

## EJERCICIO 4

Utilizando las clases JAVA provistas por la Cátedra “**TClasificador.java**” y “**GeneradorDatosGenerico.java**”, armar un paquete “**UT6**” para contener los programas de prueba de los algoritmos de clasificación.

**Se requiere:**

### PARTE 1:

Estudiar el funcionamiento de la clase “**GeneradorDatosGenerico.java**” y responder las preguntas presentadas en pantalla.

- 1) ¿cómo se genera un vector monótonamente ascendente
- 2) ¿cómo se genera un vector monótonamente descendente?
- 3) ¿cómo se genera un vector con valores aleatorios? ¿pueden existir claves repetidas? ¿cuál es el orden del tiempo de ejecución de este método?
- 4) ¿cuántos elementos contiene el vector de datos generado? ¿cómo se puede modificar esta clase para que la cantidad de elementos del vector sea parametrizable?
- 5) ¿cómo podemos verificar que un conjunto está ordenado? ¿cuál sería el orden del tiempo de ejecución de un algoritmo que lo haga?

### PARTE 2:

La clase “**TClasificador.java**” contiene ya métodos para los algoritmos de inserción directa, shellsort y burbuja. Estos métodos **contienen errores**.

- 1) Declarar un vector estático en el “*main*” con unas pocas (no más de 5) claves enteras, desordenadas
- 2) Probar la ejecución del método “clasificar” del *TClasificador*, para invocar el método Inserción Directa.
- 3) Observar el resultado emitido por consola.
- 4) Encontrar y reparar los eventuales errores en el método “*OrdenarPorInsercion*”
- 5) Volver a ejecutar, y verificar el orden de la salida mediante la ejecución de un método “*estaOrdenado*” que tome como parámetro el vector resultante de la ordenación.
- 6) Probar la ejecución con vectores con datos ascendentes, descendentes y aleatorios. Desarrollar casos de test para verificar el correcto funcionamiento.
- 7) ¿cuál es el tiempo de ejecución para cada tipo de vector (tamaño y orden)?
- 8) Repetir los pasos 1 a 7 para los métodos *Shellsort* y *Burbuja*.

**NOTA: SE SUGIERE UTILIZAR CONJUNTOS DE DATOS PEQUEÑOS PARA REALIZAR LAS PRIMERAS PRUEBAS.**