

[//https://techblog.vn/linkedList-danh-sach-lien-ket-trong-java](https://techblog.vn/linkedList-danh-sach-lien-ket-trong-java)

1. Tạo mới một danh sách liên kết.

Cú pháp:

```
import java.util.LinkedList;

public class TênClass {

    // ...

    public static void main(String[] args) {

        // khai báo 1 danh sách liên kết có tên là linkedList
        // có kiểu là Integer

        LinkedList<Integer> linkedList = new LinkedList<>();

    }

}
```

2. Hiển thị các phần tử có trong LinkedList.

Ví dụ

```
public static void main(String[] args) {

    LinkedList<Integer> linkedList = new LinkedList<>();

    // thêm các phần tử sử dụng phương thức add()

    linkedList.add(1);

    linkedList.add(0);

    linkedList.add(3);

    linkedList.add(9);

    System.out.println("Các phần tử có trong linkedList là: ");

    for (int i = 0; i < linkedList.size(); i++) {

        System.out.print(linkedList.get(i) + "\t");

    }

}
```

3. Thêm phần tử vào trong danh sách liên kết bằng add().

```
public static void main(String[] args) {
```

```
// khai báo 1 danh sách liên kết đơn có tên là linkedList
// có kiểu là Integer
LinkedList<Integer> linkedList = new LinkedList<>();

// thêm 4 phần tử (nút) có kiểu Integer
linkedList.add(6);
linkedList.add(1);
linkedList.add(10);
linkedList.add(20);
}
```

4. Truy cập phần tử

4.1 Phương thức `get()`.

Công dụng: truy cập đến 1 phần tử bất kỳ trong LinkedList thông qua chỉ số của phần tử đó.

Cú pháp : `get(int index)`;

trong đó index là chỉ số của phần tử đó trong danh sách liên kết ($0 \leq \text{index} < (\text{kích thước của danh sách} - 1)$).

4.2 Phương thức `getFirst()`.

Công dụng: truy cập đến phần tử đầu tiên trong danh sách.

4.3 Phương thức `getLast()`.

Công dụng: truy cập đến phần tử cuối cùng trong danh sách.

5. Cập nhật giá trị của phần tử

Để cập nhật giá trị của phần tử trong LinkedList, Java cung cấp cho chúng ta phương thức `set()`.

Cú pháp:

`set(int index, E element)`;

trong đó index là chỉ số của phần tử cần cập nhật và element là phần tử mới để thay thế.

6. Xóa phần tử

`remove()` Phương thức này được sử dụng trong 2 trường hợp: xóa dựa vào chỉ số của phần tử và xóa trực tiếp phần tử đó (không cần biết đến chỉ số của nó).

`removeFirst()` Xóa phần tử đầu tiên trong danh sách.

`removeLast()`

Xóa phần tử cuối cùng trong danh sách.

`removeAll()`

khác (List, Set,...)

Xóa những phần tử có trong danh sách này mà cũng tồn tại trong 1 Collection