TAD Tablas de Dispersión

Práctica 18
Aplicación de tablas de dispersión

Objetivo

El objetivo de esta práctica es que el alumno ejercite el uso de tablas de dispersión, mediante el desarrollo de una aplicación que permita realizar búsqueda de palabras en un documento.

Descripción general

La práctica consiste en desarrollar un programa para realizar búsquedas eficientes de palabras en un documento. El programa deberá leer un archivo de texto y hacer uso de una tabla de dispersión para almacenar cada palabra así como la(s) línea(s) en la(s) que se encontró.
Una vez procesado el archivo, se presentará un menú con opciones para que el usuario busque palabras o termine el uso del programa. Si la palabra fue encontrada se mostrarán las líneas correspondientes, en otro caso se indicará que la palabra no se encontró en el documento.

Material

El material de esta práctica consta de los siguientes archivos:

- **Nodo.class** clase que implementa los nodos que sirven para crear una lista.
- **InterfazLista.class** interfaz para el tipo abstracto de dato Lista.
- **Lista.class** clase con una posible implementación de la interfaz `InterfazLista`.
- **Lista$Milterador.class** clase interna de la clase `Lista` que implementa un iterador de la lista.
- **InterfazTablaDispersion.class** interfaz para trabajar con tablas de dispersión.
- **TablaDeDispersion.class** clase que implementa el TAD tablas de dispersión con el método de encadenamiento y prueba lineal para resolver colisiones.
- **TablaDeDispersion$milteradorHash.class** clase interna de la clase `TablaDeDispersion` que implementa un iterador de la tabla de dispersión.
- **Palabra.java** clase que implementará la estructura de una palabra.
- **BuscadorDePalabras.java** clase que implementará la búsqueda de palabras en un documento.
- **PruebaBuscadorDePalabras.class** programa para probar la clase `BuscadorDePalabras`.

- **AlanTuring.txt** archivo usado en el programa `PruebaBuscadorDePalabras`.
- Documentación:
    - Nodo.html documentación de la clase `Nodo`.
    - InterfazLista.html documentación de la interfaz `InterfazLista`.
    - Lista.html documentación de la clase `Lista`.
    - InterfazTablaDispersion.html documentación de la interfaz `InterfazTablaDispersion`.
    - TablaDeDispersion.html documentación de la clase `TablaDispersion`.

Desarrollo

1. Descargar los archivos `Nodo.class, InterfazLista.class, Lista.class, Lista$MiIterador.class, InterfazTablaDispersion.class, TablaDeDispersion.class, TablaDeDispersion$miIteradorHash.class, Palabra.java, BuscadorDePalabras.java, PruebaBuscadorDePalabras.class` y `documento.txt` en el directorio donde se va a desarrollar la práctica.

2. Leer la documentación y revisar el material de la práctica. Recordar que la firma de los métodos proporcionados y de los requeridos en el desarrollo de la práctica no podrán ser modificados.

3. En la clase `Palabra` se implementarán los siguientes métodos:

    a. **Constructor** que recibe una cadena de la palabra e inicializa una lista que contendrá el número de las líneas donde se encuentre la palabra.
    b. Métodos de **acceso**.
    c. Método **equals** para determinar si dos Palabras son iguales. Dos Palabras son iguales si las cadenas de cada palabra son iguales.
    d. Método **agregarLinea**(**int**) para agregar un número de línea a la lista de líneas de la palabra.
    e. Método **toString** para convertir una Palabra a una cadena de caracteres, solo devuelve la cadena de la palabra.
    f. Método **hashCode** para obtener un valor de dispersión para una Palabra considerando como llave a la cadena que representa a la palabra.

4. En la clase `BuscadorPalabras` se implementarán los siguientes métodos:

    a. **Constructor** que recibe el nombre del documento. Inicializa una tabla de dispersión y procesa el documento para llenar dicha tabla. Todo el texto será manejado con minúsculas.

b. Método **buscar**(**String**) que recibe una cadena de la palabra por buscar en el documento. Muestra en pantalla el número de las líneas en donde se haya encontrado. Si la palabra se encuentra más de una vez en una misma línea, se muestra el número de la línea tantas veces como haya sido encontrada. En caso de no encontrase, se indica al usuario que la palabra no fue encontrada.

Recordar que se pueden desarrollar los métodos privados que se necesiten.

5. Ejecutar el programa `PruebaBuscadorDePalabras` para probar la implementación realizada. Si el programa funciona adecuadamente se verán los siguientes mensajes:

```
Creando buscador de palabras para el archivo "AlanTuring.txt"

Buscando "Turing"...
Se encontro la palabra "turing" en la(s) linea(s):
1 6 14 18 27 27 35 40 44 50 59 60 62 67 70 71 74 76 82 87 93 100
105 107 110 112 112 113 115 116 118 119 126 131 140 151 154 158 161
163 166 167 171 176 177 179 181 188 190 204 209 214 216 217 217 217
218 229 230 236 237 243 248 250 256 261 265 268 271 272 272 275 276
279 281 284 288 290 291 293 294 296 298 298 300 303 305 307 309

Buscando "computación"...
Se encontro la palabra "computación" en la(s) linea(s):
2 5 6 111 114 164 174 262 263

Buscando "códigos"...
Se encontro la palabra "códigos" en la(s) linea(s):
9 134 137 143 147 150 158 285

Buscando "enigma"...
Se encontro la palabra "enigma" en la(s) linea(s):
10 10 143 150 152 156 285 308

Buscando "algoritmo"...
Se encontro la palabra "algoritmo" en la(s) linea(s):
6 92 93 94 100 113

Buscando "rompecabezas"...
Se encontro la palabra "rompecabezas" en la(s) linea(s):
36 49
```

```
Buscando "global"...
No se encontro la palabra "global".

Buscando "cantar"...
No se encontro la palabra "cantar".

Buscando "búsqueda"...
No se encontro la palabra "búsqueda".
```