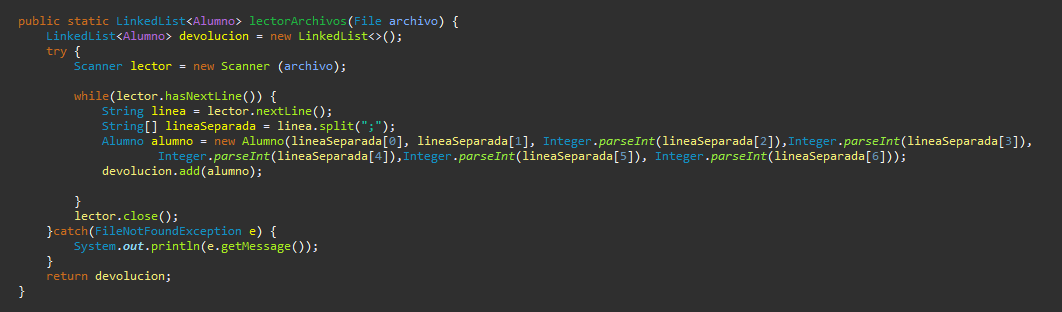
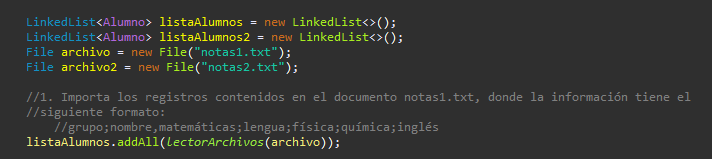
**Paquete: Operaciones LinkedList.**

**1. Importa los registros contenidos en el documento notas 1.txt, donde la información tiene el siguiente formato:**

**grupo;nombre,matemáticas;lengua;física;química;inglés**

Para este apartado se ha cogido el método de Operaciones con ArrayList y se ha modificado, devolviendo una lista y añadiendo esta a la lista generada en el main.



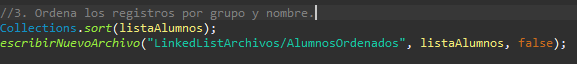


**2. Muestra el número de registros que contiene.**

****

****

**3. Ordena los registros por grupo y nombre.**

****

Se muestran los resultados de la lista ordenada en un documento txt dentro de la carpeta LnkedListArchivos, llamado AlumnosOrdenados.

**4. Añade el registro abajo indicado en la mitad de la lista.**

**1º ESO A;Plaza Gallego, Juan;1;1;3;1;1**



Se genera una variable de tipo Alumno y se instancia con los datos del registro. Se añade a la lista indicando la posición (mitad de la lista). Resultado de apartados 4, 5 ,6 y 7 se muestran en documento de texto dentro de la carpeta LnkedListArchivos, llamado Notas1MasNotas2.

**5. Añade el mismo registro al principio de la lista.**

****

Se añade el mismo registro al principio de la lista con el método addFirst(). Resultado de apartados 4, 5 ,6 y 7 se muestran en documento de texto dentro de la carpeta LnkedListArchivos, llamado Notas1MasNotas2.

**6. Importa los registros contenidos en el documento notas2.txt en una lista auxiliar.**

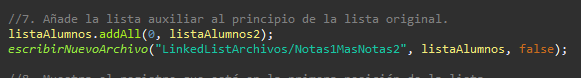
****

****

Añado el contenido de la lista de notas2 a una lista auxiliar creada utilizando el método addAll(), solo se añadirán los registros que no sean iguales que alguno de los que ya existe en la lista. Se usa el método creado de lectorArchivos para leer el archivo y devolverlo en una lista.

Resultado de apartados 4, 5 ,6 y 7 se muestran en documento de texto dentro de la carpeta LnkedListArchivos, llamado Notas1MasNotas2.

**7. Añade la lista auxiliar al principio de la lista original.**

****

Se añade la lista auxiliar a la lista original con el método addAll() indicando la posición donde debe introducir los registros de la lista auxiliar.

Resultado de apartados 4, 5 ,6 y 7 se muestran en documento de texto dentro de la carpeta LnkedListArchivos, llamado Notas1MasNotas2.

**8. Muestra el registro que está en la primera posición de la lista.**

****

****

**9. Muestra el registro que está en la última posición de la lista.**

****

****

**10. Muestra el registro que está en la mitad de la lista.**

****

****

**11. ¿Qué posición ocupa la primera ocurrencia del registro insertado?**

****

****

**12. ¿Qué posición ocupa la última ocurrencia del registro insertado?**

****

****

**13. ¿El siguiente registro está en la lista?**

**1º ESO E;Postigo Vázquez, Salvador;3;6;5;3;4**

Se crea una variable de tipo Alumno y se usa el método contains() para saber si existe o no un registro como el solicitado:

****

****

**14. ¿Y éste?**

**1º ESO A;García Fernández, María;6;5;6;5;5**

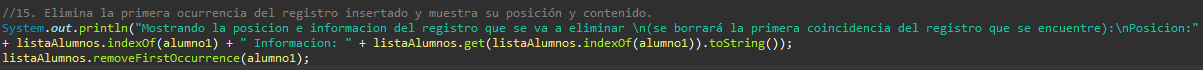
Se crea una variable de tipo Alumno y se usa el método contains() para saber si existe o no un registro como el solicitado:

****

****

**15. Elimina la primera ocurrencia del registro insertado y muestra su posición y contenido.**

Se busca la posicion del alumno insertado con el método indexOf() y se usa para obtener la información con el método get() que devuelve el objeto de la posición introducida. Luego se usa el método removeFirstOccurence() para eliminar el primer registro con esa coincidencia.

****

****

**16. Elimina el registro que ocupa la posición media de la lista mostrando su contenido.**

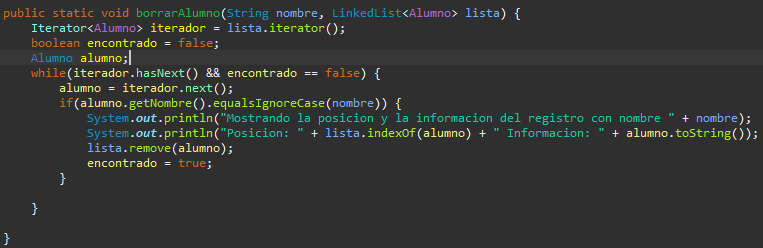
Se usa el método remove() que se le indica la posición del objeto a borrar y devuelve el objeto. Se muestra usando el método toString

****

****

**17. Elimina el registro cuyo nombre es: Carrillo Segura, Félix, indicando su posición y contenido.**

Se genera un método para pasarle un nombre y una lista y que busque alguna coincidencia con ese nombre y lo borre. Mostrará la posición y la información del registro que se va a borrar.



****

****

**18. Elimina el registro cuyo nombre es: Robles Ortiz, Rafael, indicando su posición y contenido.**

Con el método anteriormente mencionado se borra el registro con nombre solicitado.

****

****

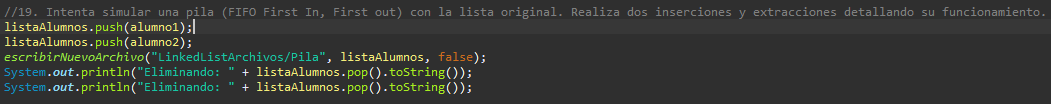
**19. Intenta simular una pila (FIFO First In, First out) con la lista original. Realiza dos inserciones y extracciones detallando su funcionamiento.**

Con los métodos de pila push() y pop() se procede a simular el funcionamiento de una pila. El método push() añade un objeto al final de la lista o pila en este caso.

El método pop() elimina el último registro de la lista o pila en este caso.

Simulan una pila donde el último elemento puesto al final de la pila es el primero en salir.

Por último se muestra la información de los objetos a borrar.

****

**20. Intenta simular una cola (LIFO, Last In, Last Out) con la lista original. Realiza dos inserciones y extracciones detallando su funcionamiento.**

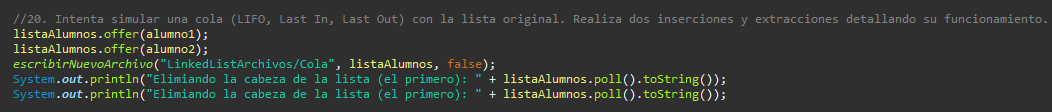
Con los métodos de cola offer() y poll() se procede a simular el funcionamiento de una cola.

El método offer() añade al final de la lista o cola en este caso el objeto indicado.

El método poll() elimina el primer elemento de la lista o cola en este caso.

Simula una cola donde el último objeto se añade al final de la cola y se elimina el objeto de la cabeza o primer objeto de la lista o cola.

Por último se muestra la información de los objetos a borrar.



****