JavaStudyIntermediate3

Date& Calendar

- 날짜와 시간을 구하기 위한 클래스 Date 클래스
- 최초의 클래스, 지역화 고려 아님. (시차 고려 안해준다는 것.)
- Deprecated : 더 이상 지원하지 않는 기능이므로 사용을 자제하라는 의미

- **java.util.SimpleDateFormat 클래스를 이용해서 원하는 형태로 출력하는 방법**
  - yyyy는 년, MM은 월, dd는 일을 표현한다.
  - hh는 시간, mm은 분, ss는 초를 표현하며 a는 오전/오후 를 표현한다.
  - zzz는 TimeZone을 나타낸다. 한국의 경우 한국표준시 KST가 TimeZone에 해당하는 값입니다.

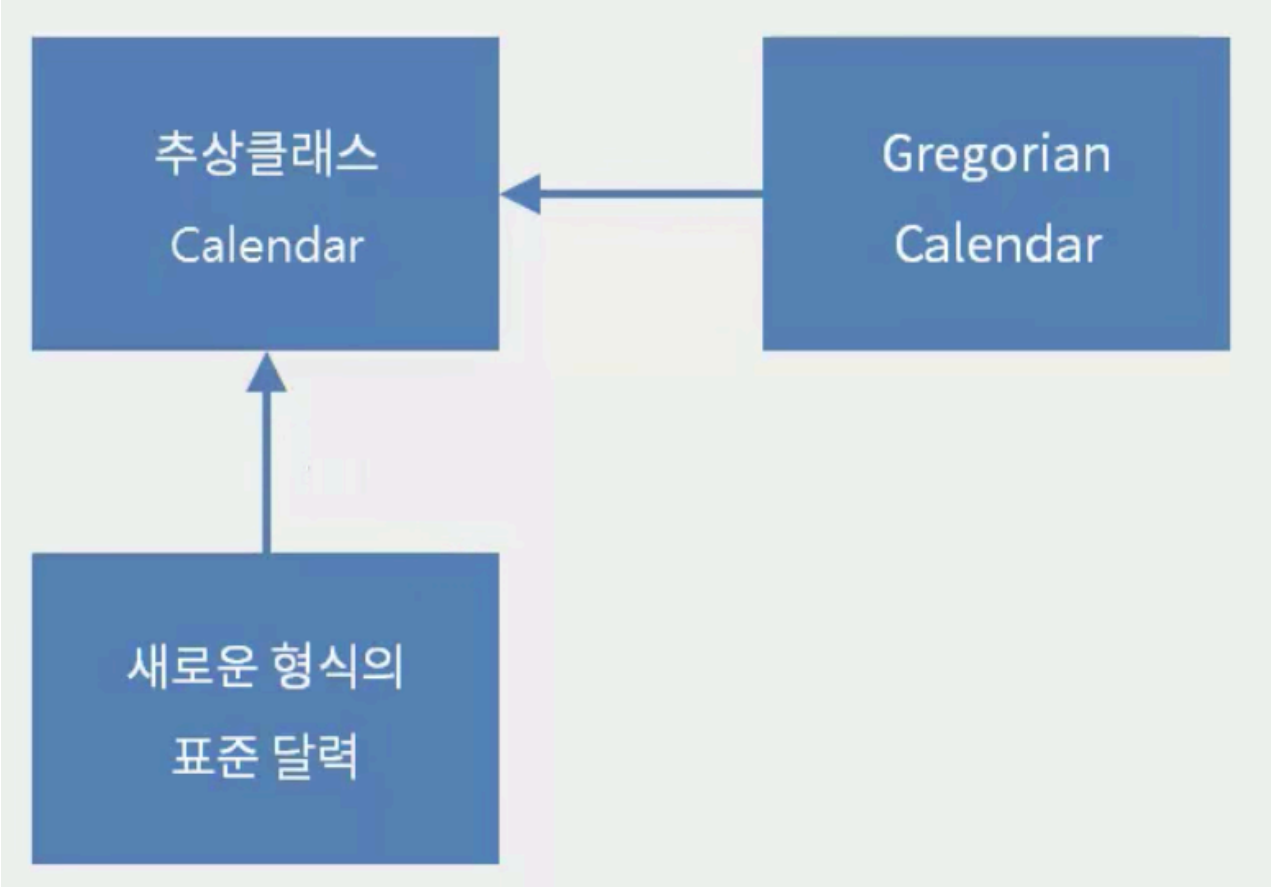
```
import java.text.SimpleDateFormat;
import java.util.Date;

public class DateExam {

  public static void main(String[] args) {
    // TODO Auto-generated method stub
    Date date = new Date();
    System.out.println(date.toString());

    SimpleDateFormat ft = new SimpleDateFormat("yyyy.MM.dd 'at' hh:mm:ss a zzz");
    System.out.print(ft.format(date));

    System.out.println(date.getTime());

    //현재 시간을 Long값으로 구하는 방법
    long today = System.currentTimeMillis();
    //system의 현재 시간을 long값으로 나한테 구해서 주세요.
    System.out.println(today);

    //시간 사이에서의 연산도 가능.
    System.out.println(today - date.getTime());

  }
}
```

Calendar

Calendar 클래스 생생 방법 : **지역성 보완**

- Calendar클래스는 추상클래스 -> **new 이용해 객체 생성 불가능**

- Calendar클래스에 대한 인스턴스를 생성하려면 Calendar가 가지고 있는 클래스 메소드 **getInstnace()**를 사용해야 한다.
- **getInstance()메소드를 호출하면 내부적으로 java.util.GregorianCalendar 인스턴스를 만들어서리턴한다.**
- GregorianCalendar는 Calendar의 자식 클래스이다. 그런데 이걸로 굳이 바로 인스턴스 만들지 않는 이유는, 표준 달력이 달라질 수도 있기 때문에.




```
import java.util.Calendar;

public class CalendarExam {

  public static void main(String[] args) {
    // TODO Auto-generated method stub
//    Calendar cal = new Calendar(); 불가능

    Calendar cal = Calendar.getInstance();

    System.out.println(cal.get(Calendar.YEAR));
    System.out.println(cal.get(Calendar.MONTH)+1);
    //자바가 0월부터 11월까지 표현.
    System.out.println(cal.get(Calendar.DATE));
    System.out.println(cal.get(Calendar.HOUR));
    System.out.println(cal.get(Calendar.HOUR_OF_DAY));
    //hour: 12시
    //hour of day : 24시

    //Int field, int amount
```

```
    cal.add(Calendar.HOUR, 5);
    //cal.add(Calendar.HOUR, -5); //전으로

    System.out.println(cal.get(Calendar.YEAR));
    System.out.println(cal.get(Calendar.MONTH)+1);
    System.out.println(cal.get(Calendar.DATE));
    System.out.println(cal.get(Calendar.HOUR));
    System.out.println(cal.get(Calendar.HOUR_OF_DAY));
  }
}
```

```
import java.util.*;

public class CalendarExam{
    public String hundredDaysAfter(){
    //오늘부터 100일 뒤의 날짜를 "2016년1월1일"의 형식으로 return하세요.
        Calendar cal = Calendar.getInstance();
        cal.add(Calendar.DATE,100);
        int yyyy = cal.get(Calendar.YEAR);
        int month = cal.get(Calendar.MONTH)+1;
        int date =  cal.get(Calendar.DATE);

        String str = yyyy + "년" + month +"월" + date+"일";
        return str;

    }
    public static void main(String[] args){
        CalendarExam ex = new CalendarExam();
        System.out.println(ex.hundredDaysAfter());
    }
}
```

Time

- Java에서 제공하는 Date, Time API는 부족한 기능 지원을 포함한 여러가지 문제점을 가지고 있었음
  - 1900년대만 제공된다던지...
  - 다양한 factory 메서드를 사용

1. now는 현재 시간을 구한다.
2. LocalDate.of, LocalTime.of

```
import java.time.LocalDate;
import java.time.LocalDateTime;
```

```java
import java.time.LocalTime;
import java.time.Month;

public class TimeExam {

  public static void main(String[] args) {
    // TODO Auto-generated method stub
    LocalDateTime timepoint = LocalDateTime.now();
    //현재의 날짜와 시간을 가진..
    System.out.println(timepoint);

    //int year, int month, int dayofMonth
    LocalDate ld1 = LocalDate.of(2021, 12, 12);
    //12 = > Month.DECEMBER
    System.out.println(ld1);

    //17시 18분에 대한 객체를 만들고 싶다.
    LocalTime lt1 = LocalTime.of(17, 18);
    System.out.println(lt1);
    //10시 15분 30초
    LocalTime lt2 =  LocalTime.parse("10:15:30");
    System.out.println(lt2);

    LocalDate theDate = timepoint.toLocalDate();
    System.out.println(theDate);
    Month month = timepoint.getMonth();
    System.out.println(timepoint.getMonthValue()); //숫자로
    System.out.println(month.getMonth()); //JANUARY

  }

}
```

3. Getter 메소드

```java
    LocalDate theDate = timePoint.toLocalDate();
    Month month = timePoint.getMonth();
    int day = timePoint.getDayOfMonth();
    int hour = timePoint.getHour();
    int minute = timePoint.getMinute();
    int second = timePoint.getSecond();
    // 달을 숫자로 출력한다 1월도 1부터 시작하는 것을 알 수 있습니다.
    System.out.println(month.getValue() + "/" + day + "  " + hour + ":" + minute
+ ":" + second);
```