**package** doublylinkedlist;

**public** **class** DoublyLinkedList {

**static** **class** Node {

**int** data;

Node prev;

Node next;

**public** Node(**int** data) {

**this**.data = data;

prev = **null**;

next = **null**;

}

}

Node head;

**public** **void** insert(**int** data) {

Node newNode = **new** Node(data);

**if** (head == **null**) {

head = newNode;

} **else** {

Node curr = head;

**while** (curr.next != **null**) {

curr = curr.next;

}

curr.next = newNode;

newNode.prev = curr;

}

}

**public** **void** traverseForward() {

Node curr = head;

**if** (head == **null**) {

System.***out***.println("List is empty");

**return**;

}

System.***out***.print("Forward traversal: ");

**while** (curr != **null**) {

System.***out***.print(curr.data + " ");

curr = curr.next;

}

System.***out***.println();

}

**public** **void** traverseBackward() {

Node curr = head;

**if** (head == **null**) {

System.***out***.println("List is empty");

**return**;

}

**while** (curr.next != **null**) {

curr = curr.next;

}

System.***out***.print("Backward traversal: ");

**while** (curr != **null**) {

System.***out***.print(curr.data + " ");

curr = curr.prev;

}

System.***out***.println();

}

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

DoublyLinkedList list = **new** DoublyLinkedList();

list.insert(3);

list.insert(5);

list.insert(8);

list.insert(2);

list.traverseForward();

list.traverseBackward();

}

}

