POO PRÁCTICA 5.1

(proyecto `prCuentaPalabrasSimpleColecciones`**)**

Se va a crear una aplicación para contar el número de veces que aparece cada palabra en un texto dado, como en la práctica 4.2, pero con algunos cambios. Se redefinirán las clases `PalabraEnTexto`, `ContadorPalabras`, `ContadorPalabrasSig` y `Main` con los siguientes cambios:

a. A la clase `PalabraEnTexto` de la práctica 4.2 se le incorporará un orden natural basado en el orden natural de la cadena de caracteres que almacena (independientemente de que estén en mayúsculas o minúsculas).

b. Crear un aplicación que cree dos objetos `PalabraEnTexto`, con dos cadenas distintas. Incremente tres veces una y dos veces la otra y compruebe cual es la mayor. Probar con otros dos objetos que contengan dos cadenas iguales, una en mayúsculas y otra en minúsculas.

c. La clase `ContadorPalabras`, ahora utilizará una colección (conjunto ordenado) con las palabras que aparecen en un texto (colección de objetos `PalabraEnTexto`) y dispondrá de:
   1. Un constructor sin argumentos que crea la colección de palabras vacía.
   2. El método protegido `void incluye(String pal)` que incrementa el número de apariciones de la palabra correspondiente a la cadena `pal` en el contador de palabras, si es que ya existe, o incluye una palabra nueva en caso contrario.
   3. El método privado `void incluyeTodas(String linea, String del)` que permite extraer de `linea` las palabras usando los delimitadores incluidos en `del`. Cada una de las palabras obtenidas se va incluyendo en el contador de palabras, creando una nueva, si no existe, o incrementando su contador, si ya existe en la colección.
   4. El método público `void incluyeTodas(String[] texto, String del)` que incluye todas las palabras que se encuentran 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`.
   5. El método público `void incluyeTodasFichero(String nomFich, String del)` que 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 aparecerán separadas por alguno/s de los delimitadores incluidos en `del`. Este método crea un flujo de entrada (`Scanner`) y llama al método privado `void leerFichero(Scanner sc, String del)` que lleva a cabo la lectura del fichero línea a línea.
   6. El método público `PalabraEnTexto encuentra(String pal)` que, dada una cadena de caracteres `pal` que representa una palabra, encuentra la instancia de `PalabraEnTexto`, en la colección de palabras, que coincide con ella y la devuelve. Si la palabra no se encuentra deberá lanzar la excepción `NoSuchElementException`.
   7. Un método para la representación textual de los objetos como la que se muestra en el ejemplo final. Usar `StringBuilder` para crear la representación, y obsérvese que, tras la última palabra, no hay coma.
   8. Dos métodos públicos, `void presentaPalabras`, que generarán una presentación del índice según el formato siguiente:

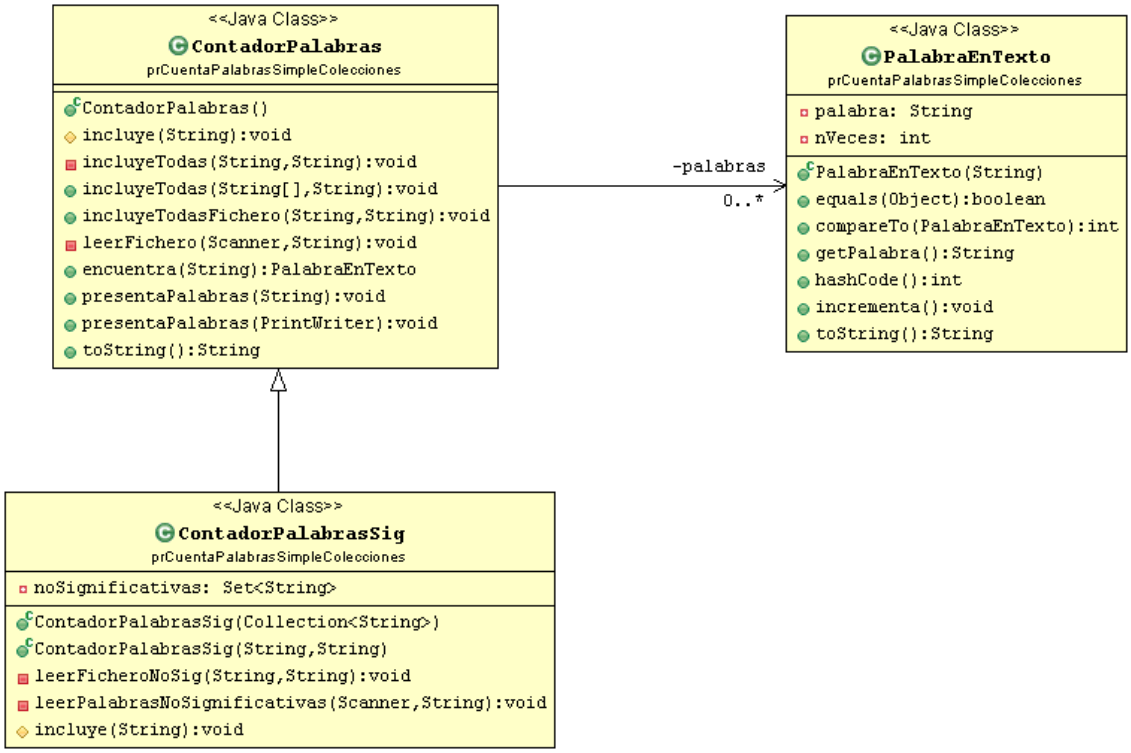
```
GUERRA: 5
TENÍA: 2
UNA: 2
JARRA: 3
Y: 1 ...
```

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

d. La clase `ContadorPalabrasSig`, cuyos objetos, en los procedimientos de inclusión, no contemplan las palabras consideradas no significativas, ahora utiliza una colección de `String` para almacenar estas palabras no significativas que deberán guardarse en mayúsculas y dispone de:
   1. Un constructor, `ContadorPalabrasSig(Collection<String> palsNS)` que recibe una colección con las palabras no significativas y crea un contador de palabras vacío.
   2. Otro constructor que permite obtener la relación de palabras no significativas desde un fichero, `ContadorPalabrasSig(String fichNoSig, String del)`, el primer argumento es el nombre de un fichero de texto que contiene las palabras no significativas y el segundo una cadena con los delimitadores que separan dichas palabras en el fichero. Ejemplo:

      `Con La A De NO SI y una`

      Este constructor llamará al método privado `void leerFicheroNoSig(String fichNoSig, String sep)` que crea un flujo de entrada (`Scanner`) y llama al método privado `void leerPalabrasNoSignificativas(Scanner sc, String del)`, al que se le pasa como parámetros el flujo de entrada (`Scanner`) y los `delimitadores` y llevará a cabo la lectura del fichero palabra a palabra.
   3. Redefine lo necesario para que los métodos de inclusión de palabras no incluyan palabras no significativas en el contador.



Aquí se presenta un ejemplo de uso de estas clases y la salida correspondiente.

```
import java.util.*;
import java.io.*;
import prCuentaPalabrasSimpleColecciones.*;

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.",
        "Guerra pegó con la porra a la perra de Parra. ",
        "¡Oiga usted buen hombre de Parra! ",
        "Por qué ha pegado con la porra a la perra de Parra.",
        "Porque si la perra de Parra no hubiera roto la jarra de Guerra,",
        "Guerra no hubiera pegado con la porra a la perra de Parra."};

    String delimitadores = "[ .,:;\\-\\!\\¡\\¿\\?]+";

    System.out.println("Creamos un contador de palabras");
    ContadorDePalabras contador = new ContadorDePalabras();
    // Incluimos todas las palabras que hay en datos teniendo en cuenta los delimitadores
    contador.incluyeTodas(datos, delimitadores);
    System.out.println(contador + "\n");
    try {
        System.out.println(contador.encuentra("parra"));
        System.out.println(contador.encuentra("Gorra"));
    } catch (NoSuchElementException e) {
        System.out.println(e.getMessage()+"\n");
    }

    //Repetimos la salida con entrada desde fichero ..............
    System.out.println("Repetimos la ejecución tomando la entrada desde fichero");
    contador = new ContadorDePalabras();
    // Incluimos todas las palabras que hay en datos.txt teniendo en cuenta los separadores
    try{
        contador.incluyeTodasFichero("datos.txt", delimitadores);
        System.out.println(contador + "\n");

        //métodos para presentar por pantalla
        System.out.println("Salida a pantalla: ");
        PrintWriter pw = new PrintWriter(System.out, true);
        contador.presentaPalabras(pw);

        //salida a fichero
        System.out.println("\nSalida a fichero: salida.txt\n");
        contador.presentaPalabras("salida.txt");

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

    // Creamos un contador de palabras significativas ....................................
    String [] noSig = {"A", "CON", "DE", "LA", "NO", "SI", "UNA", "Y"};
    Collection<String> palNS = new HashSet<String>();
    for (String p : noSig){ palNS.add(p); }

    System.out.println("Creamos un fichero de palabras significativas: ");
    ContadorDePalabrasSig contadorSig = new ContadorDePalabrasSig(palNS);
    contadorSig.incluyeTodas(datos, delimitadores);
    System.out.println(contadorSig + "\n");

    //Repetimos la salida con entrada desde fichero ..............
    System.out.println("Repetimos la ejecución tomando las entradas desde fichero");

    // Incluimos todas las palabras que hay en datos.txt y las no significativas de fichNoSig
    try{
        contadorSig = new ContadorDePalabrasSig("fichNoSig.txt", delimitadores);
        contadorSig.incluyeTodasFichero("datos.txt", delimitadores);
        System.out.println(contadorSig + "\n");

        //métodos para presentar por pantalla
        System.out.println("Salida a pantalla:");
        PrintWriter pw = new PrintWriter(System.out, true);
```

```
            contadorSig.presentaPalabras(pw);

            //salida a fichero
            System.out.println("\nSalida a fichero: salidaSig.txt");
            contadorSig.presentaPalabras("salidaSig.txt");
        }catch (IOException e){
            System.out.println("ERROR:"+ e.getMessage());
        }

    }
}
```

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

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

PARRA: 7
No existe la palabra Gorra

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

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

Salida a fichero: salida.txt

Creamos un fichero de palabras significativas:
```

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

Repetimos la ejecución tomando las entradas desde fichero
[BUEN: 1, GUERRA: 5, HA: 1, HOMBRE: 1, HUBIERA: 2, JARRA: 3, OIGA: 1, PARRA: 7, PEGADO: 2, PEGÓ: 1, PERO: 1, PERRA: 6, POR: 1, PORQUE: 1, PORRA: 3, QUÉ: 1, ROMPIÓ: 1, ROTO: 1, TENÍA: 2, USTED: 1]

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

Salida a fichero: salidaSig.txt