

## (proyecto prCuentaPalabrasSimpleFicheros)

Se va a crear una aplicación para contar el número de veces que aparece cada palabra en un texto dado. Para ello se crearán las clases `PalabraEnTexto`, `ContadorPalabras` y `ContadorPalabrasSig` en el paquete `palabras`.

- a. Crea la clase `PalabraEnTexto` para mantener información de una palabra (`String`) así como del número de veces que aparece en un determinado texto (`int`).
  1. La clase tendrá un constructor en el que se proporciona la palabra. Al construir el objeto, el número de veces que aparece la palabra se considera 1. Además, la palabra se almacenará en mayúscula.
  2. Dos objetos de la clase `PalabraEnTexto` son iguales si coinciden las palabras que contiene. El número de apariciones no se tiene en cuenta.
  3. La representación de un objeto `PalabraEnTexto` debe mostrar la palabra que contiene y el número de veces que aparece.
  4. El método `void incrementa()` incrementa en uno el número de veces que aparece la palabra.
- b. Crea una aplicación (clase distinguida `PruebaPalabraEnTexto`) para probar la clase anterior. En esta aplicación se crean dos objetos de `PalabraEnTexto` con las palabras `gorra` y `Gorra`. Después se incrementa el número de apariciones de la primera en uno, y se muestra por pantalla el contenido de ambos objetos. Por último, se comprueba si ambas palabras son iguales, indicándolo por pantalla. La ejecución de la aplicación producirá la siguiente salida por pantalla:

```
Palabra 1 = GORRA: 2
Palabra 2 = GORRA: 1
Las palabras son iguales
```
- c. Crea la clase `ContadorPalabras` que almacena en un array (array de `PalabraEnTexto`) las palabras que aparecen en un texto. También guardará el número de palabras (`int`) almacenadas en un momento dado.

Si en el transcurso de las operaciones que se hagan con objetos de esta clase, el array llega a tener un tamaño insuficiente, deberá crecer de manera que siempre quepan las palabras que se le proporcionen.

1. La clase dispondrá de dos constructores; el primero de ellos, sin argumentos, que crea el array con un tamaño de 10 (constante `TAM_INICIAL`); el segundo, con un argumento entero, que indicará el tamaño inicial del array. En ambos casos, el número de palabras almacenadas será 0.
2. El método privado `int esta(String pal)` que devuelve la posición en la que se encuentra la palabra que corresponde a `pal` en el array o -1 si no está.
3. El método protegido `void incluye(String pal)` deberá incrementar el número de apariciones de la palabra que corresponda a la cadena `pal` en el contador de palabras si es que ya existía, o incluir una palabra nueva en caso contrario.
4. El método privado `void incluyeTodas(String linea, String del)` permite extraer de `linea` las palabras usando los delimitadores incluidos en `del`. Cada una de las palabras obtenidas se irán acumulando en el contador.
5. El método público `void incluyeTodas(String [] texto, String del)` incluye todas las palabras que se encuentra en el array `texto`. Cada elemento del array será una línea de texto y en cada línea, las palabras se deben separar usando los delimitadores incluidos en `del`.
6. El método público `void incluyeTodasFichero(String nomFich, String del)` incluye todas las palabras que se encuentra en el fichero. Cada elemento del fichero será una línea de texto y en cada línea, las palabras se deben separar usando los delimitadores incluidos en `del`. Este método crea un flujo de entrada (`Scanner`) y llama al método privado

`leerFichero(Scanner sc, String del)` que llevará a cabo la lectura del fichero línea a línea.

7. El método público `PalabraEnTexto encuentra(String pal)` que, dada una cadena de caracteres `pal` que representa una palabra, encuentra la instancia de `PalabraEnTexto` en el array que coincide con ella y la devuelve. Si la palabra no se encuentra en el texto deberá lanzar la excepción `NoSuchElementException`.
8. La clase dispondrá de una representación de los objetos como la que se muestra en el ejemplo final. Usar `StringBuilder` o `StringJoiner` para crear la representación, y obsérvese que, tras la última palabra, no hay coma.
9. La clase además dispondrá de dos métodos públicos, `void presentaPalabras`, que generarán una presentación del índice en el formato:

```
GUERRA: 5
TENÍA: 2
UNA: 2
JARRA: 3
Y: 1
...
```

Uno de los métodos recibirá como parámetro el nombre del dispositivo de salida (fichero, de tipo `String`) donde llevar a cabo la acción y el otro recibirá como parámetro el flujo de salida (de tipo `PrintWriter`) donde llevar a cabo la acción.

- d. Crea una aplicación (clase distinguida `PruebaContadorPalabras`) para probar la clase anterior. En esta aplicación se crea un objeto de la clase `ContadorPalabras` con un array de tamaño 5. Posteriormente se invoca a su método `incluyeTodas` pasándole como parámetros:

- El siguiente array:

```
String [] datos = {
    "Esta es la primera frase del ejemplo",
    "y esta es la segunda frase"};
```

- La siguiente cadena (el espacio en blanco es el único delimitador) :

```
"[ ]"
```

Para terminar, la aplicación mostrará por pantalla el contenido del objeto creado. La ejecución de la aplicación producirá la siguiente salida por pantalla:

```
[ESTA: 2, ES: 2, LA: 2, PRIMERA: 1, FRASE: 2, DEL: 1, EJEMPLO: 1,
Y: 1, SEGUNDA: 1]
```

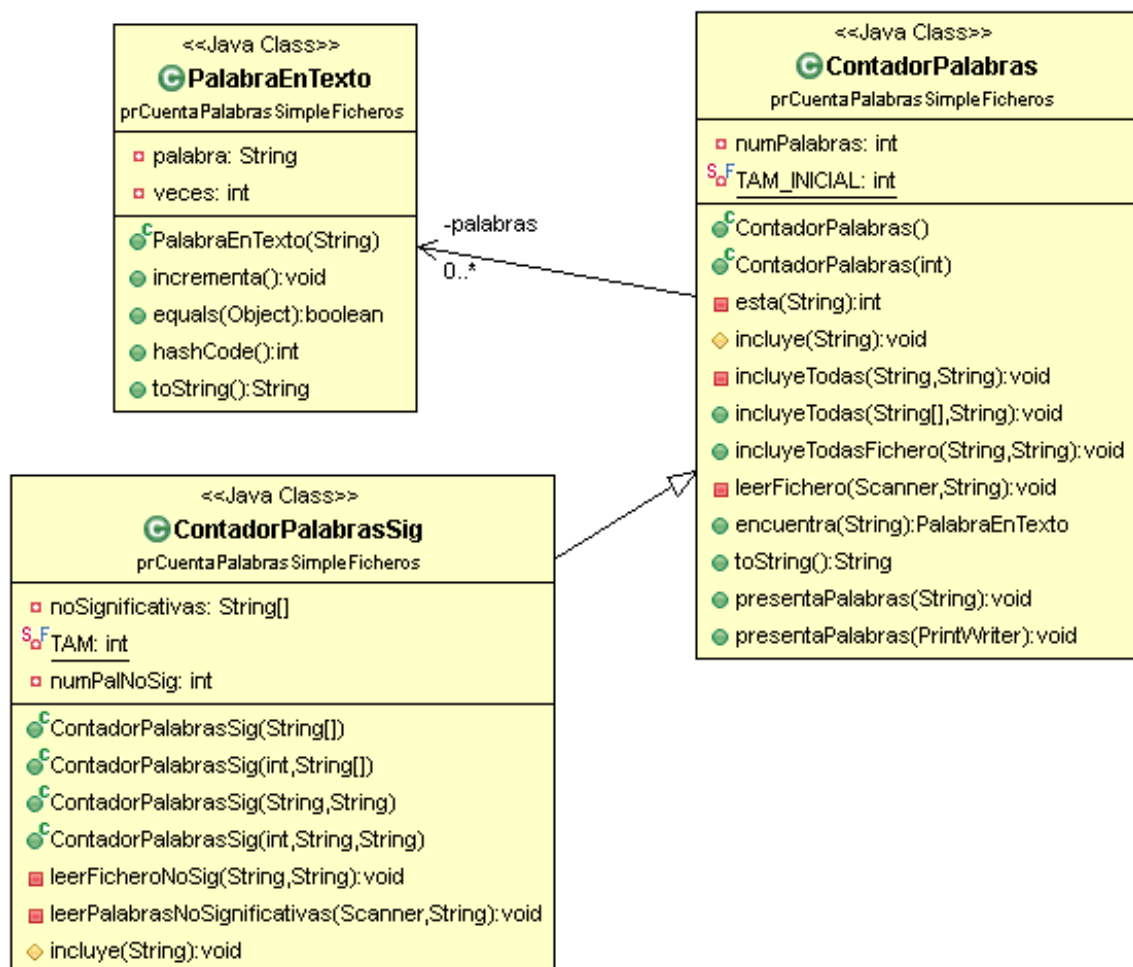
- e. Crea la clase `ContadorPalabrasSig` que representa objetos contadores de palabras que, en los procedimientos de inclusión, no contemplan las palabras consideradas “no significativas”. Para ello, la clase deberá incluir un array de `String` que almacene estas palabras no significativas (se almacenarán en mayúscula), así como el número de palabras almacenadas en él (`int`)

1. Definir dos constructores, `ContadorPalabrasSig(int n, String[] palsNS)` y `ContadorPalabrasSig(String[] palsNS)`. El primero recibe el tamaño inicial (`n`) del array de `PalabraEnTexto` y un array de `String` (`palsNS`) con las palabras no significativas. El segundo sólo recibe este array. Las palabras no significativas que vengan en este array parámetro (`palsNS`) deberán almacenarse en el objeto en mayúsculas (en el array `noSignificativas`). La variable `numPalNoSig` se inicializará al tamaño de estos arrays.
2. Definir dos constructores más, `ContadorPalabrasSig(String filNoSig, String del)` y `ContadorPalabrasSig(int n, String filNoSig, String del)`, para

permitir que la relación de palabras no significativas sea obtenida desde un fichero. Estos constructores no reciben como parámetro un array con las palabras no significativas, sino que reciben un parámetro de tipo `String` con el nombre del fichero de entrada que contendrá la relación de palabras no significativas, y otro parámetro también de tipo `String` que contendrá la cadena con los caracteres delimitadores de dichas palabras en el fichero. En ambos casos, la variable `noSignificativas` se instanciará con un array de `String` de tamaño `TAM`, y la variable `numPalNoSig` se inicializará a 0.

Ambos constructores llamarán al método privado `leerFicheroNoSig(String filNoSig, String del)` que crea un flujo de entrada (`Scanner`) y llama al método privado `leerPalabrasNoSignificativas(Scanner sc, String del)`, al que se le pasa como parámetro el flujo de entrada (`Scanner`) y que llevará a cabo la lectura del fichero palabra a palabra, rellenando el array `noSignificativas` (al igual que en el caso 1, las palabras que se obtengan del fichero se almacenarán en mayúsculas). Si se considera necesario, se pueden utilizar más métodos privados para la implementación de los pedidos.

3. Conseguir que las instancias de la clase `ContadorPalabrasSig` se comporten como las de `ContadorPalabras`, a excepción de que los procedimientos de inclusión de palabras no realicen ninguna acción cuando éstas no sean significativas.



Aquí se presenta un ejemplo de uso más completo de todas las clases y la salida correspondiente:

```

import java.util.NoSuchElementException;
import palabras.*;
import java.io.*;
public class Main {
    public static void main(String [] args) {
        String [] datos = {
            "Guerra tenía una jarra y Parra tenía una perra, ",
            "pero la perra de Parra rompió la jarra de Guerra."
        };
    }
}
  
```



```

        System.out.println("ERROR:" + e.getMessage());
    }
}

```

A continuación, se presenta la salida correspondiente a la clase Main:

```

Por defecto...
[GUERRA: 5, TENÍA: 2, UNA: 2, JARRA: 3, Y: 1, PARRA: 7, PERRA: 6, PERO: 1, LA: 10, DE: 8, ROMPIÓ: 1,
PEGÓ: 1, CON: 3, PORRA: 3, A: 3, OIGA: 1, USTED: 1, BUEN: 1, HOMBRE: 1, POR: 1, QUÉ: 1, HA: 1, PEGADO:
2, PORQUE: 1, SI: 1, NO: 2, HUBIERA: 2, ROTO: 1]

[GUERRA: 5, TENÍA: 2, JARRA: 3, PARRA: 7, PERRA: 6, PERO: 1, ROMPIÓ: 1, PEGÓ: 1, PORRA: 3, OIGA: 1,
USTED: 1, BUEN: 1, HOMBRE: 1, POR: 1, QUÉ: 1, HA: 1, PEGADO: 2, PORQUE: 1, HUBIERA: 2, ROTO: 1]

PARRA: 7
Repetimos la ejecución tomando la E/S desde/a fichero
Por defecto...
No existe la palabra Gorra
[GUERRA: 5, TENÍA: 2, UNA: 2, JARRA: 3, Y: 1, PARRA: 7, PERRA: 6, PERO: 1, LA: 10, DE: 8, ROMPIÓ: 1,
PEGÓ: 1, CON: 3, PORRA: 3, A: 3, OIGA: 1, USTED: 1, BUEN: 1, HOMBRE: 1, POR: 1, QUÉ: 1, HA: 1, PEGADO:
2, PORQUE: 1, SI: 1, NO: 2, HUBIERA: 2, ROTO: 1]

[GUERRA: 5, TENÍA: 2, JARRA: 3, PARRA: 7, PERRA: 6, PERO: 1, ROMPIÓ: 1, PEGÓ: 1, PORRA: 3, OIGA: 1,
USTED: 1, BUEN: 1, HOMBRE: 1, POR: 1, QUÉ: 1, HA: 1, PEGADO: 2, PORQUE: 1, HUBIERA: 2, ROTO: 1]

GUERRA: 5
TENÍA: 2
UNA: 2
JARRA: 3
Y: 1
PARRA: 7
PERRA: 6
PERO: 1
LA: 10
DE: 8
ROMPIÓ: 1
PEGÓ: 1
CON: 3
PORRA: 3
A: 3
OIGA: 1
USTED: 1
BUEN: 1
HOMBRE: 1
POR: 1
QUÉ: 1
HA: 1
PEGADO: 2
PORQUE: 1
SI: 1
NO: 2
HUBIERA: 2
ROTO: 1

GUERRA: 5
TENÍA: 2
JARRA: 3
PARRA: 7
PERRA: 6
PERO: 1
ROMPIÓ: 1
PEGÓ: 1
PORRA: 3
OIGA: 1
USTED: 1
BUEN: 1
HOMBRE: 1
POR: 1
QUÉ: 1
HA: 1
PEGADO: 2
PORQUE: 1
HUBIERA: 2
ROTO: 1

```