



UNIVERSIDAD DE MÁLAGA LABORATORIO DE TECNOLOGÍA DE OBJETOS

DEPARTAMENTO DE LENGUAJES Y CONVOCATORIA DE JUNIO DE 2002

CIENCIAS DE LA COMPUTACIÓN ETSII

Apellidos, Nombre: Máquina:

Especialidad: Curso:

Se desea construir un diccionario de anagramas a partir de una lista de palabras. Se dice que un anagrama de una palabra es otra palabra obtenida mediante una permutación de sus letras. Por ejemplo,

saco es un anagrama de *cosa*

mora es un anagrama de *amor* y de *roma*

Una palabra es un anagrama de otra si tienen la misma *signatura*, entendiéndose por *signatura* de una palabra otra palabra resultante de ordenar alfabéticamente las letras de esa palabra. Por ejemplo,

la signatura de *saco* es *acos*

la signatura de *cosa* es *acos*

la signatura de *examen* es *aeemnx*

Un diccionario de anagramas debe hacer corresponder a cada palabra todos sus anagramas.

1) (1.75 ptos.) Crear la clase **Signatura** cuyas instancias mantienen información de una palabra y su signatura, para esta clase se deberá:

- (0.50) poder crear una instancia conocida la palabra;
- (0.25) disponer de métodos para conocer cada una de las variables de estado que contiene una instancia (*getSignatura*, *getPalabra*);
- (0.25) definir un método *mismaSignatura* que determine cuándo dos instancias tienen la misma signatura;

- (0.50) definir todo lo que sea necesario para que los objetos de esta clase puedan ser ordenados por su componente palabra;
- (0.25) definir el método *toString*.

2) (6.25 ptos.) Crear la clase **DicAnagramas** correspondiente a este diccionario, sabiendo que la información que se desea obtener es un listado, ordenado alfabéticamente, con todas las palabras y, por cada una de ellas, todos sus anagramas ordenados.

Por ejemplo, dadas las palabras *cosa*, *lío*, *amor*, *roma*, *olí*, *mora*, *ramo*, *lió* y *saco*, se desea obtener la siguiente información:

amor (mora ramo roma)

cosa (saco)

lió ()

lío (olí)

mora (amor ramo roma)

olí (lío)

ramo (amor mora roma)

roma (amor mora ramo)

saco (cosa)

Obsérvese que sólo analiza las palabras dadas. Así, la palabra caso sería un anagrama de cosa y de saco, sin embargo, no se considera porque no aparece en el listado suministrado. La lista de palabras vendrán dadas en un fichero de texto palabras.txt separadas por espacios en blanco.

Se deberán proporcionar métodos para:

- (1) Crear la estructura vacía correspondiente al diccionario para almacenar objetos **Signatura**.
- (3.25) Crear dos métodos *generaDiccionario* que generen el diccionario proporcionándoles, respectivamente, el nombre (de tipo **String**) del fichero de entrada y el flujo de entrada (de tipo **BufferedReader**).
- (2) Crear dos métodos *presentaDiccionario* para representar la información pedida sobre un dispositivo proporcionando, respectivamente, el nombre (de tipo **String**) del fichero de salida y el flujo de salida (de tipo **PrintWriter**). Lanzar la excepción **AnagramasExcepcion** cuando alguna operación vaya mal.

3) (2ptos.) Definir un método *interactivo* que permita la ejecución interactiva, creando un diccionario de anagramas a partir de palabras recibidas por teclado (finalizar con CTRL-Z al comienzo de la línea) y presentar la información por pantalla tal como se muestra en el ejemplo anterior.

La clase **DicAnagramas** deberá responder a la siguiente prueba, dando como resultado el diccionario de anagramas con el formato indicado anteriormente con el fichero palabras.txt como entrada:

```
import java.io.*;  
  
public class TestAnagramas{  
  
    public static void main (String[] args) {  
  
        // utilizar una de las dos líneas siguientes  
  
        ejemplo();  
  
        // DicAnagramas.interactivo();  
  
    }  
  
    public static void ejemplo() {  
  
        DicAnagramas dic = new DicAnagramas();  
  
        dic.generaDiccionario (palabras.txt);  
  
        PrintWriter pw = new PrintWriter(System.out, true);  
  
        dic.presentaDiccionario(pw);  
  
        dic.presentaDiccionario(salida.txt);  
  
    }  
  
}
```

NOTAS PARA LA REALIZACIÓN DEL EJERCICIO

(Leer detenidamente antes de empezar el ejercicio)

- Borrar todos los ficheros y subdirectorios del directorio **C:\JAVA**
- Debéis copiar los ficheros palabras.txt y TestAnagramas.java desde **F:\USERS\JAVA\EXAMEN** al directorio **C:\JAVA\EXAMEN**.
- El directorio de trabajo deberá ser **C:\JAVA**. El nombre del proyecto deberá ser **Examen**. En definitiva, las clases deben estar ubicadas en **C:\JAVA\EXAMEN**

Al finalizar el ejercicio, el profesor correspondiente pasará por cada puesto. Para cuando lo haga, el directorio **C:\JAVA\EXAMEN** deberá contener:

Las clases: Signatura.java, Anagramas.java, AnagramasExcepcion.java, TestAnagramas.java

Los ficheros: palabras.txt y salida.txt

SÓLO ESTO será salvado por el profesor. Revisar que los nombres coinciden pues cualquier otro fichero no será copiado. Además, el profesor recogerá la hoja del examen debidamente cumplimentada. Es obligatorio entregar esta hoja, aun cuando no se entregue el ejercicio práctico.

No olvidéis identificar los ficheros con:

- **El nombre, especialidad, curso y máquina.**
- **EL EXAMEN DEBE COMPILAR PARA SER CORREGIDO**
- **TODAS LAS PREGUNTAS INDICAN LOS PUNTOS QUE VALEN**
- **PUEDEN CREARSE OTROS MÉTODOS ADEMÁS DE LOS PEDIDOS, SI SE CONSIDERA NECESARIO.**

IMPORTANTE:

- SALVAD LAS CLASES A MENUDO CON OBJETO DE EVITAR CUALQUIER PROBLEMA CON EL SUMINISTRO ELÉCTRICO.

PODÉIS:

- Utilizar la documentación de las APIs de Java.

NO PODÉIS:

- Utilizar apuntes o páginas WEB.
- Intercambiar documentación entre vosotros.
- Utilizar soportes magnéticos.
- Copiar vuestro ejercicio al final del examen. Si alguien desea una copia de su ejercicio, que se pase posteriormente por el despacho del profesor a retirarla.

APAGAD LOS MÓVILES.