Java 2

All about Date Class

By Assoc. Prof. Rangsit Sirirangsi

Edit by Dr.Watcharin Sarachai

Date, Calendar & DateFormat Objects

- java.util.Date: ใช้สำหรับการนำเสนอเกี่ยวกับวันเวลา
- java.text.DateFormat: จัดรูปแบบ date/time ในรูปของ Strings
- java.util.Calendar: จัดเตรียมวิธีการแปลงวันที่ระหว่างช่วงเวลาใดเวลาหนึ่ง

Date

- เป็นออปเจคที่ใช้สำหรับนำเสนอเวลาปัจจุบันหรือเวลาหนึ่ง ๆ
  - ในจาวาเวอร์ชันเก่า ๆ Date ออปเจคจะถูกใช้งานอย่างหลากหลาย
  - เมธอดส่วนใหญ่ในคลาส Date จะล้าสมัย
  - ปัจจุบันจะถูกแทนที่ด้วยคลาส Calendar

Date

- Date ประกอบไปด้วยคอนสตรัคเตอร์สองแบบ คือ
  - แบบแรกเป็นการสร้างออปเจคเพื่อนำเสนอวัน-เวลาปัจจุบัน

    public Date ()
  - แบบที่สองเป็นเป็นการนำเสนอค่าตัวเลขในหน่วย milliseconds ที่นับจาก January 1, 1970, 00:00:00 GMT

    public Date (**long** time)

Date

- คอนสตรัคเตอร์ของ Date ที่ไม่ใช้แล้ว (*Deprecated*)
  - แบบแรกเป็นการสร้างออปเจคพร้อมกำหนด ปี เดือน

    public ~~Date~~(int year, int month, int date)

    public ~~Date~~(int year, int month, int date, int hrs, int min)

    public ~~Date~~(int year, int month, int date, int hrs, int min, int sec)

Fully qualified name

- คลาส Date ถูกกำหนดไว้ในแพ็คเกจ java.util.Date
- ในกรณีที่ไม่มีการ import แพ็คเกจดังกล่าวมาใช้ อาจระบุตำแหน่งของแพ็คเกจที่ต้องการเรียกใช้ได้ดังนี้

```
public class First
{
    public static void main(String[] args)
    {
        java.util.Date d = new java.util.Date();
        System.out.println(d);
    }
}
```

Fully qualified name


```
public class First
{
    public static void main(String[] args)
    {
        java.util.Date d = new java.util.Date();
        System.out.println(d);
    }
}
```

- นำเสนอวัน-เวลาปัจจุบัน
- หากไม่กำหนดรูปแบบการแสดงผลลัพธ์จาก Date ไว้ แสดงผลดังนี้

Mon Feb 13 22:28:00 ICT 2017

Date Classes in Standard Library


```
import java.util.Date;

public class DateDemo  {

  public static void main(String[] args){
     Date date = new Date(1111111111);
     System.out.println(date);
  }
}
```

- รับค่าในหน่วย milliseconds ที่นับจาก January 1, 1970, 00:00:00 GMT
- ผลลัพธ์ดังต่อไปนี้

Wed Jan 14 03:38:31 ICT 1970

getTime()

- เมธอด getTime() ใช้สำหรับคืนค่าในหน่วย milliseconds ที่นับจาก January 1, 1970, 00:00:00 GMT

```
import java.util.Date;
class DateDemo {
  public static void main(String args[]) {
    Date date = new Date();
    System.out.println(date);
    long msec = date.getTime();
    System.out.println("Milliseconds since Jan. 1, 1970 GMT = "
                       + msec);
  }
}
```

Mon Feb 13 20:14:03 ICT 2017
Milliseconds since Jan. 1, 1970 GMT = 1486991643932

Methods of the Date class (in java.util)

- เมธอดพื้นฐานของคลาส Date คืนค่าแบบ Boolean มีดังต่อไปนี้

after()	ใช้ตรวจสอบว่า วันที่และเวลา มาหลัง วันที่และเวลาที่กำหนด หรือไม่
before()	ใช้ตรวจสอบว่า วันที่และเวลา มาก่อน วันที่และเวลาที่กำหนด หรือไม่
equals()	ใช้ตรวจสอบว่า วันที่และเวลา เท่ากันหรือไม่ โดยพิจารณาจากความแตกต่างในหน่วยมิลลิวินาที
compareTo()	ใช้ตรวจสอบว่า วันที่และเวลาใดมาก่อน วันที่และเวลาใดมาหลัง

- long getTime()
  - คืนค่าในหน่วย milliseconds นับจาก epoch (1970-01-01 00:00:00 GMT)
- void setTime(long n)

boolean : before, after, equal method


```
import java.util.*;

public class TestDate {

   public static void main(String[] args) {
    Date date_1 = new Date ();
    Date date_2 = new Date ();

    if ( date_1.after ( date_2 ) )
        System.out.println ( "date after" );
    else if (date_1.equals(date_2))
        System.out.println ( "date equals" );
     else
        System.out.println ( "date before" );
}
```

date equals

int compareTo (object_date)

- ใช้เปรียบเทียบลำดับก่อนหลังระหว่าง Date ออปเจคหลักและ Date ออปเจค ที่ถูกนำมาเปรียบเทียบในรูปของพารามิเตอร์
  - ในกรณีที่คืนค่าบวกแสดงว่า Date ออปเจคหลักมาก่อน
  - ในกรณีที่คืนค่าติดลบแสดงว่า Date ออปเจคหลักมาหลัง
  - ในกรณีที่คืนค่าศูนย์แสดงว่าสอง Date ออปเจคเท่ากัน

int compareTo (object_date)

- ตัวอย่างเช่น

```
Date date_1 = new Date ();
Date date_2 = new Date ();

int compare = date_1.compareTo ( date_2 );
```

The DateFormat Class

- เป็นคลาสแบบ abstract ที่ประกอบไปด้วยเมธอดที่ใช้สำหรับการรับค่า date และ time ที่มีรูปแบบเป็นค่า default หรือตาม locale และรูปแบบของ style ดังต่อไปนี้
  - **SHORT** เป็นค่าตัวเลขทั้งหมด เช่น 13/2/2017 (สำหรับ date) หรือ 3:30 pm (สำหรับ time)
  - **MEDIUM** เป็นรูปแบบปานกลาง เช่น Jan 12, 1952
  - **LONG** เป็นรูปแบบยาว เช่น January 12, 1952
  - **FULL** เป็นรูปแบบสมบูรณ์ เช่น

    Tuesday, April 12, 1952 AD or 3:30:42pm PST

DateFormat Class

- แต่รูปแบบการแสดงวันเดือนปีมีได้หลายรูปแบบ ดังนั้นจึงได้มีการออกแบบให้เมธอด getDateInstance() สามารถกำหนดพารามิเตอร์เพื่อให้สามารถแสดงรูปแบบได้ตามความต้องการผู้ใช้ เช่น
  - getDateInstance(DateFormat.SHORT);
  - getDateInstance(DateFormat.MEDIUM);
  - getDateInstance(DateFormat.LONG);
  - getDateInstance(DateFormat.FULL);

DateFormat Class

- การสร้าง DateFormat ออปเจคสามารถทำได้โดยการเรียกใช้เมธอด getDateInstance() ดังรูปแบบต่อไปนี้

```
DateFormat df = DateFormat.getDateInstance();
```

DateFormat Class

- คลาส DateFormat ถูกกำหนดไว้ในแพคเกจ java.text
  - ค่า Default ของ DateFormat เป็น medium ตัวอย่างเช่น

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

public class DateFormatDemo {
  public static void main(String[] args) {
     Date now = new Date();
     DateFormat df = DateFormat.getDateInstance();
     String s = df.format(now);
     System.out.println("Today is " + s);
  }
}
```

Today is Feb 13, 2017

DateFormat Class


```
import java.util.*;
import java.text.*;
public class Date1 {
   public static void main(String[] args) {
      Date now = new Date();
      DateFormat df =  DateFormat.getDateInstance();
      DateFormat df1 = DateFormat.getDateInstance(DateFormat.SHORT);
      DateFormat df2 = DateFormat.getDateInstance(DateFormat.MEDIUM);
      DateFormat df3 = DateFormat.getDateInstance(DateFormat.LONG);
      DateFormat df4 = DateFormat.getDateInstance(DateFormat.FULL);

      System.out.println("(Default) Today is " + df.format(now));
      System.out.println("(SHORT)   Today is " + df1.format(now));
      System.out.println("(MEDIUM)  Today is " + df2.format(now));
      System.out.println("(LONG)    Today is " + df3.format(now));
      System.out.println("(FULL)    Today is " + df4.format(now));
  }
}
```

(Default) Today is Feb 13, 2017
(SHORT) Today is 2/13/17
(MEDIUM) Today is Feb 13, 2017
(LONG) Today is February 13, 2017
(FULL) Today is Monday, February 13, 2017

Methods of the DateFormat class (in java.text)

- เมธอดพื้นฐานของคลาส DateFormat คืนค่า DateFormat มีดังต่อไปนี้

getInstance()	สร้าง DateFormat ให้มีรูปแบบวันที่เป็นแบบ short
getDateTimeInstance()	สร้าง DateFormat ให้มีรูปแบบวันที่และเวลาเป็นแบบ long
getDateInstance()	สร้าง DateFormat ให้มีการจัดรูปแบบวันที่เป็นแบบ medium
getDateInstance (constant, locale)	สร้าง Object DateFormat โดยมีการจัดรูปแบบวันที่ ตามลักษณะที่กำหนด
getTimeInstance()	สร้าง DateFormat โดยมีการจัดรูปแบบเวลา ให้เป็นแบบยาว
getTimeInstance (constant, locale)	ใช้ สร้าง Object DateFormat โดยมีการจัดรูปแบบเวลา ตามลักษณะที่กำหนด
format()	ใช้คืนค่า วันที่และเวลา ที่ได้หลังจากการจัดรูปแบบแล้ว

DateFormat with Locale

```
Class ShowDateFormat {
  public static void main(String [] args) {
    Date currentDate = new Date();
    Locale thailandLocale = Locale.forLanguageTag("th-TH");
    DateFormat shortFormat = DateFormat.getDateInstance(
                                DateFormat.SHORT, thailandLocale);
    DateFormat mediumFormat = DateFormat.getDateInstance(
                                DateFormat.MEDIUM, thailandLocale);
    DateFormat longFormat = DateFormat.getDateInstance(
                                DateFormat.LONG, thailandLocale);
    DateFormat fullFormat = DateFormat.getDateInstance(
                                DateFormat.FULL, thailandLocale);
    System.out.println("SHORT format: "
                       + shortFormat.format(currentDate));
    System.out.println("MEDIUM format: "
                       + mediumFormat.format(currentDate));
    System.out.println("LONG format: "
                       + longFormat.format(currentDate));
    System.out.println("FULL format: "
                       + fullFormat.format(currentDate));
  }
}
```

SHORT format: 23/12/67

MEDIUM format: 23 ธ.ค. 2567

LONG format: 23 ธันวาคม 2567

FULL format: วันจันทร์ที่ 23 ธันวาคม พุทธศักราช 2567

DateFormat Class : Time

- ในกรณีที่ต้องการสร้าง DateFormat ออปเจคสำหรับการนำเสนอค่าเวลา สามารถทำได้โดยการเรียกใช้เมธอด getTimeInstance() ดังรูปแบบต่อไปนี้

```
DateFormat df = DateFormat.getTimeInstance();
```

DateFormat Class : Time

- การนำเสนอค่าของเวลามีรูปแบบดังต่อไปนี้
  - Default: 10:12:34
  - SHORT: 10:12 น.
  - MEDIUM : 10:12:34
  - LONG : 10 นาฬิกา 12 นาที
  - FULL : 10 นาฬิกา 12 นาที 34 วินาที

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

public class DateFormatDemo
{
  public static void main(String args[]) {

      Date now = new Date();
      DateFormat defaultDf = DateFormat.getTimeInstance();
      DateFormat shortDf =
              DateFormat.getTimeInstance(DateFormat.SHORT);
      DateFormat mediumDf =
              DateFormat.getTimeInstance(DateFormat.MEDIUM);
      DateFormat longDf =
              DateFormat.getTimeInstance(DateFormat.LONG);
      DateFormat fullDf =
              DateFormat.getTimeInstance(DateFormat.FULL);

      System.out.println(" 1. " + defaultDf.format(now));
      System.out.println(" 2. " + shortDf.format(now));
      System.out.println(" 3. " + mediumDf.format(now));
      System.out.println(" 4. " + longDf.format(now));
      System.out.println(" 5. " + fullDf.format(now));
 }
}
```

1. 8:22:50 PM
2. 8:22 PM
3. 8:22:50 PM
4. 8:22:50 PM ICT
5. 8:22:50 PM ICT

SimpleDateFormat

- แม้ว่าคลาส DateFormat ที่ประกอบไปด้วยเมธอดแบบ static ที่สามารถใช้งานได้หลากหลายรูปแบบ แต่ยังคงไม่เพียงพอต่อความต้องการของผู้ใช้
- ดังนั้นจึงได้มีการสร้างคลาสสืบทอด SimpleDateFormat เพื่อช่วยให้ผู้ใช้สามารถนำเสนอรูปแบบของ Date ได้ตามต้องการ
- คลาส SimpleDateFormat ถูกกำหนดไว้ใน java.text.* ยอมให้ผู้ใช้สามารถกำหนดรูปแบบการแสดงผลของ Date objects โดยใช้ formats ต่าง ๆ
- คลาส SimpleDateFormat มักใช้ร่วมกับ Date objects เสมอ

SimpleDateFormat : Pattern

- รูปแบบ pattern บางส่วนมีดังต่อไปนี้

ตัวอย่างรูปแบบในการนำเสนอ		
0	MM/dd/yyyy	01/01/2011
1	EEEE, dd MMMM yyyy	Saturday, 01 January 2011
2	EEEE, dd MMMM yyyy HH:mm	Saturday, 01 January 2011 08:38
3	EEEE, dd MMMM yyyy hh:mm tt	Saturday, 01 January 2011 08:38 AM
4	MM/dd/yyyy HH:mm	01/01/2011 08:38
5	MM/dd/yyyy hh:mm tt	01/01/2011 08:38 AM
6	MM/dd/yyyy H:mm	01/01/2011 8:38
7	MM/dd/yyyy h:mm tt	01/01/2011 8:38 AM
8	MM/dd/yyyy HH:mm:ss	01/01/2011 08:38:31
9	MMMM dd	January 01
10	MMMM dd	January 01

SimpleDateFormat class

- การนำเสนอ Date โดยใช้ SimpleDateFormat
  - ใช้ pattern เพื่อกำหนดรูปแบบในการนำเสนอ

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

public class JavaSimpleDateFormat  {

  public static void main(String args[]){

   Date date = new Date();

   String DATE_FORMAT = "MM/dd/yyyy";

   SimpleDateFormat sdf = new SimpleDateFormat(DATE_FORMAT);

   System.out.println("Today is " + sdf.format(date) );
  }
}
```

Today is 02/13/2017

Formatting year using SimpleDateFormat


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

public class FormattingYear {

  public static void main(String[] args) {

    Date date = new Date();

    String strDateFormat = "yy";
    SimpleDateFormat sdf =
        new SimpleDateFormat(strDateFormat);

    System.out.println("Year in yy format : "
                       + sdf.format(date));

    strDateFormat = "yyyy";
    sdf = new SimpleDateFormat(strDateFormat);
    System.out.println("Year in yyyy format : " +
                       sdf.format(date));
  }
}
```

Year in yy format : 17
Year in yyyy format : 2017

Formatting month using SimpleDateFormat


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

public class FormattingMonth {
  public static void main(String[] args) {

    Date date = new Date();

    String strDateFormat = "MM";
    SimpleDateFormat sdf = new SimpleDateFormat(strDateFormat);
    System.out.println("Current Month in MM format : " 
                       + sdf.format(date));
    strDateFormat = "MMM";
    sdf = new SimpleDateFormat(strDateFormat);
    System.out.println("Current Month in MMM format : " 
                       + sdf.format(date));
    strDateFormat = "MMMM";
    sdf = new SimpleDateFormat(strDateFormat);
    System.out.println("Current Month in MMMM format : " 
                       + sdf.format(date));
  }
}
```

Current Month in MM format : 02
Current Month in MMM format : Feb
Current Month in MMMM format : February

Formatting hour using SimpleDateFormat


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

public class FormattingHour {
  public static void main(String[] args) {

    Date date = new Date();

    String strDateFormat = "h";//formatting hour in h (1-12 in AM/PM)
    SimpleDateFormat sdf = new SimpleDateFormat(strDateFormat);
    System.out.println("hour in h format : " + sdf.format(date));

    strDateFormat = "hh";       //formatting hour in hh (01-12 )
    sdf = new SimpleDateFormat(strDateFormat);
    System.out.println("hour in hh format : " + sdf.format(date));

    //formatting hour in H (0-23) format like 0, 1...23.
    strDateFormat = "H";
    sdf = new SimpleDateFormat(strDateFormat);
    System.out.println("hour in H format : " + sdf.format(date));

    // ต่อหน้าถัดไป ...
```

Formatting hour using SimpleDateFormat


```
    //formatting hour in HH (00-23) format like 00, 01..23.
    strDateFormat = "HH";
    sdf = new SimpleDateFormat(strDateFormat);

    System.out.println("hour in HH format : " + sdf.format(date));

    //formatting hour in k (1-24) format like 1, 2..24.
    strDateFormat = "k";
    sdf = new SimpleDateFormat(strDateFormat);

    System.out.println("hour in k format : " + sdf.format(date));

    //formatting hour in kk (01-24) format like 01, 02..24.
    strDateFormat = "kk";
    sdf = new SimpleDateFormat(strDateFormat);

    System.out.println("hour in kk format : " + sdf.format(date));
  }
}
```

```
hour in h format : 8
hour in hh format : 08
hour in H format : 20
hour in HH format : 20
hour in k format : 20
hour in kk format : 20
```

SimpleDateFormat

Symbol	Meaning	Type	Example
G	Era Designator	Text	“AD”
y	Year	Number	“yy” -> “03″ “yyyy” -> “2003″
M	Month	Text or Number	“M” -> “7″ “M” -> “12″ “MM” -> “07″ “MMM” -> “Jul” “MMMM” -> “December”
d	Day in month	Number	“d” -> “3″ “dd” -> “03″
h	Hour (1-12, AM/PM)	Number	“h” -> “3″ “hh” -> “03″
H	Hour (0-23)	Number	“H” -> “15″ “HH” -> “15″
m	Minute	Number	“m” -> “7″ “m” -> “15″ “mm” -> “15″
s	Second	Number	“s” -> “15″ “ss” -> “15″
S	Millisecond (0-999)	Number	“SSS” -> “007″

SimpleDateFormat

Symbol	Meaning	Type	Example
E	Day in week	Text	“EEE” -> “Tue” “EEEE” -> “Tuesday”
D	Day in year (1-365 or 1-364)	Number	“D” -> “65″ “DDD” -> “065″
F	Day of week in month (1-5)	Number	“F” -> “1″
w	Week in year (1-53)	Number	“w” -> “7″
W	Week in month (1-5)	Number	“W” -> “3″
a	AM/PM	Text	“a” -> “AM” “aa” -> “AM”
z	Time zone	Text	“z” -> “EST” “zzz” -> “EST” “zzzz” -> “Eastern Standard Time”
‘	Excape for text	Delimiter	“‘hour’ h” -> “hour 9″
”	Single quote	Literal	“ss”SSS” -> “45′876″

Formatting day of week using SimpleDateFormat

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

public class FormattingDayOfWeek {
  public static void main(String[] args) {
    Date date = new Date();

    //formatting day of week in E format like Sun, Mon etc.
    String strDateFormat = "E";
    SimpleDateFormat sdf = new SimpleDateFormat(strDateFormat);

    System.out.println("Current day of week in E format : "
                       + sdf.format(date));
    strDateFormat = "EEEE";

    sdf = new SimpleDateFormat(strDateFormat);
    System.out.println("Current day of week in EEEE format : "
                       + sdf.format(date));
  }
}
```

Current day of week in E format : Mon
Current day of week in EEEE format : Monday

Formatting in custom formats : SimpleDateFormat


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

public class FormattingDateInCustomFormat {
  public static void main(String[] args) {

    Date date = new Date();

    SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
    String strDate = sdf.format(date);
    System.out.println("formatted date in dd/MM/yyyy : "
                       + strDate);

    //format date in mm-dd-yyyy hh:mm:ss format
    sdf = new SimpleDateFormat("MM-dd-yyyy hh:mm:ss");
    strDate = sdf.format(date);
    System.out.println("formatted date in mm-dd-yyyy hh:mm:ss : "
                       + strDate);

  }
}
```

formatted date in dd/MM/yyyy : 13/02/2017
formatted date in mm-dd-yyyy hh:mm:ss : 02-13-2017 08:35:26

Understanding the parse() Method of SimpleDateFormat

- เมธอด parse() คืออะไร
  - ใช้สำหรับเปลี่ยน String เป็นออปเจค Date
  - มีประโยชน์สำหรับการตรวจสอบและประมวลผลข้อมูลอินพุตของผู้ใช้
  - parse() Method Syntax:

```
Date parse(String source) throws ParseException
```

คำอธิบาย:

- Input: A String representing the date in the specified format.
- Output: A Date object.
- Exception: Throws Parse Exception if the input string doesn’t match the format.

Understanding the parse() Method of SimpleDateFormat

- Common Date Patterns

Pattern	Example
dd/MM/yyyy	23/12/2024
MM-dd-yyyy	12-23-2024
yyyy.MM.dd	2024.12.23
EEEE, MMM d, yyyy	Monday, Dec 23, 2024

- Note: Patterns are case-sensitive.

Understanding the parse() Method of SimpleDateFormat

- Parsing a Date String

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

public class ParseExample {
    public static void main(String[] args) {
        String dateString = "23/12/2024";
        SimpleDateFormat formatter =
                        new SimpleDateFormat("dd/MM/yyyy");

        try {
            Date date = formatter.parse(dateString);
            System.out.println("Parsed Date: " + date);
        } catch (ParseException e) {
            System.out.println("Invalid date format: "
                               + e.getMessage());
        }
    }
}
```

```
Parsed Date: Mon Dec 23 00:00:00 GMT 2024
```

Understanding the parse() Method of SimpleDateFormat

- Handling Errors

```
try {
    Date date = formatter.parse("Invalid Date");
} catch (ParseException e) {
    System.out.println("Error: " + e.getMessage());
}
```

- **Key Points**:
  - Always surround parse() with a try-catch block.
  - The ParseException helps identify invalid input.

Understanding the parse() Method of SimpleDateFormat

- การเปลี่ยนเวลาจาก String เป็นออปเจค Date

```
// Define the time format
String timeString = "15:30:45"; // Example time
SimpleDateFormat timeFormat = new SimpleDateFormat("HH:mm:ss");
try {
  // Parse the time string into a Date object
  Date time = timeFormat.parse(timeString);
  System.out.println("Parsed Time: " + time);
} catch (ParseException e) {
  System.out.println("Invalid time format: " + e.getMessage());
}
```

- Explanation
  - HH: Hours in 24-hour format (00-23).
  - mm: Minutes (00-59).
  - ss: Seconds (00-59).

Calendars vs. Dates

- คลาส Date ไม่สามารถวัดในรูป months, weekdays, etc.
- ดังนั้นจึงจำเป็นต้องใช้คลาส C*alendar* เพื่อให้สามารถแปลงค่าระหว่างออปเจค Date หนึ่ง ๆ ให้อยู่ในรูปของค่าที่กำหนดไว้ในคลาส **Calendar** เช่น HOUR, YEAR, MONTH, DAY_OF_MONTH.
- โดยปกติแล้วการสร้าง Calendar ออปเจคใช้หลักการสร้างจาก Object Factory จากเมธอดดังต่อไปนี้:
  - **public static** Calendar getInstance ()
  - **public static** Calendar getInstance (Locale locale)

```
Calendar c = Calendar.getInstance();
```

Calendar Methods

getInstance()	ใช้สร้าง object calendar
getInstance(object_locale)	ใช้สร้าง object calendar และจะมีค่าปฏิทินตาม object locale ที่กำหนด
setTime()	ใช้กำหนดวันที่และเวลาให้กับ object calendar
getTime()	ใช้คืนค่า Object date ที่เก็บค่าวันที่และเวลาเอาไว้
set()	ใช้กำหนดค่า ปี เดือน และวัน ให้กับ Object Calendar
after()	ใช้ตรวจสอบว่า ปฏิทิน มาหลัง ปฏิทินที่กำหนด หรือไม่
before()	ใช้ตรวจสอบว่า ปฏิทิน มาก่อน ปฏิทินที่กำหนด หรือไม่
equals()	ใช้ตรวจสอบว่า ปฏิทิน เท่ากันกับ ปฏิทินที่กำหนด หรือไม่
get()	ใช้คืนค่า หน่วยของ ปฏิทินที่ต้องการ
set(calendar_constant, int_value)	ใช้กำหนดค่า หน่วยของ ปฏิทินที่ต้องการ
setFirstDayOfWeek()	ใช้กำหนด วันเริ่มต้นของ สัปดาห์
clear()	ใช้ ลบค่าข้อมูล ปฏิทิน ของ calendar

Calendar Object

- นอกจากนั้นคลาส Calendar ยังประกอบไปด้วย static int field เพื่อใช้ในการอ่านค่าต่าง ๆ เช่น

Access Method	Meaning
get(Calendar.YEAR)	int value of the year
get(Calendar.MONTH)	int value of the month (0-11)
get(Calendar.DAY_OF_MONTH)	int value of the day of the month (1-31)
get(Calendar.DAY_OF_WEEK)	int value of the day of the week (1-7)
get(Calendar.HOUR)	int value of the hour in 12 hour notation (0-12)
get(Calendar.AM_PM)	returns either Calendar.AM or Calendar.PM
get(Calendar.HOUR_OF_DAY)	int value of the hour of the day in 24-hour notation (0-24)
get(Calendar.MINUTE)	int value of the minute in the hour (0-59)
get(Calendar.SECOND)	int value of the second within the minute (0-59).
get(Calendar.MILLISECOND)	int value of the milliseconds within a second (0-999).

Calendar Example


```
import java.util.*;
class CalPlay {
  public static void main(String []args) {
     Calendar c = Calendar.getInstance();
     System.out.println("Today is " +
          (c.get(Calendar.MONTH)+1) + "/" +
          c.get(Calendar.DATE) + "/" +
          c.get(Calendar.YEAR));
  }
}
```

Today is 2/13/2017

Simple Java Calendar Example


```
import java.util.Calendar;

public class JavaSimpleCalendarExample {

  public static void main(String[] args) {

    // use getInstance() method to
    // get object of java Calendar class
    Calendar cal = Calendar.getInstance();

    // use getTime() method of Calendar class to get date and time
    System.out.println("Today is : " + cal.getTime());
  }
}
```

Today is : Mon Feb 13 20:37:18 ICT 2017

Another Example


```
class CalPlay {
  final static String[] DAYS ={ "Sunday","Monday",
     "Tuesday", "Wednesday", "Thursday",
     "Friday", "Saturday" };

  public static void main(String []args) {
    Calendar c = Calendar.getInstance();
    System.out.println("Today is " + dayName(c));
  }
  public static String dayName( Calendar c) {
    return(DAYS[c.get(Calendar.DAY_OF_WEEK)-1]);
  }
}
```

Today is Monday

Calendar Class


```
import java.util.Calendar;
public class Date1  {
  public static void main(String[] args)  {
    Calendar cal = Calendar.getInstance();
    cal.clear();
    cal.set(2015,3,22);

    int day = cal.get(Calendar.DATE);
    int month = cal.get(Calendar.MONTH) + 1;
    int year = cal.get(Calendar.YEAR);
    int dow = cal.get(Calendar.DAY_OF_WEEK);
    int dom = cal.get(Calendar.DAY_OF_MONTH);
    int doy = cal.get(Calendar.DAY_OF_YEAR);

    System.out.println("Current Date: " + cal.getTime());
    System.out.println("Day: " + day);
    System.out.println("Month: " + month);
    System.out.println("Year: " + year);
    System.out.println("Day of Week: " + dow);
    System.out.println("Day of Month: " + dom);
    System.out.println("Day of Year: " + doy);
  }
}
```

Current Date: Wed Apr 22 00:00:00 ICT 2015
Day: 22
Month: 4
Year: 2015
Day of Week: 4
Day of Month: 22
Day of Year: 112

Get Week of month and year using Calendar


```
import java.util.Calendar;

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

    System.out.println("Current week of month is : " +
                       now.get(Calendar.WEEK_OF_MONTH));

    System.out.println("Current week of year is : " +
                       now.get(Calendar.WEEK_OF_YEAR));

    now.add(Calendar.WEEK_OF_MONTH, 1);
    System.out.println("date after one year : " +
                      (now.get(Calendar.MONTH) + 1)
                      + "-"
                      + now.get(Calendar.DATE)  + "-" +
   now.get(Calendar.YEAR));
  }
}
```

Current week of month is : 3
Current week of year is : 7
date after one year : 2-20-2017

Get current date time values using Java Calendar


```
import java.util.Calendar;
public class GetCurrentDateTimeExample {
  public static void main(String[] args) {
    Calendar now = Calendar.getInstance();
    //get current date, year and month
    System.out.println("Current Year is : " +
                        now.get(Calendar.YEAR));
    //month start from 0 to 11
    System.out.println("Current Month is : " +
                       (now.get(Calendar.MONTH) + 1 ));
    System.out.println("Current Date is : " +
                      now.get(Calendar.DATE));
    //get current time information
    System.out.println("Current Hour in 12 hour format is : "
                     + now.get(Calendar.HOUR));
    System.out.println("Current Hour in 24 hour format is : "
                     + now.get(Calendar.HOUR_OF_DAY));
    System.out.println("Current Minute is : "
                     + now.get(Calendar.MINUTE));
    System.out.println("Current Second is : "
                     + now.get(Calendar.SECOND));
    System.out.println("Current Millisecond is : "
                     +  now.get(Calendar.MILLISECOND));
```

Get current date time values using Java Calendar


```
    System.out.println("Current full date time is : "
                    + (now.get(Calendar.MONTH) + 1) + "-"
                    + now.get(Calendar.DATE) + "-"
                    + now.get(Calendar.YEAR) + " "
                    + now.get(Calendar.HOUR_OF_DAY) + ":"
                    + now.get(Calendar.MINUTE) + ":"
                    + now.get(Calendar.SECOND) + "."
                    + now.get(Calendar.MILLISECOND));
  }
}
```

Current Year is : 2017
Current Month is : 2
Current Date is : 13
Current Hour in 12 hour format is : 8
Current Hour in 24 hour format is : 20
Current Minute is : 40
Current Second is : 21
Current Millisecond is : 43
Current full date time is : 2-13-2017 20:40:21.43

Set Year, Month, Day

- นอกจากนั้นคลาส Calendar ยังสามารถกำหนดค่าต่าง ๆ ผ่านเมธอด set() ได้ดังต่อไปนี้:

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

    System.out.println("Calendar before Setting is “
                       +c.getTime());
    c.clear();
    // Sets the values for the calendar fields YEAR,
    // MONTH,and DAY_OF_MONTH.
    c.set(2012,10,27);

    System.out.println("Calendar after Setting is “
                       +c.getTime());
  }
}
```

Calendar before Setting is Mon Feb 13 20:40:48 ICT 2017
Calendar after Setting is Tue Nov 27 00:00:00 ICT 2012

Adding and Subtracting to Year, Month, Day etc.

- นอกจากนั้นคลาส Calendar ยังยอมให้ผู้ใช้สามารถอัพเดทค่าผ่านเมธอด add() ได้อีกด้วย เช่น

```
Calendar calendar = new Calendar();
//set date to last day of 2009
calendar.set(Calendar.YEAR, 2009);
calendar.set(Calendar.MONTH, 11); // 11 = december
calendar.set(Calendar.DAY_OF_MONTH, 31); // new years eve //add one day

calendar.add(Calendar.DAY_OF_MONTH, 1); //date is now jan. 1st 2010

int year = calendar.get(Calendar.YEAR); // now 2010
int month = calendar.get(Calendar.MONTH); // now 0 (Jan = 0)
int dayOfMonth = calendar.get(Calendar.DAY_OF_MONTH); // now 1
```

Manipulate Date

- **Date** ออปเจคสามารถนำค่าออกมาคำนวณได้ ทั้งนี้เนื่องจากมีค่าเป็นตัวเลขในหน่วย millisecond
- ดังนั้นการคำนวณค่าระหว่างสอง **Date** ออปเจคสามารถทำได้ดังนี้

  (date1.getTime() - date2.getTime()) / (1000*60*60*24)

- ส่วนคลาส Calendar สามารถใช้เมธอด add() และ set() ในการคำนวณ Date และ Time เป็นหลัก

Day Difference

```
import java.util.*;
import java.text.*;
public class Date1  {
  public static void main(String[] args)   {
    Calendar cal1 = Calendar.getInstance();
    Calendar cal2 = Calendar.getInstance();

    cal1.set(2010, 9, 31);
    cal2.set(2010, 10, 1);
    long milis1 = cal1.getTimeInMillis();
    long milis2 = cal2.getTimeInMillis();
    long diff = milis2 - milis1;
    long diffSeconds = diff / 1000;
    long diffMinutes = diff / (60 * 1000);
    long diffHours = diff / (60 * 60 * 1000);
    long diffDays = diff / (24 * 60 * 60 * 1000);

    System.out.println("In milliseconds: " + diff
                       + "  milliseconds.");
        System.out.println("In seconds: "
                       + diffSeconds + " seconds.");
        System.out.println("In minutes: "
                       + diffMinutes + " minutes.");
        System.out.println("In hours: " + diffHours + " hours.");
        System.out.println("In days: " + diffDays + " days.");
   }
}
```

In milliseconds: 86400013 milliseconds.
In seconds: 86400 seconds.
In minutes: 1440 minutes.
In hours: 24 hours.
In days: 1 days.

Add or substract years to current date using Calendar


```
import java.util.Calendar;

public class AddYearToCurrentDate {
  public static void main(String[] args) {
    //create Calendar instance
    Calendar now = Calendar.getInstance();

    System.out.println("Current date : "
                       + (now.get(Calendar.MONTH) + 1)
                       + "-"
                       + now.get(Calendar.DATE)
                       + "-"
                       + now.get(Calendar.YEAR));

   //add year to current date using Calendar.add method
    now.add(Calendar.YEAR,1);

    System.out.println("date after one year : "
                       + (now.get(Calendar.MONTH) + 1)
                       + "-"
                       + now.get(Calendar.DATE)
                       + "-"
                       + now.get(Calendar.YEAR));
```

Add or substract years to current date using Java Calendar


```
        //substract year from current date
        now = Calendar.getInstance();

        now.add(Calendar.YEAR, -100);
        System.out.println("date before 100 years : "
                           + (now.get(Calendar.MONTH) + 1)
                           + "-"
                           + now.get(Calendar.DATE)
                           + "-"
                           + now.get(Calendar.YEAR));

    }
}
```

Current date : 2-13-2017
date after one year : 2-13-2018
date before 100 years : 2-13-1917