

Programación II - 2012/2013 - Universitat Jaume I

Evaluación continua - Módulo 1 - Prácticas

25 de marzo de 2013

La duración máxima de esta prueba es de 20 minutos. No puedes consultar libros ni apuntes.

Ejercicio (10 puntos)

Para representar todas las palabras que aparecen en una edición del *Diccionario de la lengua española* empleamos un vector de cadenas. En este vector, las palabras están ordenadas lexicográficamente¹ de menor a mayor y una misma palabra nunca aparece más de una vez.

Escribe un método estático en Java que tenga como parámetros dos vectores de cadenas que representan dos ediciones del diccionario de la lengua española a las que llamaremos *antigua* y *nueva*. El método debe mostrar en la salida estándar aquellas palabras que han variado entre las dos ediciones del diccionario, es decir, se deben mostrar tanto las palabras obsoletas como las nuevas palabras incorporadas. Las palabras obsoletas son aquellas que aparecían en la edición antigua pero no se han mantenido en la edición nueva. Por otra parte, las nuevas palabras incorporadas son aquellas que aparecen en la nueva edición, pero no existían en la edición antigua. Las palabras mostradas deben aparecer en orden lexicográfico, cada una en una línea y, junto a cada palabra, se debe indicar si se trata de una palabra obsoleta o de una nueva incorporación.

Se exige que tu solución tenga el mejor tiempo de ejecución posible, teniendo en cuenta que los dos vectores están ordenados.

Por ejemplo, si consideramos las versiones reducidas de las dos ediciones del diccionario que se muestran a continuación:

```
antigua --> ["a", "abajo", "abarroto", "abecedario", "abrir", "abroncar", "absoluta", "acá", "acabijo"]
```

```
nueva --> ["a", "abajo", "abecedario", "abertzale", "abrefácil", "abrir", "abs", "absoluta"]
```

el resultado de la ejecución del método debería ser:

```
abarroto (palabra obsoleta)
abertzale (palabra nueva)
abrefácil (palabra nueva)
abroncar (palabra obsoleta)
abs (palabra nueva)
acá (palabra obsoleta)
acabijo (palabra obsoleta)
```

¹Te recordamos que, si *c1* y *c2* son de tipo `String`, `c1.compareTo(c2)` devuelve un número negativo, cero o un número positivo según *c1* sea menor que, igual o mayor que *c2*, respectivamente, de acuerdo con el orden lexicográfico de las cadenas. Por simplicidad, puedes suponer que el orden lexicográfico que establece el método `compareTo` coincide con el orden del diccionario. En la práctica esto no es así, por ejemplo, por la existencia de vocales acentuadas.