[HW#Hash] 자료구조론 실습

Hash

2014037901 컴퓨터공학
나 윤 환

```
class Hashtable{

    LinkedList table[];

    Hashtable(){
        this.table = new LinkedList[6];
        for(int i = 0; i < table.length; i++){
            this.table[i] = new LinkedList();
        }
    }
    public int hashFunction(String key){
        return key.length() % table.length;
    }

    public String getValue(String key){

        if(hasValue(key)) return table[hashFunction(key)].getValue(key);
        else return null;
    }

    public void setValue(String key, String value){
        int hash = hashFunction(key);
        table[hash].insert(table[hash].count, key, value);
    }

    public boolean hasValue(String key){
        return table[hashFunction(key)].hasKey(key);
    }

    public void showAll(){
        for(int i = 0; i < table.length; i++){
            if(table[i].count == 0){
                System.out.print("["+i+", NULL]");
            }
            for(int j = 0; j < table[i].count; j++){
                System.out.print("["+i+", ");
                table[i].get_data(j);
            }
        }
        System.out.println();
    }
}
```

```java
class LinkedList{

    private Node head;
    public int count; // 리스트의 노드갯수

    private class Node{

        String key;
        String value;
        Node next;


        Node(String key, String value){    //todo#1
            this.key = key;
            this.value = value;
            this.next = null;
        }

        Node(){
            this.key = null;
            this.value = null;
            this.next = null;
        }

    }

    public LinkedList(){
        this.head = new Node();
        this.count = 0;
    }
```

```java
public void insert(int position, String key, String value){

    Node node = head;

    for(int i = 0; i <= count; i++){
        if(i == position){
            this.count++;
            Node newNode = new Node(key, value);
            newNode.next = node.next;
            node.next = newNode;
            break;
        }
        node = node.next;
    }
}

public void delete(int position){
    Node node = head;

    for(int i = 0; i <= count; i++){
        if(i == position){
            this.count--;
            node.next = node.next.next;
            break;
        }
        node = node.next;
    }
}

public void print_list(){
    // Node node = head.next;

    for(int i = 0; i < count; i++){
        get_data(i);
        System.out.print(", ");
        // node = node.next;
    }
    System.out.println();

}
```

```java
public void get_data(int position){
    Node node = head.next;
    for(int i = 0; i < count; i++){
        if(i == position){
            System.out.print(node.key+", "+node.value+"]");
            break;
        }
        node = node.next;
    }
}

public String getValue(String key){
    // System.out.println("?");
    Node node = head.next;
    while(node.key != key){

        node = node.next;
    }
    return node.value;
}

public boolean hasKey(String key){
    Node node = head.next;
    // System.out.println(count);
    while(node != null){
        // System.out.println(node.key);
        if(node.key == key) return true;
        node = node.next;
    }
    return false;
}
```

```
Appleui-MacBook-Air-4:src nayunhwan$ java Run
444-4444
555-5555
null
[0, Sandra, 333-3333][0, Jaewon, 000-0000][1, NULL][2, No, 123-4567][3, Sam, 222-2222][3, Jee, 456-7890][4, Jonh, 111-1111][4, Lisa, 444-4444][5, Chris, 555-5555]
Appleui-MacBook-Air-4:src nayunhwan$
```