

Práctica 7: Clases de utilidad

(Object - System - Math – Random - String - StringBuffer – StringTokenizer- Wrappers - Colecciones)

1.- Crea un objeto de la clase **Random**, a continuación genera un número aleatorio, y por último busca algún método que te informe del nombre de la clase a la que pertenece un objeto, así como del nombre del paquete.

Averigua cómo se usa y para qué sirve el método `nanoTime` de la clase **System**. ¿Podría servirnos para averiguar cuánto tarda en ejecutarse un determinado método, y así comprobar si es más o menos eficiente?

2.- Prueba el funcionamiento de algunos de los métodos de la clase **Math**, y además:

- Genera un número aleatorio entre 1 y 25 utilizando la clase **Math**.
- Calcula la raíz cúbica del número 13 y de 125.
- Para qué sirve el método `IEEERemainder`.
- Para qué sirve el método `signum`, prueba con distintos valores.
- Calcula la hipotenusa de un triángulo según el Teorema de Pitágoras.

3.- Indica varias formas de poder obtener como tipo primitivo un número dado en una cadena cuyo contenido es "67.235".

4.- Manejo de las clases **String** y **StringBuffer**:

- Crea un objeto de tipo **String** que contenga varias palabras
- Averigua la longitud de la cadena.
- Muestra el carácter que se encuentra en la posición 2 de tu cadena.
- Extrae y almacena en otro objeto del mismo tipo toda la cadena excepto los 2 primeros caracteres subcadena).
- Modifica en la cadena anterior todas las veces que aparezca la vocal 'a' (minúscula) por 'A' (mayúscula).
- Crea un array de caracteres cuyos elementos serán los de la cadena anterior.
- Convierte la subcadena que creaste anteriormente a mayúscula, y que se quede modificada internamente, es decir, no solo a la hora de imprimirla.
- Comprueba si la cadena empieza por la palabra "Hola".
- Busca una subcadena que exista dentro de tu cadena, y otra que no exista (`indexOf`), recordad que en la cadena original la letra 'a' se había cambiado a mayúscula.
- Extrae de tu cadena la segunda palabra y almacénala en una nueva cadena.
- A partir de un array de caracteres crea un objeto de tipo cadena.
- Muestra el valor Unicode de cada uno de los caracteres de la cadena anterior (`codePointAt`).
- Crea un objeto de tipo **StringBuffer**, y prueba diferentes formas de concatenación con el método `Append`.
- A partir de un array de tipo byte que contiene números de un dígito, y con la ayuda del método `reverse` de la clase **StringBuffer**, invierte el citado array.

5. A partir de una cadena de entrada con formato “dd/mm/aaaa”, descompón la cadena de forma que obtengas tres números uno con el día, otro con el mes, y otro con el año.

A partir de una cadena de entrada con formato “nombre_usuario@gestor_correo”, descompón la cadena de forma que obtengas dos nuevas cadenas, una con el nombre de usuario, y otra con el del gestor de correo.

6. Crear la clase **Carácter** con los dos métodos de clase siguientes:

- `int ordinal(char c)`: devuelve el valor ascii del carácter *c*.
- `char ascii(int i)`: devuelve el carácter correspondiente el valor ascii *i*.

Crear la clase **Cadena** de forma que nos permita almacenar una cadena de caracteres, con al menos la siguiente interface:

- `inviertecadena()`. Invierte la cadena miembro.
- `encriptaCadena()`. Encripta la cadena miembro.
- `desencriptaCadena()`. Desencripta la cadena miembro.
- `visualizaCadena()`. Visualiza la cadena miembro.

Hacer una clase para probar la clase Cadena.

7. Crear la clase **Pila**, a partir de la clase **Vector**, con la siguiente interface: *pop*, *push*, *vacía* y *cima*. Utilizando dicha clase, hacer un programa para evaluar una expresión aritmética escrita en notación postfija e introducida desde teclado. Haz una segunda versión de la clase Pila, a partir de la clase **ArrayList** o **LinkedList**.

8. Con la ayuda de las colecciones, genera una lista de 50 números aleatorios que no estén repetidos. Finalmente muestra la lista en pantalla.

9. Crea un diccionario de términos informáticos y su definición, con la ayuda de las colecciones. El diccionario deberá estar ordenado por clave de forma alfabética. Para probarlo, pide al usuario un término y muéstrale su definición.

10. Crea una colección de libros que esté ordenada por autor, y en caso de estar repetido el autor, por título del libro. Añade varios libros a la colección, y finalmente muestra la colección por pantalla.

11. Investiga sobre el manejo de fechas y horas en Java. Consulta el Anexo I sobre manejo de fechas y horas, prueba los ejemplos que vienen en el mismo.