + sdf.format(calendar.getTime()));
        calendar.roll(Calendar.MONTH, false);
        calendar.roll(Calendar.MONTH, false);
        System.out.println("1개월 전 : " + sdf.format(calendar.getTime()));
        calendar.roll(Calendar.MONTH, 2);
        System.out.println("1개월 후 : " + sdf.format(calendar.getTime()));
        calendar.roll(Calendar.MONTH, -6);
    }
}

```

```

        System.out.println("6개월 전 : " + sdf.format(calendar.getTime()));
    }
}

```

결과

```

오늘 : 2021-10월-06(수요일)
1개월 후 : 2021-11월-06(토요일)
1개월 전 : 2021-9월-06(월요일)
1개월 후 : 2021-11월-06(토요일)
6개월 전 : 2021-5월-06(목요일)

```

```

package kr.green.utilcalendar;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Locale;

// public boolean after(Object when)
// public boolean before(Object when)
public class Ex16_Calendar {
    public static void main(String[] args) {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MMM-dd(EEEE)");
        Locale locale = Locale.getDefault();
        Calendar calendar1 = Calendar.getInstance(locale);
        Calendar calendar2 = Calendar.getInstance(locale);
        calendar2.roll(Calendar.DATE, -5);

        System.out.println(sdf.format(calendar1.getTime()));
        System.out.println(sdf.format(calendar2.getTime()));

        if(calendar2.after(calendar1)) System.out.println("오늘보다 이후입니다.");
        if(calendar2.before(calendar1)) System.out.println("오늘보다 이전입니다.");

        calendar2.roll(Calendar.DATE, 5);
        System.out.println(sdf.format(calendar1.getTime()));
        System.out.println(sdf.format(calendar2.getTime()));

        if(calendar2.after(calendar1)) System.out.println("오늘보다 이후입니다.");
        if(calendar2.before(calendar1)) System.out.println("오늘보다 이전입니다.");

        calendar2.roll(Calendar.DATE, 1);
        System.out.println(sdf.format(calendar1.getTime()));
        System.out.println(sdf.format(calendar2.getTime()));

        if(calendar2.after(calendar1)) System.out.println("오늘보다 이후입니다.");
        if(calendar2.before(calendar1)) System.out.println("오늘보다 이전입니다.");
    }
}

```

결과

2021-10월-06(수요일)
2021-10월-01(금요일)
오늘보다 이전입니다.
2021-10월-06(수요일)
2021-10월-06(수요일)
2021-10월-06(수요일)
2021-10월-07(목요일)
오늘보다 이후입니다.

```
package kr.green.utilcalendar;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Locale;

// public final void clear()
// public final void clear(int field)
public class Ex17_Calendar {
    public static void main(String[] args) {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MMM-dd(EEEE)");
        Locale locale = Locale.getDefault();
        Calendar calendar = Calendar.getInstance(locale);
        System.out.println(sdf.format(calendar.getTime()));

        calendar.clear(Calendar.MONTH);
        System.out.println(sdf.format(calendar.getTime()));

        calendar.clear();
        System.out.println(sdf.format(calendar.getTime()));
    }
}
```

결과

2021-10월-06(수요일)
2021-1월-06(수요일)
1970-1월-01(목요일)

```
package kr.green.utilcalendar;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Locale;

// public int getActualMaximum(int field)
// public int getActualMinimum(int field)
public class Ex18_Calendar {
    @SuppressWarnings("static-access")
    public static void main(String[] args) {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MMM-dd(EEEE)");
        Locale locale = Locale.getDefault();
        Calendar calendar = Calendar.getInstance(locale);
        System.out.println(sdf.format(calendar.getTime()));

        int maxYear = calendar.getActualMaximum(Calendar.YEAR);
    }
}
```

```

        int minYear = calendar.getActualMinimum(Calendar.YEAR);
        System.out.println("year:" + minYear + " ~ " + maxYear);

        int maxMonth = calendar.getActualMaximum(Calendar.MONTH);
        int minMonth = calendar.getActualMinimum(Calendar.MONTH);
        System.out.println("month:" + minMonth + " ~ " + maxMonth);

        int maxDate = calendar.getActualMaximum(Calendar.DATE);
        int minDate = calendar.getActualMinimum(Calendar.DATE);
        System.out.println("date:" + minDate + " ~ " + maxDate);

        calendar.set(Calendar.MONTH, 1);
        System.out.println(sdf.format(calendar.getTime()));

        maxDate = calendar.getActualMaximum(Calendar.DATE);
        minDate = calendar.getActualMinimum(Calendar.DATE);
        System.out.println("date:" + minDate + " ~ " + maxDate);

        calendar.set(Calendar.YEAR, 2000);
        System.out.println(sdf.format(calendar.getTime()));

        maxDate = calendar.getActualMaximum(Calendar.DATE);
        minDate = calendar.getActualMinimum(Calendar.DATE);
        System.out.println("date:" + minDate + " ~ " + maxDate);
    }
}

```

결과

```

2021-10월-06(수요일)
year:1 ~ 292278993
month:0 ~ 11
date:1 ~ 31
2021-2월-06(토요일)
date:1 ~ 28
2000-2월-06(일요일)
date:1 ~ 29

```

```

package kr.green.utilcalendar;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Locale;

// public int getMaximum(int field)
// public int getMinimum(int field)
public class Ex19_Calendar {
    @SuppressWarnings("static-access")
    public static void main(String[] args) {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MMM-dd(EEEE)");
        Locale locale = Locale.getDefault();
        Calendar calendar = Calendar.getInstance(locale);
        System.out.println(sdf.format(calendar.getTime()));

        int maxYear = calendar.getMaximum(Calendar.YEAR);
        int minYear = calendar.getMinimum(Calendar.YEAR);
    }
}

```



```

        System.out.println("year:" + minYear + " ~ " + maxYear);

        int maxMonth = calendar.getMaximum(Calendar.MONTH);
        int minMonth = calendar.getMinimum(Calendar.MONTH);
        System.out.println("month:" + minMonth + " ~ " + maxMonth);

        int maxDate = calendar.getMaximum(Calendar.DATE);
        int minDate = calendar.getMinimum(Calendar.DATE);
        System.out.println("date:" + minDate + " ~ " + maxDate);

        calendar.set(Calendar.MONTH, 1);
        System.out.println(sdf.format(calendar.getTime()));

        maxDate = calendar.getMaximum(Calendar.DATE);
        minDate = calendar.getMinimum(Calendar.DATE);
        System.out.println("date:" + minDate + " ~ " + maxDate);

        calendar.set(Calendar.YEAR, 2000);
        System.out.println(sdf.format(calendar.getTime()));

        maxDate = calendar.getMaximum(Calendar.DATE);
        minDate = calendar.getMinimum(Calendar.DATE);
        System.out.println("date:" + minDate + " ~ " + maxDate);
    }
}

```

결과

```

2021-10월-06(수요일)
year:1 ~ 292278994
month:0 ~ 11
date:1 ~ 31
2021-2월-06(토요일)
date:1 ~ 31
2000-2월-06(일요일)
date:1 ~ 31

```

```

package kr.green.utilcalendar;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Locale;

// public int getFirstDayOfWeek()
public class Ex20_Calendar {
    @SuppressWarnings("static-access")
    public static void main(String[] args) {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MMM-dd(EEEE)");
        Locale locale = Locale.getDefault();
        Calendar calendar = Calendar.getInstance(locale);
        System.out.println(sdf.format(calendar.getTime()));

        System.out.println("First day is : " + calendar.getFirstDayOfWeek());
        calendar.set(Calendar.DAY_OF_WEEK, calendar.getFirstDayOfWeek());
        System.out.println(calendar.getDisplayName(Calendar.DAY_OF_WEEK,
            calendar.LONG, locale));
    }
}

```

```

        locale = Locale.FRANCE;
        calendar = Calendar.getInstance(locale);
        System.out.println("First day is : " + calendar.getFirstDayOfWeek());
        calendar.set(Calendar.DAY_OF_WEEK, calendar.getFirstDayOfWeek());
        System.out.println(calendar.getDisplayName(Calendar.DAY_OF_WEEK,
        Calendar.LONG, Locale.KOREA));
    }
}

```

결과

```

2021-10월-06(수요일)
First day is : 1
일요일
First day is : 2
월요일

```

Java8 날짜-시간 API

Java 8은 ISO-8601 날짜-시간 표준을 기반으로 하는 새로운 날짜-시간 API를 도입했습니다.

날짜-시간 API는 ISO-8601에 정의된 달력 시스템을 기본 달력으로 사용합니다. 이 달력은 그레고리력 체계를 기반으로 합니다.

Date-Time API의 LocalDateTime, ZonedDateTime 및 OffsetDateTime은 모두 ISO 달력 시스템을 사용합니다.

날짜-시간 API는 로케일 데이터에 대해 유니코드 CLDR(Common Locale Data Repository)을 사용합니다.

Date-Time API는 시간대에 대한 정보를 위해 TZDB(Time-Zone Database)를 사용합니다.

날짜-시간 API 설계 원칙

API의 메소드가 잘 정의되어 있습니다.

Date-Time API는 코드를 읽기 쉽게 만드는 유창한 인터페이스를 제공합니다. 메서드 호출은 함께 연결할 수 있습니다. 예를 들어 다음 코드를 사용하여 새 날짜를 만들 수 있습니다.

```

LocalDate aDay =
    LocalDate.now().with(TemporalAdjusters.lastDayOfMonth()).minusDays(2);

```

Java Date-Time API는 대부분 변경할 수 없는 클래스로 구성됩니다. 개체를 만든 후에는 수정할 수 없습니다. 값을 변경하려면 원본의 수정된 복사본으로 새 개체를 만들어야 합니다.

Date-Time API는 스레드로부터 안전합니다.

날짜 나 시간 개체를 만드는 데 사용되는 방법으로 시작하는 `of`, `from` 또는 `with` 오히려 생성자보다. 방법이 없습니다 `set`. 예를 들어:

```

LocalDate dateOfBirth = LocalDate.of(2014, Month.MAY, 14);

```

Date-Time API는 가능한 모든 곳에서 확장할 수 있습니다.

Java Date-Time API에는 기계 기반 시간과 달력 기반 인간 시간을 처리하는 별도의 클래스 세트가 있습니다.

날짜-시간 패키지

Java Date-Time API는 기본 패키지 `java.time` 및 4개의 하위 패키지로 구성됩니다 .

- `java.time` 날짜와 시간을 나타내는 API의 핵심을 포함합니다. 여기에는 날짜, 시간, 날짜 및 시간 결합, 시간대, 순간, 기간 및 시계에 대한 클래스가 포함됩니다. `LocalDate`, `LocalTime`, `LocalDateTime`, `ZonedDateTime`, `Period`, `Duration`, and `Instant` 클래스는 이 패키지에 있습니다. 이 패키지의 클래스는 ISO-8601 표준을 기반으로 합니다.
- `java.time.chrono` 기본 ISO-8601 이외의 달력 시스템을 나타내는 API를 포함합니다. 예를 들어 히즈라 달력, 태국 불교 달력 등
- `java.time.format` 날짜 및 시간을 형식화하고 구문 분석하기 위한 클래스가 포함되어 있습니다.
- `java.time.temporal` 날짜 및 시간 클래스 간의 상호 운용, 쿼리 및 조정을 허용하는 API가 포함되어 있습니다. 예를 들어, `TemporalField` 및 `ChronoField`, `TemporalUnit` 및 `ChronoUnit`이 있습니다.
- `java.time.zone` 표준 시간대, 표준 시간대 오프셋 및 표준 시간대 규칙을 지원하는 클래스를 포함합니다. 예: `ZonedDateTime` 및 `ZoneId` 또는 `ZoneOffset`.