Chapter 10

날짜와 시간 & 형식화

date, time and formatting

[연습문제]

[10-1] Calendar클래스와 SimpleDateFormat클래스를 이용해서 2010년의 매월 두 번째 일요일의 날짜를 출력하시오.

[실행결과]

```
2010-01-10은 2번째 일요일입니다.
2010-02-14은 2번째 일요일입니다.
2010-03-14은 2번째 일요일입니다.
2010-04-11은 2번째 일요일입니다.
2010-05-09은 2번째 일요일입니다.
2010-06-13은 2번째 일요일입니다.
2010-07-11은 2번째 일요일입니다.
2010-08-08은 2번째 일요일입니다.
2010-09-12은 2번째 일요일입니다.
2010-10-10은 2번째 일요일입니다.
2010-11-14은 2번째 일요일입니다.
2010-12-12은 2번째 일요일입니다.
```

```
public static void main(String[] args) {
    Calendar cal = Calendar.getInstance();

    cal.set(2010, 0, 1);

    for (int i = 0; i < 12; i++) {
        int weekday = cal.get(Calendar.DAY_OF_WEEK); // 1일의 요일
        int secondSunday = (weekday == 1) ? 8 : 16 - weekday;

        cal.set(Calendar.DAY_OF_MONTH, secondSunday);

        Date d = cal.getTime();
        SimpleDateFormat sdf = new SimpleDateFormat("y-M-d은 F번째 E요일 입니다.");
        System.out.println(sdf.format(d));

        cal.add(Calendar.MONTH, 1);
        cal.set(Calendar.DAY_OF_MONTH, 1);
    } // for
}
```

[10-2] 어떤 회사의 월급날이 매월 21일이다. 두 날짜 사이에 월급날이 몇 번있는지 계산해서 반환하는 메서드를 작성하고 테스트 하시오.

[연습문제]/ch10/Exercise10_2.java

```
import java.util.*;
import java.text.*;

class Exercise10_2 {
    static int paycheckCount(Calendar from, Calendar to) {
        /*
          (1) 아래의 로직에 맞게 코드를 작성하시오.
          1. from 또는 to가 null이면 0을 반환한다.
          2. from와 to가 같고 날짜가 21일이면 1을 반환한다.
          3. to와 from이 몇 개월 차이인지 계산해서 변수 monDiff에 담는다.
          4. monDiff가 음수이면 0을 반환한다.
          5. 만일 from의 일(DAY_OF_MONTH)이 21일이거나 이전이고
             to의 일(DAY_OF_MONTH)이 21일이거나 이후이면 monDiff의 값을 1 증가시킨다.
          6. 만일 from의 일(DAY_OF_MONTH)이 21일 이후고
               to의 일(DAY_OF_MONTH)이 21일 이전이면 monDiff의 값을 1 감소시킨다.
        */

        return monDiff;
    }

    static void printResult(Calendar from, Calendar to) {
        Date fromDate = from.getTime();
        Date toDate   = to.getTime();

        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
```

```
        System.out.print(sdf.format(fromDate)+" ~ "
                         +sdf.format(toDate)+":");
        System.out.println(paycheckCount(from, to));
    }

    public static void main(String[] args) {
        Calendar fromCal = Calendar.getInstance();
        Calendar toCal   = Calendar.getInstance();

        fromCal.set(2010,0,1);
        toCal.set(2010,0,1);
        printResult(fromCal, toCal);

        fromCal.set(2010,0,21);
        toCal.set(2010,0,21);
        printResult(fromCal, toCal);

        fromCal.set(2010,0,1);
        toCal.set(2010,2,1);
        printResult(fromCal, toCal);

        fromCal.set(2010,0,1);
        toCal.set(2010,2,23);
        printResult(fromCal, toCal);

        fromCal.set(2010,0,23);
        toCal.set(2010,2,21);
        printResult(fromCal, toCal);

        fromCal.set(2011,0,22);
        toCal.set(2010,2,21);
        printResult(fromCal, toCal);

    }
}

static int payCheckCount(Calendar from, Calendar to) {
    if (from == null || to == null) {return 0;}

    if (from.equals(to) && from.get(Calendar.DAY_OF_MONTH) == 21)
        {return 1;}

    int fromYear = from.get(Calendar.YEAR);
    int fromMon = from.get(Calendar.MONTH);
    int fromDay = from.get(Calendar.DAY_OF_MONTH);

    int toYear = to.get(Calendar.YEAR);
    int toMon = to.get(Calendar.MONTH);
    int toDay = to.get(Calendar.DAY_OF_MONTH);

    // 몇 개월 차이
    int monDiff = (toYear * 12 + toMon) - (fromYear * 12 + fromMon);

    if (monDiff < 0) {return 0;}

    if (fromDay <= 21 && toDay >= 21) {monDiff++;}

    if (fromDay > 21 && toDay < 21) {monDiff--;}

    return monDiff;
}
```

[실행결과]

```
2010-01-01 ~ 2010-01-01:0
2010-01-21 ~ 2010-01-21:1
2010-01-01 ~ 2010-03-01:2
2010-01-01 ~ 2010-03-23:3
2010-01-23 ~ 2010-03-21:2
2011-01-22 ~ 2010-03-21:0
```

[10-3] 문자열 "123,456,789.5"를 소수점 첫 번째 자리에서 반올림하고, 그 값을 만 단위마다 콤마(,)로 구분해서 출력하시오.

```
public static void main(String[] args) {
    String data = "123,456,789.5";

    DecimalFormat df = new DecimalFormat("#,###.#"); // 변환할 문자열의 형식을 지정
    DecimalFormat df2 = new DecimalFormat("#,####"); // 만단위

    try {
        Number num = df.parse(data);
        double d = num.doubleValue(); // 1.234567895E8

        System.out.println("data:" + data);
        System.out.println("반올림: " + Math.round(d));
        System.out.println("만단위: " + df2.format(d));
    } catch (Exception e) {

    }
}
```

[실행결과]

```
data:123,456,789.5
반올림:123456790
만단위:1,2345,6790
```

```
public static void main(String[] args) {
    String pattern = "yyyy/MM/dd";
    String pattern2 = "입력하신 날짜는 E요일입니다.";

    SimpleDateFormat df = new SimpleDateFormat(pattern); // input parsing
    SimpleDateFormat df2 = new SimpleDateFormat(pattern2); // output format
    Scanner s = new Scanner(System.in);

    Date inDate = null;

    do {
        System.out.println("날짜를" + pattern + "의 형태로 입력해주세요");

        try {
            System.out.println(">>");
            inDate = df.parse(s.nextLine());
            break;
        } catch (Exception e) {}

    } while (true);

    System.out.println(df2.format(inDate));

}
```

[10-4] 화면으로부터 날짜를 "2007/05/11"의 형태로 입력받아서 무슨 요일인지 출력하는 프로그램을 작성하시오.
단, 입력된 날짜의 형식이 잘못된 경우 메세지를 보여주고 다시 입력받아야 한다.

[실행결과]

```
날짜를 yyyy/MM/dd의 형태로 입력해주세요.(입력예:2007/05/11)
>>2009-12-12
날짜를 yyyy/MM/dd의 형태로 입력해주세요.(입력예:2007/05/11)
>>2009/12/12
입력하신 날짜는 토요일입니다.
```

[10-5] 다음과 같이 정의된 메서드를 작성하고 테스트하시오.

메서드명 : getDayDiff
기　　능 : yyyymmdd형식의 두 문자열을 넘겨받으면 두 날짜의 차이를 일(day)단위로 반환한다.
　　단, 첫 번째 날짜 빼기 두 번째 날짜의 결과를 반환한다.
　　만일 주어진 문자열이 유효하지 않으면 0을 반환한다.
반환타입 : int
매개변수 : String yyyymmdd1 - 시작날짜
　　String yyyymmdd2 - 끝 날짜

[연습문제]/ch10/Exercise10_5.java

```
import java.util.*;

class Exercise10_5 {
   /*
      (1) getDayDiff메서드를 작성하시오.
   */

   public static void main(String[] args){
       System.out.println(getDayDiff("20010103","20010101"));
       System.out.println(getDayDiff("20010103","20010103"));
       System.out.println(getDayDiff("20010103","200103"));
   }
}
```

[실행결과]

```
2
0
0
```

```
static int getDayDiff(String yyyymmdd1, String yyyymmdd2) {
    int diff = 0;
    try {
        int year1 = Integer.parseInt(yyyymmdd1.substring(0, 4));
        int month1 = Integer.parseInt(yyyymmdd1.substring(4, 6)) - 1;
        int day1 = Integer.parseInt(yyyymmdd1.substring(6, 8));

        int year2 = Integer.parseInt(yyyymmdd2.substring(0, 4));
        int month2 = Integer.parseInt(yyyymmdd2.substring(4, 6)) - 1;
        int day2 = Integer.parseInt(yyyymmdd2.substring(6, 8));

        Calendar date1 = Calendar.getInstance();
        Calendar date2 = Calendar.getInstance();

        date1.set(year1, month1, day1);
        date2.set(year2, month2, day2);

        diff = (int) ((date1.getTimeInMillis() - date2.getTimeInMillis()) / (24 * 60 * 60 * 1000)); // 시분초밀리

    } catch (Exception e) {
        diff = 0; // substring, parseInt에서 예외 발생시
    }
    return diff;
}
```

[10-6] 자신이 태어난 날부터 지금까지 며칠이 지났는지 계산해서 출력하시오.

[실행결과]

```
birth day=2000-01-01
today    =2016-01-29
5872 days
```

```
public static void main(String[] args) {
    LocalDate birthDay = LocalDate.of(1995, 1, 11);
    LocalDate now = LocalDate.now();

    long days = birthDay.until(now, ChronoUnit.DAYS); // enum 중 열거형 상수
    System.out.println(days);
}
```

[10-7] 2016년 12월 네번째 화요일의 날짜를 아래의 실행결과와 같은 형식으로 출력하시오.

[실행결과]

```
2016-12-27
```

```
public static void main(String[] args) {
        LocalDate date = LocalDate.of(2016, 12, 1);
//        System.out.println(date.with(ChronoField.ALIGNED_DAY_OF_WEEK_IN_MONTH(4, TUESDAY)));
    }
```

[10-8] 서울과 뉴욕간의 시차가 얼마인지 계산하여 출력하시오.

[실행결과]

```
2016-01-28T23:01:00.136+09:00[Asia/Seoul]
2016-01-28T09:01:00.138-05:00[America/New_York]
sec1=32400
sec2=-18000
diff=14 hrs
```

```
public static void main(String[] args) {
    ZonedDateTime zdt = ZonedDateTime.now();
    ZoneId nyId = ZoneId.of("America/New_York");
    ZonedDateTime zdtNY = ZonedDateTime.now().withZoneSameInstant(nyId);

    System.out.println(zdt); //2022-07-05T19:22:48.102+09:00[Asia/Seoul]
    System.out.println(zdtNY); //2022-07-05T06:22:48.103-04:00[America/New_York]

    long sec1 = zdt.getOffset().getTotalSeconds();
    long sec2 = zdtNY.getOffset().getTotalSeconds();
    long diff = (sec1 - sec2) / 3600;

    System.out.println(sec1);
    System.out.println(sec2);
    System.out.println("diff: " + diff + "hrs");
}
```