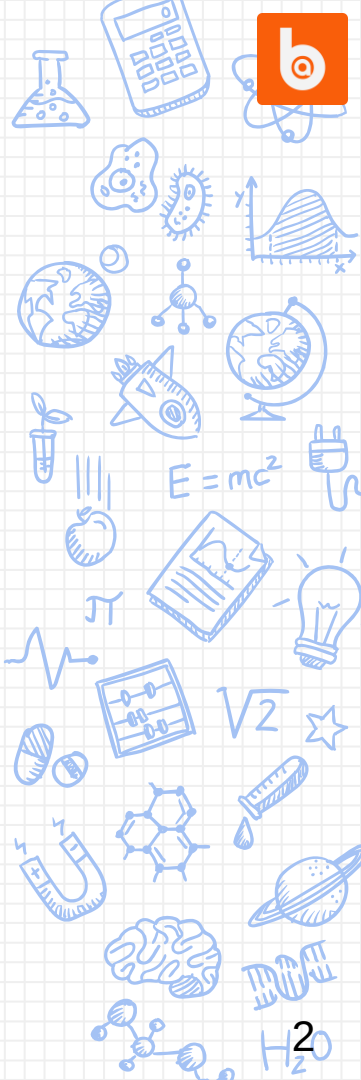
Bài 2.3: Thêm node vào danh sách liên kết đôi

https://braniumacademy.net

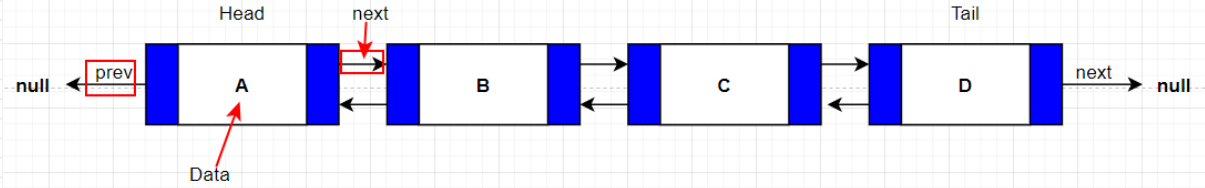
Nội dung bài học

- ❑Tạo danh sách liên kết đôi
- ❑Thêm vào đầu danh sách liên kết đôi
- ❑Thêm vào cuối danh sách liên kết đôi
- ❑Thêm vào sau node có giá trị x
- ❑Duyệt danh sách liên kết đôi



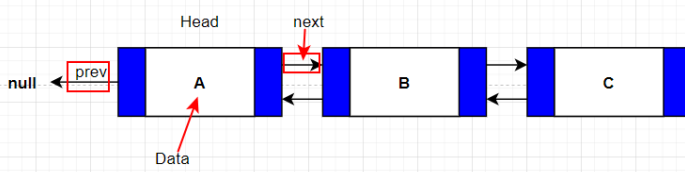
Tạo danh sách liên kết đôi

- Tạo node: inner class tên là Node chứa thuộc tính data, một node next, một node prev.
- Tạo doubly linked list: outer class đặt tên DoublyLinkedList chứa một node head và một node tail.



Tạo danh sách liên kết đôi

❑Triển khai:




```
public class DoublyLinkedList<T> {
    private Node<T> head;
    private Node<T> tail;

    static class Node<T> {
        private T data;
        private Node<T> next;
        private Node<T> prev;

        public Node(T data) {
            this.data = data;
            this.next = null;
            this.prev = null;
        }
    }
}
```

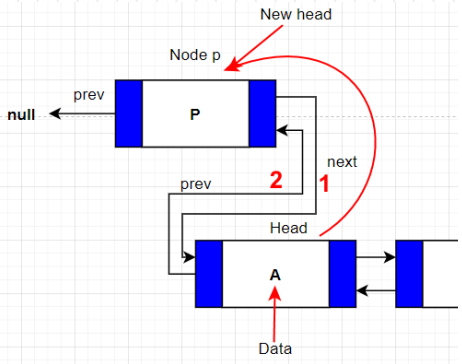
Thêm vào đầu danh sách

❑Giả sử ta muốn chèn node p vào đầu danh sách liên kết. Các bước thực hiện:

- ✓ Kiểm tra xem node head có null hay không, nếu null ta gán luôn head và tail là p.
- ✓ Nếu head khác null:
  B1: gán head cho p.next: p.next = head;
  B2: gán p cho head.prev: head.prev = p;
  B3: cập nhật lại head mới là p: head = p;

Thêm vào đầu danh sách

❑Triển khai:



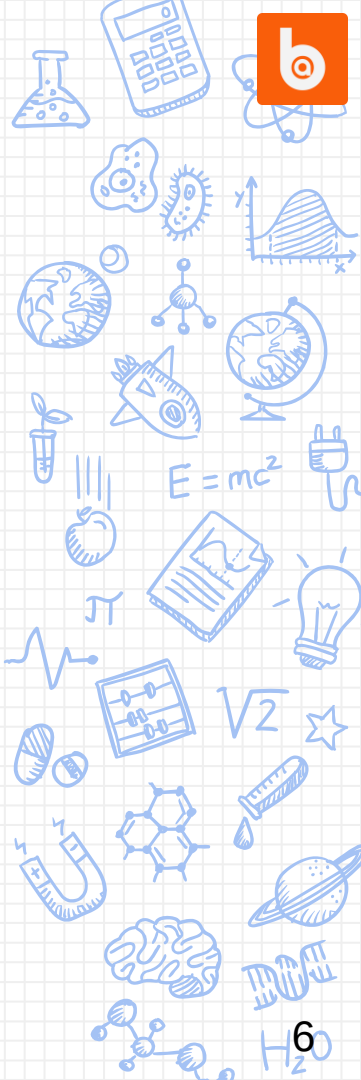

```
public void insertHead(T data) {
    Node<T> p = new Node<>(data);
    if(head == null) {
        head = tail = p;
    } else {
        p.next = head;  // next của p chính là head cũ
        head.prev = p;  // prev của head cũ là p
        head = p;       // cập nhật lại head mới
    }
}
```

Thêm vào cuối

- Giả sử ta muốn thêm node p vào cuối danh sách liên kết hiện tại. Các bước thực hiện:

✓ Kiểm tra xem head có null hay không. Nếu null ta gán p cho head và tail.

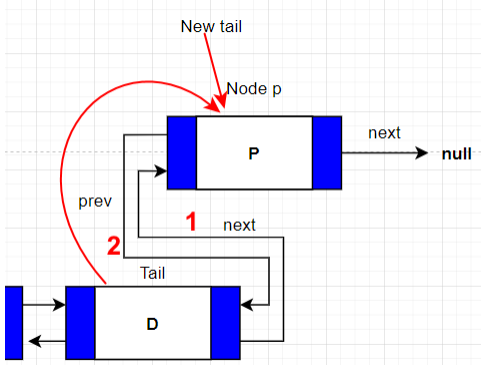
✓ Nếu head không null, ta thực hiện:
B1: gán p cho tail.next: tail.next = p;
B2: gán tail cho p.prev: p.prev = tail;
B3: gán lại tail: tail = p;



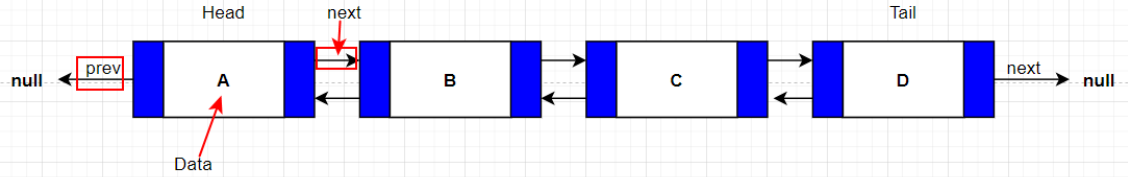
Thêm vào cuối

❑Triển khai code:

```
public void insertTail(T data) {
    Node<T> p = new Node<>(data);
    if(head == null) {
        head = tail = p;
    } else {
        tail.next = p;  // next của tail cũ là p
        p.prev = tail;  // prev của p là tail cũ
        tail = p;       // cập nhật lại tail mới
    }
}
```



Thêm vào sau node có giá trị x



❑Giả sử ta muốn thêm node p sau node có giá trị B:

✓ Tìm node chứa giá trị X == B, gọi là nodeX

✓ Nếu tìm thấy:

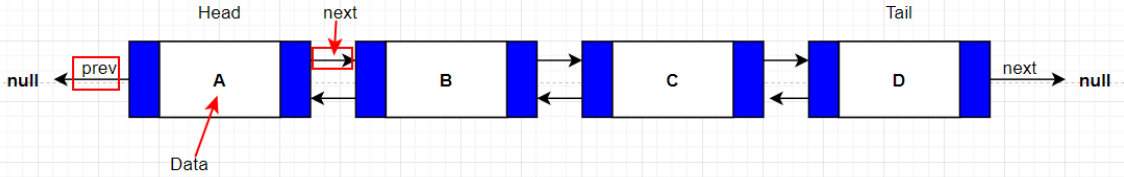
B1: gán nodeX.next cho p.next: p.next = nodeX.next;

B2: gán nodeX cho p.prev: p.prev = nodeX;

B3: gán p cho nodeX.next.prev: nodeX.next.prev = p;

B4: gán p cho nodeX.next: nodeX.next = p;

Thêm vào sau node có giá trị x

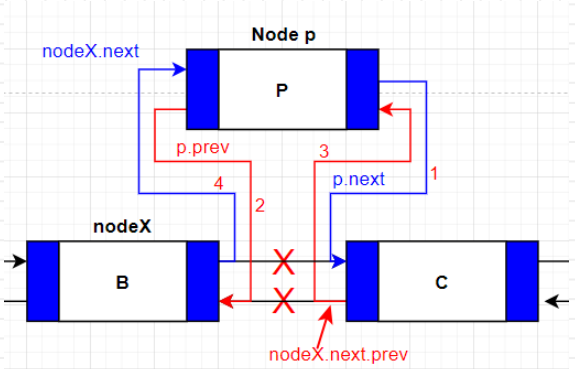


- Giả sử ta muốn thêm node p sau node có giá trị C:
  - ✓ Nếu không tìm thấy ta thông báo ra màn hình và kết thúc.

Thêm vào sau node có giá trị x

- Triển khai code:

```
public void insertAfterX(T data, T x) {
    Node<T> p = new Node<>(data);
    Node<T> nodeX = head;
    while (nodeX != null) {
        if(nodeX.data == x) {
            break;
        }
        nodeX = nodeX.next;
    }
    if(nodeX != null) {
        p.next = nodeX.next; // cập nhật next của p
        p.prev = nodeX;      // cập nhật prev của p
        nodeX.next.prev = p; // cập nhật prev của node sau p
        nodeX.next = p;      // cập nhật next của nodeX
    } else {
        System.out.println("Không tìm thấy node mục tiêu.");
    }
}
```



Duyệt danh sách liên kết

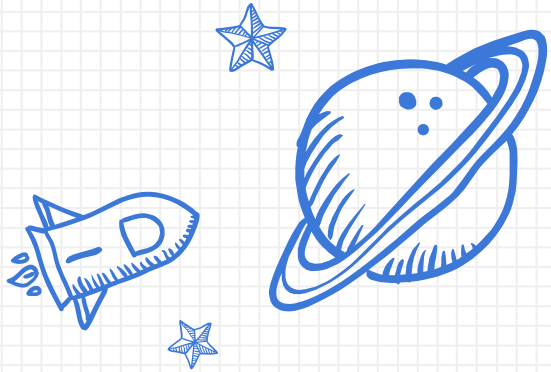
- Có thể duyệt từ đầu hoặc cuối danh sách.
- Các bước thực hiện duyệt từ đầu:
  - ✓ Khai báo node p và khởi tạo bằng head: p = head;
  - ✓ Lặp chừng nào p còn khác null:
    B1: xuất ra giá trị của p: xuất p.data;
    B2: cập nhật p: p = p.next;

```
public void traversalFromHead() {
    for (var node : DoublyLinkedList<...>.Node<...> = head; node != null; node = node.next) {
        System.out.print(node.data + " -> ");
    }
    System.out.println();
}
```

Duyệt danh sách liên kết

- Có thể duyệt từ đầu hoặc cuối danh sách.
- Các bước thực hiện duyệt từ cuối:
  - ✓ Khai báo node p và khởi tạo bằng tail: p = tail;
  - ✓ Lặp chừng nào p còn khác null:
    B1: xuất ra giá trị của p: xuất p.data;
    B2: cập nhật p: p = p.prev;

```
public void traversalFromTail() {
    for (var node : DoublyLinkedList<...>.Node<...> = tail; node != null; node = node.prev) {
        System.out.print(node.data + " -> ");
    }
    System.out.println();
}
```



Tiếp theo

Cập nhật dữ liệu cho node của danh sách liên kết đơn