

3 – Trabajando con fechas en Java

Objetivo: operar con las distintas clases que ofrece Java para manipular objetos de tipo fecha.

Para manipular fechas el procedimiento consiste en crear un objeto instanciando alguna de las clases que lo permiten (Date ó Calendar) y luego usar el objeto enviándole mensajes según los métodos que proporcione cada clase. Existen tres clases en la documentación Java API que permiten manipular fechas:

- **Date** crea un objeto Date (paquete java.util), que representa un momento específico del tiempo en milisegundos.
- **Calendar** configura o cambia la fecha de un objeto Date (paquete java.util). permitiendo obtener números enteros que especifican el día, mes, año, hora, etc. de dicha fecha.
- **DateFormat** muestra la fecha en diferentes formatos (paquete java.text).

La jerarquía de clases es la siguiente: a travez través

java.util

```
java.lang.Object
└─ java.util.Date

java.lang.Object
└─ java.util.Calendar
    └─ java.util.GregorianCalendar
```

java.text

```
java.lang.Object
└─ java.text.Format
    └─ java.text.DateFormat
        └─ java.text.SimpleDateFormat
```

Clase Date

Para obtener una instancia de la clase Date utilizar el método constructor Date(). Devuelve una fecha con los valores por defecto establecidos en el sistema. Existen varios métodos para esta clase, sin embargo la mayoría es deprecated, dado que a partir de las nuevas versiones de java se trabaja preferentemente con la clase Calendar

```
Date f_Date = new Date();
System.out.println("Fecha actual con Date: " + f_Date);
```

Muestra con este formato → Fecha actual con Date: Tue Sep 06 20:31:30 GMT-03:00 2011

También es posible obtener un Date desde un objeto Calendar:

```
Date f_Date = f_Calendar.getTime();
```

Clase Calendar

Se puede obtener una instancia de la clase Calendar, con los valores por defecto establecidos en el sistema en cuanto a hora y formato, de las siguientes maneras:

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

Calendar f_Calendar = new GregorianCalendar();
```

Luego se obtienen los valores de interés mediante el método get(), que admite como parámetros una serie de constantes definidas en la clase Calendar que representan día, mes, año, etc. (Calendar.DATE, Calendar.MONTH, Calendar.YEAR, etc.)

```
System.out.println("Fecha actual con Calendar: "
    + f_Calendar.get(Calendar.DATE) + "/"
    + (f_Calendar.get(Calendar.MONTH)+1) + "/"
    + f_Calendar.get(Calendar.YEAR) );
```

Atención: trabaja los meses como un array con subíndice comenzando en 0 para Enero. Por lo tanto se debe **sumar “1”** al valor que retorna.

Muestra con este formato → 27/8/2013

3 – Trabajando con fechas en Java

Otro modo de imprimir objetos de tipo Calendar, por medio de tabulaciones, es el siguiente:

```
Calendar f_c = Calendar.getInstance();
System.out.format("%tB %te, %tY%n", f_c, f_c, f_c);
System.out.format("%tl:%tM %tp%n", f_c, f_c, f_c);
System.out.format("%tD%n", f_c);
```

Salida

```
agosto 27, 2013
4:40 pm
08/27/13
```

Se puede establecer una fecha determinada con Calendar: `f_Calendar.set(2011, 9, 16);`
También se puede establecer la fecha desde un objeto Date: `f_Calendar.setTime(f_Date);`

Otras operaciones con fechas

Para sumar/restar días, meses, etc. a una fecha (método add de la clase Calendar)

```
f_Calendar.add(Calendar.DATE, -5);
f_Calendar.add(Calendar.MONTH, 2);
```

Otros métodos para sumar días

```
f_Calendar.add(Calendar.DAY_OF_YEAR, -5);
System.out.println("Otra Forma ***** :Fecha -5 días:" + formato2.format(f_Calendar.getTime()));
```

```
f_Calendar.add(Calendar.DAY_OF_MONTH, 2);
System.out.println("Otra Forma ***** :Fecha +2 días:" + formato2.format(f_Calendar.getTime()));
```

Para comparar fechas utilizar métodos after, before, equals de la clase Calendar

```
if (fecha_1.after(fecha_2)) {
    System.out.println("fecha_1 > fecha_2");
}
else
    if (fecha_1.before(fecha_2)) {
        System.out.println("fecha_1 < fecha_2");
    }
}
```

Formatos

Para presentar las fechas con distintos formatos, se pueden utilizar las clases *DateFormat* o *SimpleDateFormat*. Para ello es necesario crear un objeto de estas clases con el formato deseado. A este objeto se aplica luego el método format, pasándole como parámetro la fecha a la cual se quiere formatear (debe ser de tipo Date).

- Con DateFormat

```
DateFormat formato1 = DateFormat.getDateInstance(DateFormat.FULL);
System.out.println("Fecha formateada DateFormat:" + formato1.format(f_Date));
```

Estilos de formato:

- **SHORT**, totalmente numérico (18/11/05 O 9:01)
- **MEDIUM**, formato largo (18-nov-05 O 9:00:20)
- **LONG**, otro formato largo (18 de noviembre de 2005 O 9:02:12 GMT+01:00)
- **FULL**, con la especificación completa (lunes 18 de noviembre de 2005 O 09H03' GMT+01:00)

- Con SimpleDateFormat

```
SimpleDateFormat formato2 = new SimpleDateFormat("dd/MM/yyyy");
System.out.println("Fecha formateada SimpleDateFormat:" + formato2.format(f_Date));
```

Algunos formatos:

- dd.MM.yy → 09.04.11
- yyyy.MM.dd G 'at' hh:mm:ss z → 2011.04.09 AD at 06:15:55 PDT
- EEE, MMM d, 'yy → Thu, Apr 9, '11
- h:mm a → 6:15 PM
- H:mm → 18:15
- H:mm:ss.SSS → 18:15:55.624
- K:mm a,z → 6:15 PM,PDT
- yyyy.MMMMMM.dd GGG hh:mm aaa

Ejemplo

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

public class FechaDemo{
    public static void main(String a[]){
        // con Date
        Date f_Date = new Date();
        System.out.println("Fecha actual con Date: " + f_Date);

        // con Calendar
        Calendar f_Calendar = Calendar.getInstance();
        System.out.println("Fecha actual con Calendar: "
            + f_Calendar.get(Calendar.DATE) + "/"
            + (f_Calendar.get(Calendar.MONTH)+1) + "/"
            + f_Calendar.get(Calendar.YEAR));

        // establecer una fecha determinada con Calendar
        f_Calendar.set(2006, 9, 16);
        // obtener un Date desde un Calendar
        f_Date=f_Calendar.getTime();
        // obtener un Calendar desde un Date
        f_Calendar.setTime(f_Date);

        // crear objeto con formato deseado. Aplicar método format(Date)
        DateFormat formato1 = DateFormat.getDateInstance(DateFormat.FULL);
        System.out.println("Fecha formateada DateFormat: " + formato1.format(f_Date));

        SimpleDateFormat formato2=new SimpleDateFormat("dd.MM.yy");
        System.out.println("Fecha formateada SimpleDateFormat: " + formato2.format(f_Date));

        // sumar/restar días, meses, años
        f_Calendar.add(Calendar.DATE, -5);
        f_Calendar.add(Calendar.MONTH, 2);
        System.out.println("Fecha -5 días + 2 meses: " + formato2.format(f_Calendar.getTime()));

        // otros métodos para sumar días
        f_Calendar.add(Calendar.DAY_OF_YEAR, -5);
        System.out.println("Otra Forma ***** :Fecha -5 días:" + formato2.format(f_Calendar.getTime()));

        f_Calendar.add(Calendar.DAY_OF_MONTH, 2);
        System.out.println("Otra Forma ***** :Fecha +2 días:" + formato2.format(f_Calendar.getTime()));

        // comparar fechas
        Calendar f1=Calendar.getInstance();
        Calendar f2=Calendar.getInstance();
        f1.set(2006, 9, 16);
        f2.set(2006, 10, 6);
        if(f1.equals(f2)){
            System.out.println("Fechas iguales");
        }
        if(f1.after(f2)){
            System.out.println("f1 > f2");
        }
        if(f1.before(f2)){
            System.out.println("f1 < f2");
        }
    }
}
```