Printf

Se ha añadido un intérprete para el formateo de texto al estilo ***printf*** del lenguaje *C*. La sintaxis es muy parecida aunque tiene algunas peculiaridades propias de *Java*.

```
// en Java:

System.out.println( “x:” + x + “ y:” + y );

// en C

System.out.printf( “x: %d y: %d”, x, y );
```

La clase principal es java.util.Formatter. Para hacer más sencilla la programación se ha dotado a clases como java.io.PrintStream, java.io.PrintWriter, String, de los métodos:

format(String format, Object... args)
format(Locale l, String format, Object... args)

para acceder al formateo directamente, sin tener que crear un objeto java.util.Formatter.

También puede utilizar String.format para construir un String usando un esquema de formateo.

Su principal utilidad está en hacer que las salidas por consola sean mucho más legibles.
También para escribir en ficheros de columnas fijas,.

printf es un método que se echaba en falta pero ahora tenemos disponible en Java 5.

http://java.sun.com/j2se/1.5.0/docs/api/java/util/Formatter.html#syntax

Tipos

Conversión de tipos en Formatter

Conversion symbol	Description
%%	Escape sequence to allow printing of % in a String.
%a, %A	Formats the value as a floating-point number in exponential notation, using base-16 for the decimal part, and base-10 for the exponent part. Arguments must be Float, Double, or BigDecimal.
%b, %B	Formats the value as either "true" or "false" (or "TRUE" or "FALSE", for %B). For boolean values, this works as expected. For all other values, any non-null value is "true", while null values are "false".
%c, %C	Formats the value supplied as a single character. Supplied value must be a Byte,Short, Character, or Integer.
%d	Formats the value as a base-10 integer. Arguments must be Byte, Short,Integer, Long, or BigInteger.
%e, %E	Formats the value as a base-10 floating-point number, using exponential notation. Arguments must be Float, Double, or BigDecimal.
%f	Formats the value as a floating-point number in base-10, without exponential notation. Arguments must be Float, Double, or BigDecimal.
%g, %G	Formats the value as a base-10 floating point number, with no more than 6 significant digits (if precision is not supplied). Arguments must be Float, Double, or BigDecimal.
%h, %H	Formats the value as a hexadecimal representation of the value's hashcode.
%n	Outputs the line separator for the platform.
%o	Formats the value as a base-8 octal integer. Arguments must be Byte, Short,Integer, Long, or BigInteger.
%s, %S	Formats the value supplied as a String, usually through calling toString( ) on the object.
%t, %T	The prefix for all date/time conversions. All date/time types listed in Table 9-2) requires a Date, Calendar, or Long argument. Note that the t or T determines uppercase/lowercase, rather than the case of the letter following the t/T.
%x, %X	Formats the value as a base-16 hexadecimal integer. Arguments must be Byte,Short, Integer, Long, or BigInteger.

%[argument][flags][width][.precision]type

Precisión

```
System.out.printf("Balance: $%.2f", balance);
```

Daría como salida un real con dos decimales, por ejemplo 2510.00
Para %e, %E, %f, la precisión por defecto es 6.

Width

El número mínimo de caracteres que

```
System.out.printf("Balance: $%6.2f", balance);
```

El balance aparecerá con al menos seis dígitos en total.

Flags

Son caracteres no numéricos que aparecen justo antes de las indicaciones de width y precision.

Format flags

Flag	**Description**
`-`	Indicates that the formatted value should be left-justified, based on `width` .
`#`	Indicates that the formatted output should appear in alternate form. For `%o`, this means a leading `0`. For `%x` and `%X`, output will include a leading `0x` (`0X`). For `%s` and `%S`, the flag is passed on to the object's `formatTo( )` method.
`+`	Indicates that numeric output should always include a sign (`+` or `-`).
‘ ‘	The space value (which is hard to show in a book) indicates that non-negative values should be prefixed with a space. This is generally used for alignment with negative numbers.
`(`	Indicates that negative numbers should appear in parentheses.
`0`	Indicates that numeric values should be padded on the left.
`,`	Indicates that the locale-specific grouping character should be used for numeric values.

Argumentos

Printf() realiza cada conversión con los argumentos pasados al método, siguiendo el orden en que aparecen.

También puede referirse a uno por su posición con la sintaxis [arg-number]$.

Nos referimos al segundo argumento como %2$

```java
package com.mp.java5;
import java.util.Calendar;
import java.util.Date;

public class PruebaPrintf {
    public static void main(String[] args) {
        String cadena = "Metodología";
        int integer = 24;
        double real = 12*Math.PI;

        System.out.printf("Pruebas con String:\n");
        System.out.printf("Una cadena----------------------------------------|%s|\n", cadena);
        System.out.printf("Una cadena en 15 espacios-------------------------|%15s|\n", cadena);
        System.out.printf("Una cadena en 15 espacios justificada a la izquierda---|%-15s|\n", cadena);
        System.out.printf("Una cadena con 5 caracteres-----------------------|%.5s|\n", cadena);
        System.out.printf("Una cadena en un espacio de 5 con 7 caracteres---------|%5.7s|\n", cadena);
        System.out.printf("\n");

        System.out.printf("Pruebas con integer:\n");
        System.out.printf("Un entero-----------------------------|%d|\n", integer);
        System.out.printf("Un entero como octal------------------|%o|\n", integer);
        System.out.printf("Un entero como hex--------------------|%x|\n", integer);
        System.out.printf("\n");

        System.out.printf("Pruebas con double:\n");
        System.out.printf("Un double-------------------------------|%f|\n", real);
        System.out.printf("Un double en notación cientifica--------|%e|\n", real);
        System.out.printf("Un double con percisión 2---------------|%.2f|\n", real);
        System.out.printf("Un double con percisión 10--------------|%.10f|\n", real);
        System.out.printf("Un double en hexadecimal----------------|%a|\n", real);
        System.out.printf("\n");

        Date hoy = Calendar.getInstance().getTime();
        System.out.printf("Pruebas con Date:\n");
        System.out.printf("Un objeto Date como un string--------------|%s|\n", hoy);
        System.out.printf("Objeto Date formateado---------------------|%tc|\n", hoy);
        System.out.printf("Objeto Date formateado como ISO8602--------|%tF|\n", hoy);
        System.out.printf("Hora en formato 12-------------------------|%tr|\n", hoy);
        System.out.printf("Hora en formato 24-------------------------|%tR|\n", hoy);
        System.out.printf("Hora en formato 24 y segundos--------------|%tT|\n", hoy);
        System.out.printf("Fecha--------------------------------------|%tD|\n", hoy);
        System.out.printf("\n");

        System.out.printf("Pruebas con multiples argumentos:\n");
        System.out.printf("%1$s %2$d, %1$s %3$d, %1$s %4$d\n", "etiqueta:", 1, 2, 3);
        System.out.printf("Fecha dd-mm-yyyy-------------------------|%1$td-%1$tm-%1$tY|\n", hoy);
    }
}
```

```
Pruebas con String:
Una cadena------------------------------------------|Metodología|
Una cadena en 15 espacios---------------------------|    Metodología|
Una cadena en 15 espacios justificada a la izquierda---|Metodología    |
Una cadena con 5 caracteres-------------------------|Metod|
Una cadena en un espacio de 5 con 7 caracteres---------|Metodol|

Pruebas con integer:
Un entero-----------------------------|24|
Un entero como octal------------------|30|
Un entero como hex--------------------|18|

Pruebas con double:
Un double-----------------------------|37,699112|
Un double en notación cientifica--------|3.769911e+01|
Un double con percisión 2---------------|37,70|
Un double con percisión 10--------------|37,6991118431|
Un double en hexadecimal----------------|0x1.2d97c7f3321d2p5|

Pruebas con Date:
Un objeto Date como un string--------------|Thu Mar 23 19:32:41 CET 2006|
Objeto Date formateado---------------------|jue mar 23 19:32:41 CET 2006|
Objeto Date formateado como ISO8602--------|2006-03-23|
Hora en formato 12-------------------------|07:32:41 PM|
Hora en formato 24-------------------------|19:32|
Hora en formato 24 y segundos--------------|19:32:41|
Fecha--------------------------------------|03/23/06|

Pruebas con multiples argumentos:
etiqueta: 1, etiqueta: 2, etiqueta: 3
Fecha dd-mm-yyyy---------------------------|23-03-2006|
```