Programación Orientada a Objetos. Práctica 5.1

Tema 5. Colecciones

Ejercicio 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:

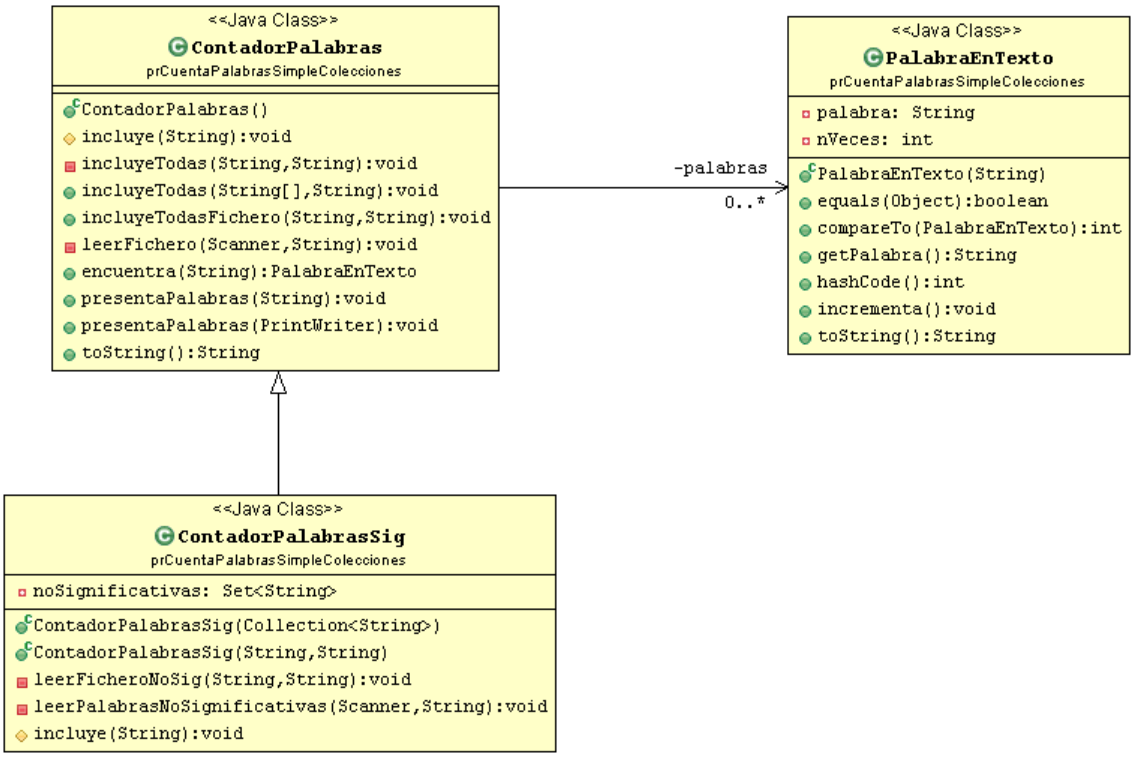

Figura 1: Diagrama de clases UML

Clase `PalabraEnTexto`

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).

Aplicación `TestPalabraEnTexto`

Crear una aplicación que cree dos objetos `PalabraEnTexto`, con dos cadenas distintas. Incremente tres veces a una y dos veces a 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.

Clase `ContadorDePalabras`

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.
   - `public ContadorPalabras();`

2. El siguiente método protegido 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.

   - `protected void incluye(String pal);`

3. El siguiente método privado 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.

   - `private void incluyeTodas(String linea, String del);`

4. El siguiente método público 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`.

   - `public void void incluyeTodas(String[] texto, String del);`

5. El siguiente método público incluye todas las palabras que se encuentran 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`) e invoca al método privado `leerFichero()` que lleva a cabo la lectura del fichero línea a línea.

   - `public void incluyeTodasFichero(String nomFich, String del);`
   - `private void leerFichero(Scanner sc, String del);`

6. El siguiente método público 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`.

   - `public PalabraEnTexto encuentra(String pal);`

7. Un método para la representación textual de los objetos como la que se muestra en el ejemplo final, obsérvese que, tras la última palabra, no hay coma.

8. La clase además dispondrá de los dos siguientes métodos públicos que generarán una presentación del índice en el siguiente formato:

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

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

   - `public void presentaPalabras(String fichero);`
   - `public void presentaPalabras(PrintWriter pw);`

Clase `ContadorPalabrasSig`

La clase `ContadorPalabrasSig`, cuyos objetos, en los procedimientos de inclusión, no incluyen 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 que recibe una colección con las palabras no significativas, que deberán almacenarse en mayúsculas, y crea un contador de palabras vacío.

   - `public ContadorPalabrasSig(Collection<String> palsNS);`

2. Otro constructor que permite obtener la relación de palabras no significativas desde un fichero, donde 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`

   - `public ContadorPalabrasSig(String fichNoSig, String del);`

   Este constructor llamará al método privado `leerFicheroNoSig()` que crea un flujo de entrada (`Scanner`), que a su vez llama al método privado `leerPalabrasNoSignificativas()`, que recibe como parámetro el flujo de entrada (`Scanner`) y que llevará a cabo la lectura del fichero palabra a palabra (al igual que en el caso 1, las palabras que se obtengan del fichero se almacenarán en mayúsculas).

   - `private void leerFicheroNoSig(String filNoSig, String del);`
   - `private void leerPalabrasNoSignificativas(Scanner sc, String del);`

   En el fichero de palabras no significativas, las palabras no significativas se encuentran organizadas en líneas, es decir, puede haber multiples líneas con las palabras, donde cada línea, tiene a su vez palabras que estarán separadas por los delimitadores.

3. Redefine lo necesario para que los métodos de inclusión de palabras no incluyan palabras no significativas en el contador.

Aplicación `Main`

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

```
import prCuentaPalabrasSimpleColecciones.*;
import java.util.Collection;
import java.util.HashSet;
import java.util.NoSuchElementException;
import java.io.PrintWriter;
import java.io.IOException;
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");
        ContadorPalabras contador = new ContadorPalabras();
        // 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 ContadorPalabras();
        // 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: ");
        ContadorPalabrasSig contadorSig = new ContadorPalabrasSig(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 ContadorPalabrasSig("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());
        }
    }
}
```

Los ficheros `datos.txt` y `fichNoSig.txt` deberán ser copiados a la carpeta raíz (carpeta base) del proyecto, en el espacio de trabajo del alumno. **Nota**: al copiar el contenido de los ficheros del documento PDF, las letras con tildes se copian con una codificación **errónea**.

El fichero `datos.txt` contiene la siguiente información:

```
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.
```

El fichero `fichNoSig.txt` contiene la siguiente información:

```
Con La A De NO SI y una
```

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
```