

■ Calendar

- Calendar는 추상클래스이므로 직접 객체를 생성할 수 없음
- 국가 및 지역에 따라서 사용되는 달력의 종류가 다르므로
getInstance() 메소드를 통해 생성하도록 설계 (코드 수정 최소화)

```
switch (caltype) {  
case "buddhist":  
cal = new BuddhistCalendar(zone, aLocale);  
break;  
case "japanese":  
cal = new JapaneseImperialCalendar(zone, aLocale);  
break;  
case "gregory":  
cal = new GregorianCalendar(zone, aLocale);  
break;  
}
```

```
if (aLocale.getLanguage() == "th" && aLocale.getCountry() == "TH") {  
cal = new BuddhistCalendar(zone, aLocale);  
} else if (aLocale.getVariant() == "JP" && aLocale.getLanguage() == "ja"  
    && aLocale.getCountry() == "JP") {  
cal = new JapaneseImperialCalendar(zone, aLocale);  
} else {  
cal = new GregorianCalendar(zone, aLocale);  
}
```

■ Calendar 기본 사용

● calendar.CalendarExam1.java (1 / 3)

```
public class CalendarExam1 {  
    public static void main(String[] args) {  
        Calendar cal = Calendar.getInstance();  
  
        // 년  
        int year = cal.get(Calendar.YEAR);  
        System.out.println("현재 년도 : " + year);  
  
        // 월 (0 ~ 11)  
        int month = cal.get(Calendar.MONTH);  
        System.out.println("현재 월 : " + month);  
  
        // 해당 월의 일자 (1 ~ 해당 월의 마지막 날)  
        int date = cal.get(Calendar.DAY_OF_MONTH);  
        int date2 = cal.get(Calendar.DATE); // DAY_OF_MONTH와 동일  
        System.out.println("현재 일 : " + date + ", " + date2);  
    }  
}
```

■ Calendar 기본 사용

● calendar.CalendarExam1.java (2 / 3)

```
// 요일 (일:1 ~ 토:7)
```

```
int dayOfWeek = cal.get(Calendar.DAY_OF_WEEK);
```

```
System.out.println("요일 : " + dayOfWeek);
```

```
// 현재 일이 해당 월의 몇번째 주인지 출력
```

```
int dayOfWeekInMonth = cal.get(Calendar.DAY_OF_WEEK_IN_MONTH);
```

```
System.out.println("현재 월의 몇번째 주 : " + dayOfWeekInMonth);
```

```
// 1년 중 몇번째 날짜인지 출력
```

```
int dayOfYear = cal.get(Calendar.DAY_OF_YEAR);
```

```
System.out.println("현재 년도의 몇번째 일 : " + dayOfYear);
```

■ Calendar 기본 사용

● calendar.CalendarExam1.java (3 / 3)

```
// 시, 분, 초 출력
int hour = cal.get(Calendar.HOUR_OF_DAY); // 0 ~ 23
int hour2 = cal.get(Calendar.HOUR); // 0 ~ 11
int ampm = cal.get(Calendar.AM_PM); // 오전:0, 오후:1
int min = cal.get(Calendar.MINUTE);
int sec = cal.get(Calendar.SECOND);
System.out.println(hour + "시 " + hour2 + "시 " + ampm + "(오전오후) " + min + "분
" + sec + "초");

int lastDay = cal.getActualMaximum(Calendar.DAY_OF_MONTH);
System.out.println("현재 월의 마지막 일 : " + lastDay);
}
}
```

■ Calendar 기본 사용

● 실행결과

현재 년도 : 2016

현재 월 : 3

현재 일 : 11, 11

요일 : 2

현재 월의 몇번째 주 : 2

현재 년도의 몇번째 일 : 102

14시 2시 1 (오전오후) 58분 29초

현재 월의 마지막 일 : 30

■ Calendar 날짜 지정 및 차이 구하기

● calendar.CalendarExam2.java (1 / 2)

```
public class CalendarExam2 {  
    public static void main(String[] args) {  
        Calendar cal1 = Calendar.getInstance();  
        Calendar cal2 = Calendar.getInstance();  
  
        // 2007년 3월 27일로 날짜 지정  
        cal1.set(Calendar.YEAR, 2007);  
        cal1.set(Calendar.MONTH, 2);  
        cal1.set(Calendar.DATE, 27);  
        cal1.set(2007, 2, 27); // 위 3줄과 동일한 기능
```

■ Calendar 날짜 지정 및 차이 구하기

● calendar.CalendarExam2.java (2 / 2)

```
// 날짜 차이 구하기
```

```
long diffSec = cal2.getTimeInMillis() - cal1.getTimeInMillis();
```

```
diffSec = diffSec / 1000; // 밀리초 -> 초
```

```
System.out.println(diffSec + "초");
```

```
int diffHour = (int) (diffSec / (60 * 60));
```

```
System.out.println(diffHour + "시간");
```

```
int diffDay = (int) (diffSec / (60 * 60 * 24));
```

```
System.out.println(diffDay + "일");
```

```
}
```

```
}
```

285379200초

79272시간

3303일

■ Calendar 월의 마지막 일자 구하기, 두자리 형식 출력

● calendar.CalendarExam3.java (1 / 2)

```
public class CalendarExam3 {  
    public static void main(String[] args) {  
        Calendar cal = Calendar.getInstance();  
        int year = cal.get(Calendar.YEAR);  
        int month = cal.get(Calendar.MONTH) + 1; // 0 ~ 11월  
        int day = cal.get(Calendar.DAY_OF_MONTH);  
  
        System.out.println(year + "" + month + "" + day);  
  
        cal.set(Calendar.MONTH, 3); // 4월로 지정  
        cal.set(Calendar.DAY_OF_MONTH, 1); // 1일로 지정  
  
        cal.add(Calendar.DAY_OF_MONTH, -1); // -1일 => 3월의 마지막 날짜로 이동
```


■ Calendar 월의 마지막 일자 구하기, 두자리 형식 출력

● calendar.CalendarExam3.java (2 / 2)

```
// 날짜 출력
```

```
year = cal.get(Calendar.YEAR);
```

```
month = cal.get(Calendar.MONTH) + 1; // 0 ~ 11월
```

```
day = cal.get(Calendar.DAY_OF_MONTH);
```

```
String fullDate = "" + year + (month < 10 ? "0" + month : month) +  
    (day < 10 ? "0" + day : day);
```

```
System.out.println(fullDate);
```

```
}
```

```
}
```

■ Calendar 월의 마지막 일자 구하기, 두자리 형식 출력

● 실행결과

```
2016411
```

```
20160331
```

■ Calendar 구현

● calendar.CalendarExam4.java (1 / 2)

```
public class CalendarExam4 {  
    public static void main(String[] args) {  
        Calendar cal = Calendar.getInstance();  
  
        // 현재 월의 1일로 설정  
        cal.set(Calendar.DATE, 1);  
  
        int startDayOfWeek = cal.get(Calendar.DAY_OF_WEEK);  
  
        System.out.println(" SU MO TU WE TH FR SA");  
        for(int i = 1; i < startDayOfWeek; i++) {  
            System.out.print("  ");  
        }  
    }  
}
```

■ Calendar 구현

● calendar.CalendarExam4.java (2 / 2)

```
int endDate = cal.getActualMaximum(Calendar.DATE);
for(int i = 1, n = startDayOfWeek; i <= endDate; i++, n++) {
    System.out.print((i < 10 ? " " + i : " " + i));
    if(n % 7 == 0) {
        System.out.println();
    }
}
}
```

■ Calendar 구현

● 실행결과

SU	MO	TU	WE	TH	FR	SA
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30