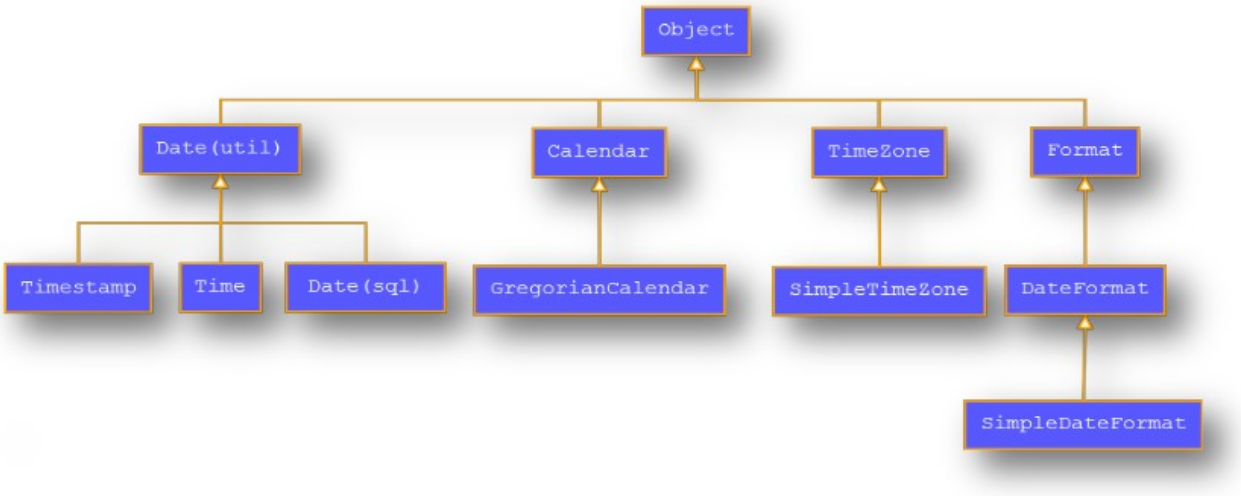
FECHAS Y HORAS



Date (util)

Hereda de la clase Object e implementa las interfaces Serializable, Cloneable y Comparable.

Esta clase, Date, representa un instante específico expresado en milisegundos.

Antes de JDK 1.1 esta clase tenía dos funciones adicionales, interpretar fechas y formatear y analizar cadenas de fechas. Pero con JDK1.1 aparecieron la clase Calendar, encargada de realizar las conversiones entre campos de fecha y tiempo, y la clase DateFormat, encargada del formateo y análisis de las cadenas de fechas.

En todos los métodos de la clase Date que acepten o devuelvan año, mes, día, horas, minutos y segundos como valores, las siguientes representaciones son usadas:

- Un año "y" es representado por el entero "y" - 1990.
- Un mes es representado por un entero entre 0 y 11; 0 es Enero y 11 es Diciembre.
- Una fecha (día del mes) es representado por un entero entre 1 y 31.
- Una hora es representada por un entero entre 0 y 23.
- Un minuto es representado por un entero entre 0 y 59.
- Un segundo es representado por un entero entre 0 y 61. Con 60 y 61 se cubre el conocido como segundo intercalar. Un segundo intercalar o segundo adicional es un ajuste de un segundo para mantener los estándares de emisión de tiempo cercanos al tiempo solar medio. Los segundos intercalares son necesarios para mantener los estándares sincronizados con los calendarios civiles, cuya base es astronómica.

Aun así, los valores proporcionados no han de estar dentro de los límites, por ejemplo, si introducimos el 32 de enero, ésta fecha será el 1 de Febrero.

Time

Hereda de Date(util) e implemente las interfaces Serializable, Cloneable y Comparable.

Permite trabajar con la API de JDBC(Java DataBase Connectivity) como si fueran valores Time de SQL. La clase Time añade el formateo y análisis de operaciones para apoyar la sintaxis de salida de JDBC para los valores de tiempo.

Los componentes de fecha deberían estar establecidos en el formato "zero epoch" (el tiempo, en milisegundos, transcurridos desde el 1 de enero de 1970).

Trabaja con hh:mm:ss

Timestamp

Hereda de la clase Date e implementa las interfaces Serializable, Cloneable y Comparable.

Esta clase permite trabajar con la API de JDBC como si fuese un valor Timestamp de SQL. Timestamp o sello de tiempo es una secuencia de caracteres que denotan la fecha y hora en la cual ocurrió un determinado evento. Esta información es comúnmente presentada en un formato consistente, lo que permite la fácil comparación entre dos diferentes registros y seguimiento de progresos a través del tiempo.

Mejora la precisión añadiendo los nanosegundos y provee de formateo y análisis de operaciones para apoyar la sintaxis de salida de los valores de timestamp en JDBC.

La precisión de Timestamp es calculada por:

- 19, que es el número de caracteres en yyyy-mm-dd hh:mm:ss (año-mes-dia hora:minuto:segundo)
- 20 + s, que es el número de caracteres en yyyy-mm-dd hh:mm:ss[ffff...] y s representa la magnitud dada en Timestamp,su precisión de fracciones de segundo.

Date (Sql)

Hereda de la clase Date(util) e implementa las interfaces Serializable, Cloneable y Comparable.

Esta clase es hija de java.util.Date y es usada para la representación de la fecha en JDBC, es decir, son los campos almacenados en una base de datos cuyo tipo es una fecha. Tiene una precisión de milisegundos y su formato de salida muestra el día, el mes y el año.

TimeZone

Hereda de Object e implementa las interfaces Serializable y Cloneable.

Se trata de una clase que representa zonas horarias y también tiene en cuenta los cambios horarios(las horas que sumamos o restamos en verano o invierno). Se usa en conjunto con la clase Calendar.

Normalmente esta clase se usa con la zona horaria por defecto, es decir, la que está basada en la localización desde la que el programa se está ejecutando. Para ello usamos el método getDefault(). También es posible crear un objeto de TimeZone usando una ID de zona. Por ejemplo, si queremos crear un TimeZone con la zona horaria de Madrid, sería:

```
TimeZone tz = Timezone.getTimeZone("Europe/Madrid");
```

SimpleTimeZone

Hereda de TimeZone e implementa las interfaces Serializable y Cloneable.

Se trata de una subclase contreta de TimeZone que representa una zona horaria para el uso con el calendario Gregoriano. El calendario gregoriano es un calendario originario de Europa, actualmente utilizado de manera oficial en casi todo el mundo.

La clase refleja el cambio horario establecido durante el horario de verano.

A los constructores se les pasa el desplazamiento horario con respecto al GMT que es el tiempo medio de Greenwich. GMT es un estándar de tiempo que originalmente se refería al tiempo solar medio en el Real Observatorio de Greenwich, cerca de Londres.

Cómo especificar el comienzo y fin del verano: se utilizan combinaciones entre el mes, día del mes y día de la semana. El valor del mes está representado con un valor de Calendar. Ej: Calendar.MARCH. El valor del día de la semana está representado por un valor DAY_OF_WEEK de Calendar, como Sunday. Las posibles combinaciones de valores son las siguientes:

- Día exacto del mes: se deja el día de la semana a cero.
- Día de la semana a partir de un día del mes: en el día del mes se pone el día a partir del cual se aplica la regla, y el día de la semana a un valor negativo del campo DAY_OF_WEEK. Por ejemplo, para especificar el segundo domingo de abril, en el mes se pone abril, el día del mes a 8 y el día de la semana a -Domingo.
- Día de la semana antes de un día exacto del mes: se establece el día del mes y el día de la semana en negativo. Por ejemplo, para especificar el miércoles antes del 21 de marzo, en el mes se pone marzo, el día del mes es -21 y el día de la semana es -Miércoles.
- Último día de la semana del mes: se pone el día del mes a -1. Por ejemplo, para especificar el último domingo de octubre, en el mes de pone octubre, en el día de la semana Domingo y en el día del mes -1.

  El momento del día en que el horario de verano se inicia o termina se especifica por un valor en milisegundos.

Calendar

Hereda de Object e implementa las interfaces Serializable, Cloneable y Comparable.

La clase Date no es muy adecuada para operar con las representaciones humanas de tiempo (día, minutos, etc.) Para esto se emplea la clase "Calendar", la cual se puede considerar que "contiene" un Date, pero proporciona métodos convenientes para diversas operaciones.

Se trata de una clase abstracta que provee métodos para convertir entre un instante específico y un conjunto de campos de Calendar como YEAR, MONTH, DAY_OF_MONTH, HOUR y para poder manipularlos.

Ofrece un método estático getInstance para obtener un objeto de uso general de este tipo. Ahora bien, funciona realmente creando una instancia de una subclase. Se le permite llamarlo porque es un método estático. El valor de retorno se refiere a una instancia de la subclase correspondiente, pero el tipo de retorno es Calendar, lo cual está bien, debido a las reglas normales de la herencia.

Provee de campos y métodos para la implementación de un sistema de calendario.

Un objeto Calendar puede producir todos los campos de valores necesarios para implementar el formato de tiempo y fecha para un lenguaje y estilo de calendario particulares (por ejemplo el Japonés-Tradicional). Calendar define el rango de valores que pueden ser devueltos por los campos del calendario.

El campo de calendar ERA puede valer:

- 0: antes de cristo (BC)
- 1: después de cristo (AD)

GregorianCalendar

Hereda de Calendar e implementa las interfaces Serializable, Cloneable y Comparable.

Se trata de una subclase concreta de Calendar que usa el sistema de calendario estándar usado en la mayor parte del mundo. Es una calendario híbrido que soporta tanto el sistema de calendario Gregoriano como el Juliano.

Valores por defecto de los campos:

Campo	Valor por defecto
ERA	AD
YEAR	1970
MONTH	JANUARY
DAY_OF_MONTH	1
DAY_OF_WEEK	The first day of week
WEEK_OF_MONTH	0
DAY_OF_WEEK_IN_MONTH	1
AM_PM	AM
HOUR, HOUR_OF_DAY, MINUTE, SECOND, MILLISECOND	0

Format

Hereda de Object e implementa las interfaces Serializable y Cloneable.

Se trata de una clase abstracta para el formateo de información sensible a su situación (geográfica, política o cultural) tales como fechas, mensajes y números.

Esta clase define la interfaz de programación para convertir los objetos en String y viceversa.

DateFormat

Hereda de Format e implementa las interfaces Serializable y Cloneable.

Se trata de una clase abstracta para el formateo de fechas y horas según la región. Por ejemplo, no se representará igual la fecha en España (12/10/2014) que en EEUU(10/12/2014).

Para dar formato a una fecha con la localización actual, se usa uno de los métodos generadores estáticos:

```
myString = DateFormat.getDateInstance().format(myDate);
```

Para dar formato a una fecha con una localización diferente, se llama a getDateInstance():

```
DateFormat df = DateFormat.getDateInstance(DateFormat.LONG, Locale.FRANCE);
```

Se puede controlar la longitud del resultado: corto, medio, largo o completo.

También se puede establecer la zona horaria.

SimpleDateFormat

Hereda de DateFormat e implementa las interfaces Seriealizable y Cloneable.

Es una clase concreta para analizar y dar formato a las fechas por regiones. Permite formatear (fecha → texto), el análisis (texto → fecha), y la normalización. También permite la elección de algunos patrones definidos para el formato de fecha y hora o construir nuestros propios formatos. Su uso es similar al que se le da a su padre, pero con la diferencia de que nos permite construir nuestros propios formatos de fecha. Es decir, las instancias creadas por DateFormat no nos proporciona muchas posibilidades de formato, por ello la clase SimpleDateFormat permite que sus constructores reciban por parámetro un String que especificará un patrón para formatear y parsear.

Patrones para fechas y horas:

Letra	Componente de fecha u hora	Presentación	Ejemplo
G	Designador de era	Texto	AD
y	Año	Año	1993;93
Y	Semana año	Año	2002;02
M	Mes en año	Mes	January; Jan, 01
w	Semana en año	Número	22
W	Semana en mes	Número	3
D	Día en año	Número	150
d	Día en mes	Número	8
F	Día de la semana en mes	Número	4
E	Nombre de día en semana	Texto	Tuesday; Tue
u	Número de día de la semana	Número	2
a	Marca AM/PM	Texto	AM
H	Hora del día(0-23)	Número	2
k	Hora del día(1-24)	Número	24
K	Hora en am/pm(0-11)	Número	3
h	Hora en AM/PM(1-12)	Número	45
m	Minutos de horas	Número	28
s	Segundos de horas	Número	55
S	Milisegundos	Número	288
z	Zona horaria	General time zone	Pacific Standard Time; PST; GMT-08:00
Z	Zona horaria	Zona horaria RFC 822	-0800
X	Zona horaria	Zona horaria ISO8601	-08; -0800; -08:00