

Programación Orientada a Objetos. Práctica 5.3

Ejercicio 1. (proyecto prKWIC, paquete kwic)

El objetivo de esta práctica es realizar un glosario o índice de palabras (*KeyWord In Context*, KWIC) atendiendo a su aparición en un listado de frases, desechando aquéllas que no se consideren significativas. Para ello, contaremos con dos ficheros de entrada. El primero contendrá la relación de palabras no significativas (y que, por lo tanto, no aparecerán en el índice KWIC). El segundo contendrá una relación de frases, a partir de las cuales deberemos obtener el correspondiente índice. Un ejemplo de fichero con palabras no significativas podría contener las siguientes líneas:

```
el la los las un una unos unas
y o
a ante bajo cabe con contra de desde en entre hacia hasta
para por según sin sobre tras
si no
al del
corre toma llama
```

El siguiente listado de títulos de películas podría servir como ejemplo de contenido de un fichero con las frases (una frase por línea) a partir de las cuales hay que construir un índice KWIC.

```
El color del dinero
Color púrpura
Misión imposible
La misión
La rosa púrpura del Cairo
El dinero llama al dinero
La rosa del azafrán
El nombre de la rosa
Toma el dinero y corre
```

El índice que se desea generar debe tener el siguiente aspecto:

```
AZAFRÁN
  La rosa del azafrán
CAIRO
  La rosa púrpura del cairo
COLOR
  Color púrpura
  El color del dinero
DINERO
  El color del dinero
  El dinero llama al dinero
  Toma el dinero y corre
IMPOSIBLE
  Misión imposible
MISIÓN
  La misión
  Misión imposible
NOMBRE
  El nombre de la rosa
PÚRPURA
  Color púrpura
  La rosa púrpura del Cairo
ROSA
  El nombre de la rosa
  La rosa del azafrán
  La rosa púrpura del Cairo
```

Como vemos, el índice está ordenado por palabras significativas y, por cada una de ellas, aparecen todas las frases que la contienen (ordenadas alfabéticamente). Teniendo en cuenta esto, se deberán desarrollar las siguientes clases y aplicaciones.

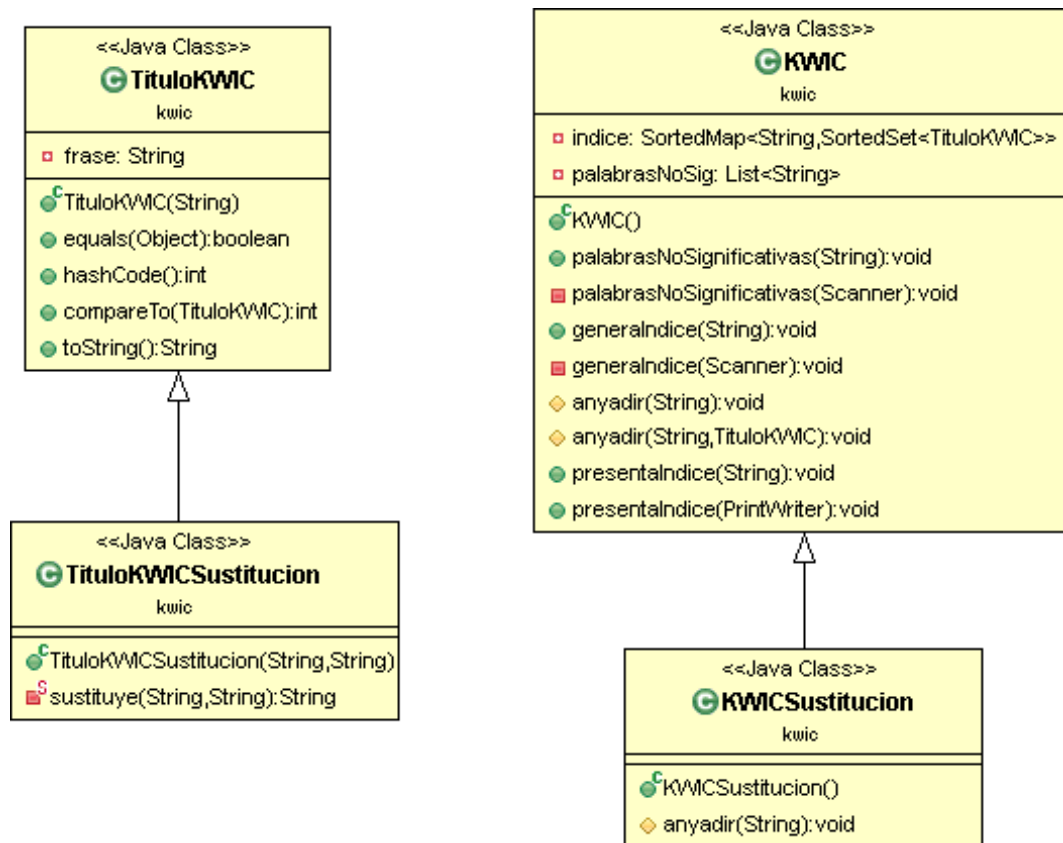


Figura 1: Diagrama de clases UML

Clase TituloKWIC

Definir una clase `TituloKWIC` en el paquete `kwic` que dé envoltura a una frase o título (de tipo `String`), y que permita ordenar y comparar frases independientemente de si éstas contienen caracteres en minúsculas o mayúsculas. El método `toString()` devolverá la frase contenida en la clase. Además, también proporcionará soporte adecuado para la *ordenación natural* de los objetos.

Aplicación PruebaTituloKWIC

Definir una clase distinguida `PruebaTituloKWIC` que haga lo siguiente: (1) cree un array de elementos de tipo `TituloKWIC`; (2) almacene en el array todas las frases de ejemplo proporcionadas anteriormente, y (3) muestre por pantalla el menor y el mayor elemento del array. Para las frases usadas en el ejemplo se debe mostrar por pantalla:

```

Menor = Color púrpura
Mayor = Toma el dinero y corre
  
```

Clase KWIC

Definir una clase `KWIC` en el paquete `kwic` que incluya los métodos necesarios para:

- Construir un objeto de la clase `KWIC` que cree el índice y la lista de palabras no significativas. Los dos estarán vacíos hasta que no se introduzcan los datos mediante la llamada a alguno de los métodos descritos a continuación.
 - `public KWIC();`
- Leer y almacenar la información de las palabras no significativas.

```

▶ public void palabrasNoSignificativas(String nombreFichero);
▶ private void palabrasNoSignificativas(Scanner sc);

```

- Generar la estructura del índice a partir de un fichero de texto (p.ej. títulos de películas) teniendo en cuenta las palabras no significativas leídas previamente. Para ello, se utilizará el método protegido `anyadir(String)`, al que se le pasa como parámetro una cadena de caracteres correspondiente a un título de película y procesa dicho título obteniendo las palabras que lo componen y añadiendo al índice (correspondencia) las parejas oportunas de *palabras* y *títuloKWIC*, invocando para ello al método `anyadir(String, TituloKWIC)`, considerando que todas las palabras del mismo título compartirán el mismo objeto *TituloKWIC*.

```

▶ public void generaIndice(String nombreFichero);
▶ private void generaIndice(Scanner sc);
▶ protected void anyadir(String frase);
▶ protected void anyadir(String pal, TituloKWIC titulo);

```

- Representar el índice sobre un dispositivo.

```

▶ public void presentaIndice(String nombreFichero);
▶ public void presentaIndice(PrintWriter pw);

```

Como se ve, cada una de las acciones de los dos apartados anteriores se podrán realizar proporcionando bien el nombre del dispositivo (fichero) de entrada o salida (de tipo `String`), o bien el flujo de entrada (de tipo `Scanner`) o salida (de tipo `PrintWriter`).

Aplicación PruebaKWIC

Definir una clase distinguida `PruebaKWIC` que cree un objeto `KWIC`. Usando el método `palabrasNoSignificativas()` deben leerse las palabras no significativas desde el fichero "noClaves.txt". Luego, usando el método `generaIndice()` debe leerse el texto desde el fichero "frases.txt". A continuación se almacenará en el fichero "salida.txt" el índice generado por la clase `KWIC`. El índice debe tener el aspecto mostrado anteriormente. También se debe mostrar el índice por la salida estándar (`new PrintWriter(System.out, true)`).

Clase TituloKWICSustitucion

Definir una clase `TituloKWICSustitucion` en el paquete `kwic`, que se comporte como la clase `TituloKWIC`, considerando que el constructor recibe tanto la frase correspondiente al título, como la palabra que queremos que sea sustituida en el título por *puntos suspensivos*. Así, el constructor deberá sustituir en el título la palabra recibida por los puntos suspensivos (utilizando el método de clase `sustituye()`) y almacenar el nuevo título transformado.

```

▶ public TituloKWICSustitucion(String frase, String pal);
▶ private static String sustituye(String frase, String palabra);

```

Nótese que el método `sustituye()` recibe una frase y una palabra, y devuelve un nuevo `String` con la misma frase, pero sustituyendo todas las apariciones de la palabra por *puntos suspensivos*. Es *método de clase (estático)* para que pueda ser invocado adecuadamente, desde el constructor de la clase, al invocar al constructor de la superclase.

Clase KWICSustitucion

Definir una clase `KWICSustitucion` en el paquete `kwic`, que se comporte como la clase `KWIC` pero que redefina los métodos necesarios para que al representar el índice sobre un dispositivo de salida (salida estándar o un fichero), muestre la salida de forma que, para cada palabra del índice, las frases en las que esa palabra aparece debe mostrarse sustituyendo dicha palabra por "...".

En este caso hay que considerar que, a diferencia de la clase KWIC, no es posible compartir el mismo objeto `TituloKWICSustitucion` entre todas las palabras de la misma frase, ya que cada palabra, distinta, será sustituida en el título por *puntos suspensivos*, dando lugar a diferentes frases para palabras diferentes.

```
► public KWICSustitucion();
► protected void anyadir(String frase);
```

El índice que se debería generar en este caso debe tener el siguiente aspecto:

```
AZAFRÁN
    La rosa del ...
CAIRO
    La rosa púrpura del ...
COLOR
    ... púrpura
    El ... del dinero
DINERO
    El color del ...
    El ... llama al ...
    Toma el ... y corre
IMPOSIBLE
    Misión ...
MISIÓN
    La ...
    ... imposible
NOMBRE
    El ... de la rosa
PÚRPURA
    Color ...
    La rosa ... del Cairo
ROSA
    El nombre de la ...
    La ... del azafrán
    La ... púrpura del Cairo
```

Aplicación EjemploKWIC

La clase KWIC deberá responder a la siguiente prueba situada en el paquete por defecto:

```
import java.io.*;
import kwic.*;
public class EjemploKWIC {
    public static void main(String[] args) {
        try {
            KWIC k = new KWIC();
            k.palabrasNoSignificativas("noClaves.txt");
            k.generaIndice("frases.txt");
            System.out.println("Presenta indice sin aplicar sustitucion de palabras");
            System.out.println("-----");
            PrintWriter pw = new PrintWriter(System.out, true);
            k.presentaIndice(pw);
            k.presentaIndice("salida.txt");
            KWICSustitucion ksust = new KWICSustitucion();
            ksust.palabrasNoSignificativas("noClaves.txt");
            ksust.generaIndice("frases.txt");
            System.out.println("\n\nPresenta indice aplicando sustitucion de palabras");
            System.out.println("-----");
            ksust.presentaIndice(pw);
            ksust.presentaIndice("salida-sust.txt");
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

Fichero noClaves.txt:

el la los las un una unos unas
y o
a ante bajo cabe con contra de desde en entre hacia hasta
para por según sin sobre tras
si no
al del
corre toma llama

Fichero frases.txt:

El color del dinero
Color púrpura
Misión imposible
La misión
La rosa púrpura del Cairo
El dinero llama al dinero
La rosa del azafrán
El nombre de la rosa
Toma el dinero y corre

Nota: al copiar el contenido de los ficheros del documento PDF, las letras con tildes se copian con una codificación **errónea**.