Listas, Pilas y Colas en Java

ArrayList

Cuando se usa un array estático su tamaño no se puede modificar por lo que no sería útil utilizarlo en programas en los que no sabemos cuántos elementos va a tener nuestro array.

Un ArrayList es un array que, aunque internamente está representado por un array estático, puede aumentar y disminuir su tamaño en tiempo de ejecución del programa y el usuario del programa no se preocupa de cómo hacerlo.

https://medium.com/@rajagoyal815/day-1-list-adt-and-arraylist-100daysofcode-7fa32536dfa2

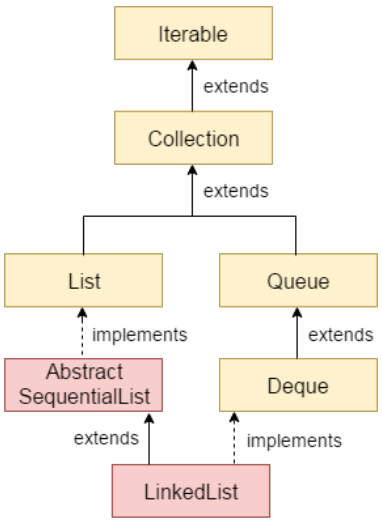
ArrayList viene de que implementa la interfaz de lista con un array, es decir que el adt de esta estructura de datos viene definida en la interfaz de List que se encuentra en:

```
java.util.List
```

https://docs.oracle.com/javase/7/docs/api/java/util/List.html

Linked List

La clase Java LinkedList usa una lista doblemente enlazada para almacenar los elementos. Proporciona una estructura de datos de lista enlazada. Hereda la clase AbstractList e implementa las interfaces List y Deque.



Los puntos importantes sobre Java LinkedList son:

- La clase Java LinkedList puede contener elementos duplicados.
- La clase Java LinkedList mantiene el orden de inserción.
- En la clase Java LinkedList, la manipulación es rápida porque no es necesario realizar cambios.
- La clase Java LinkedList se puede usar como una lista, pila o cola.

Jerarquía de la clase LinkedList

Como se muestra en el diagrama anterior, la clase Java LinkedList extiende la clase AbstractSequentialList e implementa las interfaces List y Deque.

Lista doblemente enlazada

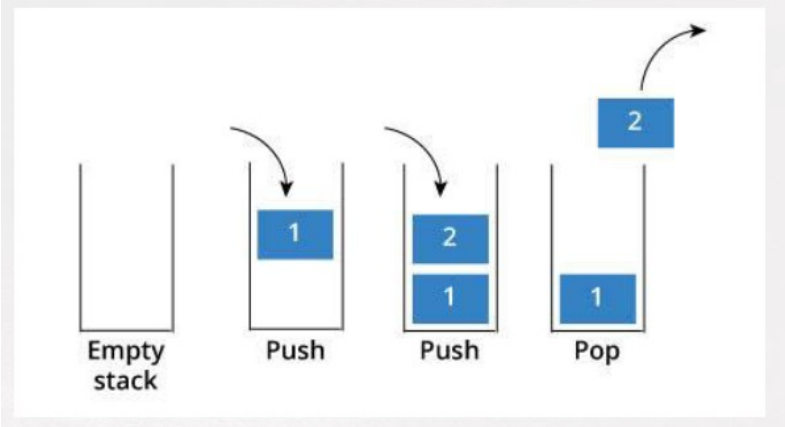
En el caso de una lista doblemente enlazada, **podemos agregar o eliminar elementos de ambos lados**.

fig- doubly linked list

Pilas

Una pila es una estructura de datos que tiene principalmente dos operaciones apilar(push) y desapilar(pop). Es un tipo de estructura que sigue un patrón LIFO (Last in, First out)

Es decir, el último elemento que insertemos (apilar) será el primero en salir (desapilar)



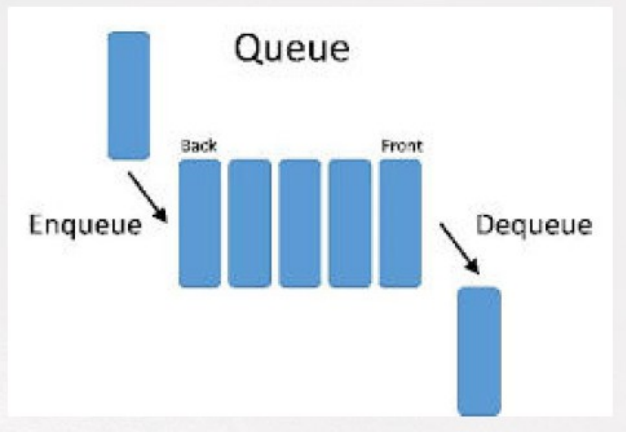
Sus principales usos son:

- Implementar la acción deshacer
- Nos sirve para implementar la búsqueda en profundidad ()
- Evaluar la sintaxis de una calculadora los () y los {}
- Para guardar las llamadas de las funciones recursivas en el orden en que se tienen que ir ejecutando

COLA

Una cola es una estructura de datos que tiene principalmente dos operaciones encolar y desencolar. Es un tipo de estructura que sigue un patrón FIFO (First in, First out)

Es decir el primer elemento que insertemos (encolar) será el primero en salir (desencolar).



Sus principales usos son:

- Transferir datos entre procesos asíncronos
- En un servidor podemos implementar una cola para la venta de entradas
- En una impresora para controlar el orden de los documentos que se van a imprimir.

EJERCICIOS

1) Implemente una pila utilizando una lista como estructura de datos. La pila debe permitir almacenar cualquier tipo de objeto y debe implementar al menos los siguientes métodos:

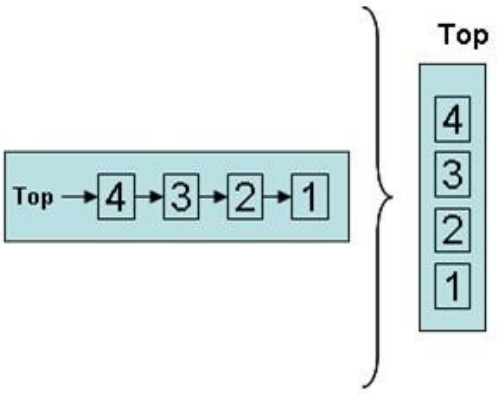
- Inserte un objeto en la pila (push).

- Recuperar un objeto de la pila (pop).

- Recupere el objeto desde la parte superior de la pila, sin extraerlo. (Top())

- Averigua si hay algún objeto en la pila. (isEmpty)

- Devuelve el número de objetos en la pila.



2) Implemente una cola usando una lista vinculada como estructura de datos. La cola debe permitir almacenar cualquier tipo de objeto y debe implementar al menos los siguientes métodos:

- Inserte un objeto en la cola (enqueue).

- Recuperar un objeto de la cola (dequeue).

- Recupere el primer objeto en la cola, sin extraerlo (first).

- Averigüe si hay algún objeto en la cola (isEmpty).

- Devuelve el número de objetos en la cola (tamaño).

3) Crea las clases PilaG y ColaG para cualquier tipo genérico <T> y a continuación, crea un programa que tenga una ColaG de Coches y otro que compruebe si una expresión usando () es correcta utilizando una pila. Por ejemplo:

Reglas:

- Cada vez que nos encontremos (se mete en la pila.
- Cada vez que nos encontremos) se saca de la pila un elemento
- Si al final del proceso, la pila está vacía entonces la cadena de paréntesis es correcta, sino está mal.