UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Facultad de Ciencias

Estructuras de Datos

Práctica 8: LinkedList

Profesora:
Amparo López Gaona
Ayudante: Adrián Aguilera Moreno
Ayudante de Laboratorio: Kevin Jair Torres Valencia

Objetivo

El objetivo de esta práctica es que los estudiantes se familiaricen con el uso del Java Collections Framework (JCF), en particular con la estructura LinkedList. Se implementará una aplicación que maneja la inscripción y organización de jugadores en los Juegos del Calamar, aplicando distintos métodos para manipular la lista de participantes.

Introducción

En Java, la interfaz de colección `java.util.Collection` y la interfaz de mapa `java.util.Map` son las dos interfaces ”raíz” principales de las clase de colección de Java.

El Java Collections Framework (JCF) es una parte fundamental de la biblioteca estándar de Java, diseñada para proporcionar estructuras de datos eficientes y flexibles para almacenar, manipular y administrar colecciones de objetos.

El paquete de utilidades `java.util` contiene todas las clases e interfaces que requiere el JCF. Contiene una interfaz denominada interfaz iterable que proporciona el `iterador` para iterar a través de todas las colecciones.

Esta interfaz se extiende mediante la interfaz principal `Collection` que actúa como raíz del JCF. Todas las colecciones extienden esta interfaz, extendiendo así las propiedades del `iterador` y los métodos de esta interfaz. La siguiente figura ilustra la jerarquía del marco de trabajo de recopilación.

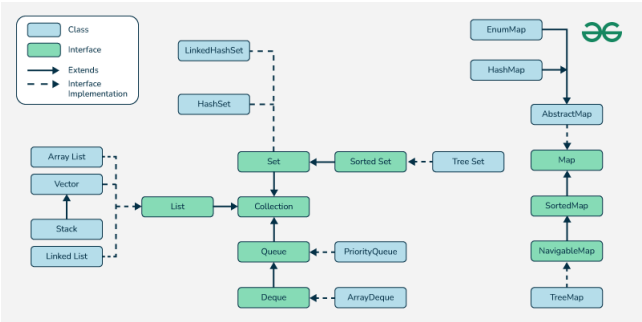

Figure 1: Lista doblemente ligada con cuatro nodos.

El JCF ofrece una arquitectura unificada para manejar grupos de elementos, mejorando el rendimiento y la facilidad de uso en comparación con las estructuras de datos tradicionales basadas en arreglos.

Esta nos incluye interfaces clave como `List`, `Set`, `Queue` y `Map`, cada una con implementaciones específicas para diferentes necesidades. Entre las implementaciones más utilizadas se encuentran `ArrayList`, `LinkedList`, `HashSet` y `HashMap`, cada una con ventajas y desventajas dependiendo del caso de uso.

Dentro del JCF, `LinkedList` es una implementación de la interfaz `List` basada en una estructura de datos de lista doblemente enlazada. A diferencia de `ArrayList`, que usa un arreglo dinámico, `LinkedList` permite inserciones y eliminaciones eficientes en cualquier posición de la lista, ya que no requiere un redimensionamiento o desplazamiento de elementos.

Gracias a su flexibilidad, LinkedList es ideal para escenarios donde las operaciones de inserción y eliminación son frecuentes y la eficiencia en el acceso aleatorio no es una prioridad.

Desarrollo

Los organizadores de los Juegos del Calamar necesitan gestionar la lista de participantes. Para ello, se utilizará una `LinkedList` donde cada elemento representa un jugador con su nombre y número de participante asignado en el orden de inscripción. La aplicación nos permitir:

- Agregar jugadores a la lista (hasta un máximo de 30 participantes),asignándoles un número de jugador secuencial según el orden de inscripción.
- Eliminar jugadores de la lista por nombre o número de participante.
- Invertir la lista utilizando dos métodos distintos:
  - Usando un bucle `for` que modifique la lista en su lugar.
  - Usando un `ListIterator` para recorrer e invertir la lista.
- Reordenar la lista después de invertirla, moviendo al inicio los jugadores cuyos nombres comiencen con una vocal.

Para ello, tienen que implementar los siguientes métodos dentro de la clase `JuegosDelCalamar`:

- `public void agregarJugador(String nombre)`: Agrega un jugador con un número de participante único.
- `public void eliminarJugador(String nombreOId)`: Elimina un jugador de la lista por nombre o número.
- `public void invertirConFor()`: Invierte la lista usando un bucle for.
- `public void invertirConIterador()`: Invierte la lista usando un `ListIterator`.
- `public void reordenarVocales()`: Después de invertir la lista, mueve los nombres que comiencen con vocal al inicio de la lista.
- `public void mostrarJugadores()`: Muestra la lista de jugadores inscritos con su número de participante.

Formato de Entrega

1. Las prácticas se entregarán en parejas.
2. NO incluir los archivos .class dentro de la carpeta.
3. Los archivos de código fuente deben estar documentados.
4. Se pueden discutir y resolver dudas entre los integrantes del grupo. Pero cualquier práctica plagiada total o parcialmente será penalizada con cero para los involucrados.
5. La práctica se debe subir al Github Classroom correspondiente.
6. La entrega en classroom debe contener el link HTTPS y SSH de su repositorio y es lo único que se debe entregar.
7. El horario y día de entrega se acordará en la clase de laboratorio y no deberá sobrepasar 2 clases de laboratorio.