

Programación. 1º DAW



Un paseo por la API:
Algunas clases de utilidad de Java

Un paseo por la API: Algunas Clases de Utilidad en Java

- ❑ Clase Object
- ❑ Clase String. Métodos de la clase
- ❑ Clases Double, Integer, Character,...
- ❑ Funciones Matemáticas
- ❑ Fecha y Hora
- ❑ Otras clases del paquete java.util

La clase Object

- ❑ Es la raíz de la jerarquía de clases
- ❑ Métodos:
 - **equals** ([Object](#) obj)
 - ❑ Indica si algún otro objeto es "igual a" un presente.
 - **toString** ()
 - ❑ Devuelve una cadena de la representación del objeto.
 - **notify** ()
 - ❑ Despierta un único hilo que está a la espera en este objeto del monitor.
 - **notifyAll** ()
 - ❑ Despierta todos los hilos que están esperando en este objeto del monitor.
 - **wait** ()
 - ❑ El hilo actual debe de esperar esperar hasta que acabe otro hilo invocado por [notify\(\)](#) o [notifyAll\(\)](#).

La clase String

- ❑ Clase para representar cadenas de caracteres en Java
- ❑ Por ejemplo:

```
String str = "abc";
```

es equivalent to: es equivalente a:

```
char datos [] = ( 'a', 'b', 'c');
```

```
String str = new String (datos);
```

La Clase String

❑ Métodos:

- **charAt** (int index)
 - ❑ Devuelve el carácter especificado en el índice
- **compareTo** ([String](#) anotherString)
 - ❑ Compara dos cadenas Lexicograficamente
- **compareToIgnoreCase** ([String](#) str)
 - ❑ Compara dos cadenas Lexicograficamente, haciendo caso omiso a mayúsculas o minúsculas
- **equals** ([Object](#) anObject)
 - ❑ Compara esta cadena para el objeto especificado.
- **equalsIgnoreCase** ([String](#) anotherString)
 - ❑ Compara dos cadenas, haciendo caso omiso a mayúsculas o minúsculas.
- **indexOf** ([String](#) str)
 - ❑ Devuelve el índice dentro de esta cadena de la primera aparición de la subcadena especificada.
- **length** ()
 - ❑ Devuelve la longitud de esta cadena.
- **replace** (char oldChar, char newChar)
 - ❑ Devuelve una nueva cadena resultante de la sustitución de todas las apariciones de oldChar en esta cadena con newChar

La clase String

- ❑ Métodos:
 - **substring** (int beginIndex, int endIndex)
 - ❑ Devuelve una nueva cadena que es una subcadena de esta cadena.
 - **toCharArray** ()
 - ❑ Convierte esta cadena a una nueva matriz de caracteres.
 - **toLowerCase** ()
 - ❑ Convierte todos los caracteres de String cadena a minúsculas utilizando las reglas de la localización por omisión.
 - **toUpperCase** ()
 - ❑ Convierte todos los caracteres de String cadena a mayúsculas utilizando las reglas de la localización por omisión.
 - **trim** ()
 - ❑ Devuelve una copia de la cadena, con los principales rezagados en blanco y omitido.
 - **valueOf** (Object obj)
 - ❑ Retorna la cadena de la representación Object argumento.

La clase Double

❑ Métodos:

- **compare** (double d1, double d2)
 - ❑ Compara las dos valores double.
- **floatValue** ()
 - ❑ Devuelve el float valor de este Double objeto.
- **intValue** ()
 - ❑ Devuelve el valor de esta Double como un int
- **isNaN** ()
 - ❑ Devuelve true si este Double es un valor no-un-Número (NaN)
- **longValue** ()
 - ❑ Devuelve el valor de esta Double como un long (por el tipo de fundición long)
- **parseDouble** (String s)
 - ❑ Devuelve un nuevo double para inicializar el valor representado por String cadena especificada, como la realizada por `valueOf` método de la clase Double
- **toString** ()
 - ❑ Devuelve una cadena de representación de este Double objeto.
- **valueOf** (String s)
 - ❑ Devuelve un Double objeto la celebración de la double valor representado por el argumento de cadena s

La clase Integer

- ❑ Métodos:
 - **compareTo** ([Integer](#) anotherInteger)
 - ❑ Compara dos objetos Integer numéricamente.
 - **doubleValue** ()
 - ❑ Devuelve el valor de este Integer como un double
 - **equals** ([Object](#) obj)
 - ❑ Compara este objeto para el objeto especificado.
 - **floatValue** ()
 - ❑ Devuelve el valor de este Integer como un float
 - **intValue** ()
 - ❑ Devuelve el valor de este Integer como un int
 - **longValue** ()
 - ❑ Devuelve el valor de este Integer como un long
 - **parseInt** ([String](#) s)
 - ❑ Analiza la cadena como un argumento firmado decimal entero.
 - **toString** ()
 - ❑ Devuelve String cadena que representa este objeto Integer 's valor.
 - **valueOf** ([String](#) s)
 - ❑ Devuelve un Integer objeto de explotación el valor de String cadena especificada.

La clase Character

❑ Métodos:

- **static int** getNumericValue (**char** c);
 - ❑ Devuelve el valor int que el carácter especificado representa
- **static boolean** isDigit (**char** c);
 - ❑ Determina si el carácter especificado es un dígito
- **static boolean** isLetter (**char** c);
 - ❑ Determina si el carácter especificado es una letra
- **static boolean** isLetterOrDigit (**char** c);
 - ❑ Determina si el carácter especificado es una letra o dígito
- **static boolean** isLowerCase (**char** c);
 - ❑ Determina si el carácter especificado es un carácter en minúscula.
- **static boolean** isUpperCase (**char** c);
 - ❑ Determina si el carácter especificado es un carácter en mayúscula.
- **static char** toLowerCase (**char** c);
 - ❑ Convierte el argumento de caracteres en minúsculas
- **static char** toUpperCase (**char** c);
 - ❑ Convierte el argumento de caracteres en mayúsculas

Funciones Matemáticas

□ Clase Math

- La clase Math contiene métodos para realizar operaciones numéricas básicas, como el elemental exponencial, logaritmo, raíz cuadrada, y funciones trigonométricas
- Propiedades:
 - E :Devuelve el valor del número e.
 - PI:Devuelve el valor del número pi.
- Métodos:
 - abs (tipoDato a)
 - Devuelve el valor absoluto de un valor.
 - asin (double a)
 - Devuelve el arco seno de un ángulo.
 - atan (double a)
 - Devuelve el arco tangente de un ángulo.
 - cos (double a)
 - Devuelve el coseno trigonométricas de un ángulo.
 - sin (double a)
 - Devuelve el seno trigonométricas de un ángulo.
 - tan (double a)
 - Devuelve la tangente trigonométricas de un ángulo.

Funciones Matemáticas

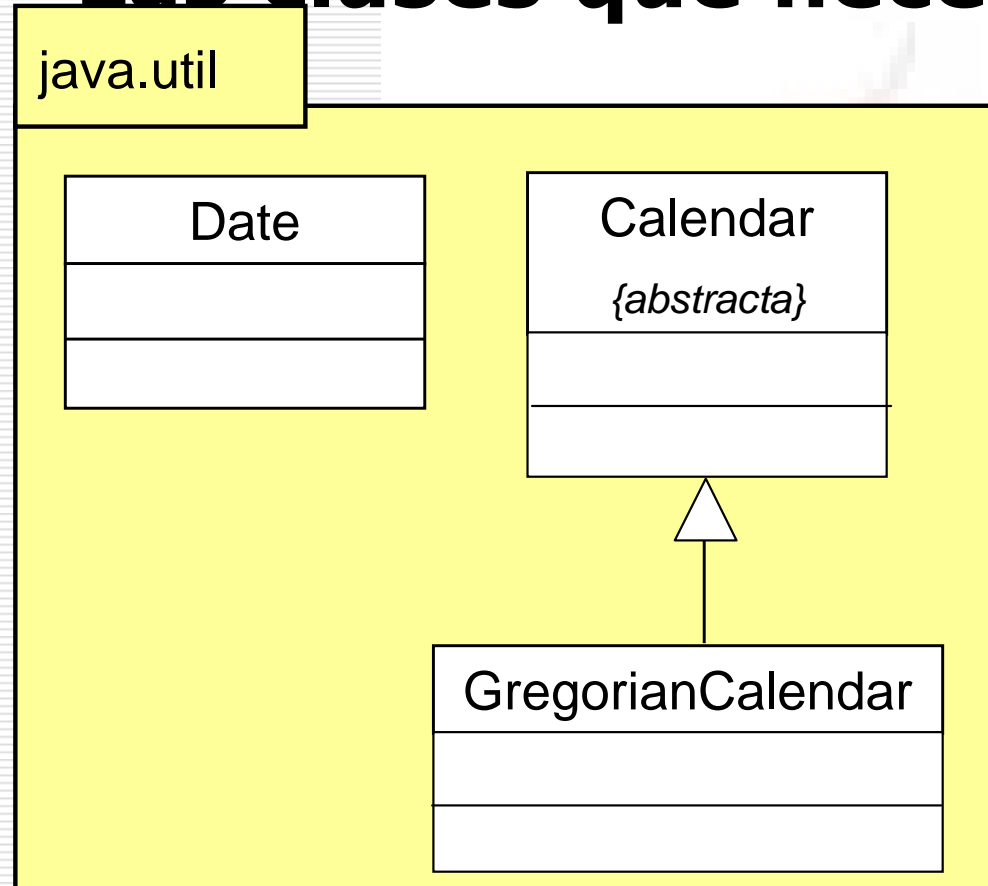
□ Clase Math

■ Métodos:

- exp (double a)
 - Devuelve el número de e elevado a la potencia de un double valor.
- ceil (double a)
 - Devuelve el menor (más cercano al infinito negativo) valor que no es menor que el argumento y es igual a un entero matemático.
- floor (double a)
 - Devuelve el valor más grande (más cercano al infinito positivo) que no es mayor que el argumento y es igual a un entero matemático.
- max (tipoDato a, mismotipoDato b)
 - Devuelve el mayor de dos valores.
- min (tipoDato a, mismotipoDato b)
 - Devuelve el menor de dos valores.
- random ()
 - Devuelve un valor double con un signo positivo, igual o superior 0.0 e inferior a 1.0
- round (double a)
 - Devuelve el argumento redondeado.
- sqrt (double a)
 - Devuelve la raíz cuadrada de un valor.

Fecha y Hora

□ Las clases que necesitamos son:



Date: representa un instante de tiempo específico con precisión de milisegundos.

Calendar: es una colección de propiedades y métodos que facilitan el trabajo con instantes de tiempo.

GregorianCalendar: implementa los métodos abstractos heredados de `Calendar` tomando como referencia el calendario gregoriano (usado en casi todo el mundo)

Fecha y Hora

□ **Date:**

- Constructor sin parámetros. Ejemplo:
 - `Date fechaActual = new Date();`
- Métodos útiles:
 - Para comparar fechas:
 - `+ boolean before (Date instante);` // Devuelve cierto si el objeto *actual* representa una fecha anterior a la del parámetro *instante*
 - `+ boolean after (Date instante);` // Devuelve cierto si el objeto *actual* representa una fecha posterior a la del parámetro *instante*
 - `+ boolean equals (Date instante);` // Devuelve cierto si el objeto *actual* representa el mismo instante temporal (en milisegundos) que el parámetro *instante*
- Casi el resto de las operaciones de la clase Date están desaconsejadas (*deprecated*) y son remitidas a otros métodos de la clase Calendar.

Fecha y Hora

□ **Calendar:**

- Al ser una clase abstracta no podemos crear un objeto de la clase Calendar, sino que tendremos que hacerlo a través de `GregorianCalendar`.

- `GregorianCalendar ahoraCal = new GregorianCalendar();`

O bien:

- `Calendar ahoraCal = new GregorianCalendar();`

- Propiedades públicas y estáticas de Calendar:

- **YEAR:** Año.

- **MONTH:** Mes, entre 0 y 11

- **DAY_OF_MONTH:** Día del mes.

- **DAY_OF_WEEK:** Día de la semana entre 1 (SUNDAY) y 7 (SATURDAY).

- **HOUR:** Hora antes o después del medio día (en intervalos de 12 horas).

- **HOUR_OF_DAY:** Lo hora absoluta del día (en intervalos de 24 horas).

- **MINUTE:** El minuto dentro de la hora.

- **SECOND:** El segundo dentro del minuto.

- ... muchos más

Fecha y Hora

- Métodos útiles:

- Para consultar valores de una fecha:

- + int get (int campo);

- Ejemplo:

- int dia, mes, anio;

- GregorianCalendar ahoraCal = GregorianCalendar ();

- anio = ahoraCal.get(Calendar.YEAR);

- mes = ahoraCal.get(Calendar.MONTH);

- dia = ahoraCal.get(Calendar.DAY_OF_MONTH);

- System.out.println("Hoy es "+dia+"/"+mes+"/"+anio);

- Para establecer valores de una fecha:

- + void set (int anio, int mes, int dia);

- + void set (int anio, int mes, int dia, int hora, int min, int seg);

- + void set (int campo, int nuevoValor);

Fecha y Hora

- Para sumar o restar tiempo a una fecha:
 - + void add (int campo, int cantidad);
 - Ejemplo:

```
GregorianCalendar calen = GregorianCalendar ();  
calen.set(Calendar.YEAR, -1);  
System.out.println("Año pasado = "  
+calen.get(Calendar.YEAR));
```
- Para comparar instantes temporales (igual que Date):
 - + boolean before (Calendar instante);
 - + boolean after (Calendar instante);
 - + boolean equals (Calendar instante);

Otras clases del Paquete Java.util

- ❑ *Clase DateFormat*
- ❑ *Clase SimpleDateFormat*
- ❑ *Clase TimeZone*
- ❑ *Clase SimpleTimeZone*

Copyright ©

Para la confección de parte de esta documentación se ha utilizado material con derechos reservados del Copyright del autor **Enrique José Royo Sánchez**, autorizando este su uso a **Antonio Blázquez Pérez** como material didáctico en el IES Polígono Sur de Sevilla

Copyright © Enrique José Royo Sánchez, 2009

Reservados todos los derechos. Queda rigurosamente prohibida, sin la autorización escrita de los titulares del "Copyright", bajo las sanciones establecidas en las leyes, la reproducción parcial o total de esta obra por cualquier medio o procedimiento, incluidos la reprografía y el tratamiento informático, así como la distribución de ejemplares mediante alquiler o préstamo públicos