

Tipos de clases:

1 - Utilitaria:

- Tiene todos los métodos static.
- No se pueden construir objetos, por tanto, no existen métodos constructores.
- Los métodos se usan:
Clase.método()).

2 - No Utilitaria:

- Método constructor (el nombre es el mismo que el de la clase).
- Los métodos se usan:
Objeto.método()).

Clase Scanner - No Utilitaria:

Esta clase nos permite obtener la entrada de datos primitivos.

1 - Constructor:

- Scanner(System.in)

2 - Métodos:

public int .nextInt() → método que permite leer un número entero.

public double .nextDouble() → método que permite leer un número decimal.

public String .nextLine() → método que permite leer una cadena de caracteres.

public char .nextLine().charAt(0) → método que permite leer un carácter.

public boolean .hasNextInt() → método que devuelve si el número introducido es entero o no.

public boolean .hasNextDouble() → método que devuelve si el número introducido es decimal.

Clase String - No Utilitaria:

Esta clase sirve para manejar cadenas de caracteres.

1 - Constructores:

public String() → inicializa una cadena de caracteres vacía.

public String(char[] value) → es una función que pasa un array de caracteres y devuelve un String.

2 - Métodos:

public char .charAt(int index) → método que busca un carácter en concreto dentro de una cadena, busca la posición que se le pasa por parámetro.

public int .length() → método que devuelve un entero con la longitud de la cadena de caracteres.

public String .toUpperCase() → método que convierte toda la cadena en mayúsculas.

public String .toLowerCase() → método que convierte toda la cadena en minúsculas.

public boolean .equals(String cadena2) → método que compara dos cadenas y devuelve el resultado.

public boolean .equalsIgnoreCase(String cadena2) → método que compara dos cadenas (ignorando mayúsculas y minúsculas) y devuelve el resultado.

public int .indexOf(char carácter o String subcadena) → Devuelve la primera posición del carácter que se está pasando por parámetro, devuelve un valor negativo en caso de no encontrar el carácter pasado por parámetro.

public int .lastIndexOf(char carácter o String subcadena) → Igual que el método anterior, pero devuelve la última posición del carácter pasado por parámetro.

public String .substring(int desde_incluido,int hasta_no_incluido) → Devuelve una subcadena comprendida entre dos posiciones.

public String .replace(char oldChar, char newChar) → Reemplaza el carácter antiguo por un carácter nuevo (ambos pasados por parámetro).

public String [] .split(String separador (" ")) → devuelve un array de Strings (con las subcadenas) eliminando el separador introducido por parámetro de la cadena.

public boolean .contains ("String") → devuelve booleano diciendo si está presente el carácter/subcadena en la cadena original.

Clase Math - Utilitaria

Genera números.

1 - Métodos:

static int .abs(int a) → devuelve el valor absoluto de un número.

static int .ceil → método que redondea un número decimal a la alza. Devuelve un entero.

static int .floor → método que redondea un número decimal a la baja. Devuelve un entero.

static int .round → método que redondea a la baja del 0.1 al 0.4 y a la alza del 0.5 al 0.9. Devuelve un entero.

static int .max(int a, int b) → método que devuelve el valor máximo entre dos valores.

static int .min(int a, int b) → método que devuelve el valor mínimo entre dos valores.

static double .pow(double base, double exponente) → método que recibe por parámetros dos números, el primero la base y el segundo el exponente. Si quieres hacer raíz, sería número como base y el exponente sería la inversa del índice de la potencia (1/índice).

static double .sqrt (double num) → por parámetro se introduce el número y hace la raíz cuadrada.

static double .random() → método que devuelve un número aleatorio entre 0.0 y 1.0 sin incluir.

static double .PI → Devuelve número PI.

static double .E → Devuelve número E.

Clase Random - No Utilitaria:

Genera un número aleatorio.

1 - Constructor:

Random()

2 - Métodos:

public int nextInt(int bound) → método que devuelve un número entero entre [0,bound).

Clases Wrappers - Utilitaria:

Encapsula un tipo de dato primitivo. Todas las clases son utilitarias.

Métodos:

1 - Clase Integer:

.MAX_VALUE → Nos permite saber el máximo número representable:

Integer.MAX_VALUE

.MIN_VALUE → Nos permite saber el mínimo número representable:

Integer.MIN_VALUE

static int Integer.parseInt(String s) → Nos permite transformar un valor que se encuentre con un valor String y pasarlo a un valor int.

2 – Clase Double:

.MAX_VALUE → Nos permite saber el máximo número double representable:

Double.MAX_VALUE

.MIN_VALUE → Nos permite saber el mínimo número double representable:

Double.MIN_VALUE

static double

Double.parseDouble(String s) → Nos permite transformar un valor que se encuentre con un valor String y pasarlo a un valor double.

3 – Clase Boolean:

static boolean isNaN(double v) → Comprueba si NO es un número.

static boolean isInfinite(double v) → Comprueba si el resultado de una operación es infinito.