

Data Structure

(Java programming)

Chapter 11.



Contents

- **Dictionary**
- **Practice**

- *Practice*

- **Practice**

LinkedList를 이용한 Dictionary 구현

- Dictionary Class
- DictionaryNode Class
- Main Class

- **Practice**

DictionaryNode class

- key와 value를 멤버변수로 가짐(String)

```
public class DictionaryNode {  
  
    private DictionaryNode prev;  
    private DictionaryNode next;  
    private String key; |  
    private String value;  
}
```

- getter, setter 구현

- **Practice**

Dictionary class

- LinkedList를 이용
- DictionaryNode header, trailer 가짐

```
public class Dictionary {  
  
    private DictionaryNode header;  
    private DictionaryNode trailer;  
}
```

- Practice

Dictionary class

- insert : key와 value를 입력으로 받음
Dictionary에 순차적으로 입력
(key값 중복되게 입력 가능)

```
public void insert(int key, String value){  
    DictionaryNode newNode = new DictionaryNode(trailer.ge  
    this.trailer.getPrev().setNext(newNode);  
    this.trailer.setPrev(newNode);  
}
```

- **Practice**

Dictionary class

- remove : key를 입력으로 받아 해당 key에 해당되는 값 모두 삭제
- find : key를 입력으로 받아 해당 key에 해당되는 첫번째 값의 value return
- findAll : key를 입력으로 받아 해당 key를 가지는 value값 모두 출력

- Practice

Dictionary class

```
public void remove(int key){  
  
    DictionaryNode temp = header.getNext();  
  
    while(temp!=trailer){  
  
        // all key remove  
        if(key == temp.getKey()){  
  
            DictionaryNode tempPrev = temp.getPrev();  
            DictionaryNode tempNext = temp.getNext();  
            tempPrev.setNext(tempNext);  
            tempNext.setPrev(tempPrev);  
        }  
  
        temp = temp.getNext();  
  
    }  
  
}
```

- Practice

Dictionary class

```
public String find(int key){  
  
    DictionaryNode temp = header.getNext();  
    String rtValue = "";  
    boolean findOk = false;  
  
    while(temp!=trailer){  
  
        // first key return  
        if(key == temp.getKey()){  
            rtValue = temp.getValue();  
            findOk = true;  
            break;  
        }  
  
        temp = temp.getNext();  
    }  
  
    if(findOk == false){  
        System.out.println("NOT FOUND!");  
    }  
  
    return rtValue;  
}
```

• Practice

```
public void findAll(int key){
    DictionaryNode temp = header.getNext();
    String rtValue = "";
    boolean findOk = false;
    while(temp!=trailer){
        // record all value
        if(key == temp.getKey()){
            rtValue += "[";
            rtValue += temp.getValue();
            rtValue += "] ";
            findOk = true;
        }
        temp = temp.getNext();
    }
    if(findOk == false){
        System.out.println("NOT FOUND!");
    }
    System.out.println(rtValue);
}
```

```
public String find(int key){
    DictionaryNode temp = header.getNext();
    String rtValue = "";
    boolean findOk = false;
    while(temp!=trailer){
        // first key return
        if(key == temp.getKey()){
            rtValue = temp.getValue();
            findOk = true;
            break;
        }
        temp = temp.getNext();
    }
    if(findOk == false){
        System.out.println("NOT FOUND!");
    }
    return rtValue;
}
```

• Practice

```
Dictionary dic = new Dictionary();  
  
dic.insert("apple", "sweet");  
dic.insert("banana", "sweet");  
dic.insert("apple", "fresh");  
dic.insert("bear", "pretty");  
dic.insert("bear", "smart");  
  
dic.print();  
  
System.out.println(dic.find("bear"));  
|  
dic.findAll("apple");  
dic.findAll("bear");
```

```
terminated main (/) java / application  
key : apple, value : sweet  
key : banana, value : sweet  
key : apple, value : fresh  
key : bear, value : pretty  
key : bear, value : smart  
pretty  
[sweet] [fresh]  
[pretty] [smart]
```

• Submission

e-mail : 2017kudatastructure@gmail.com

Attach methods :

1. Create zip file (java project folder)
2. Modify the file names :
ID_name_datastructure[Classcode].dat
ex) 201173378_이명재_datastructure_chapter11[2403-1].dat
3. e-mail title :
datastructure_name_chapter
ex) datastructure_최수용_chapter11[2404-1]

Classcode

2403-1 [A-1반]

2403-2 [A-2반]

2404-1 [B-1반]

2404-2 [B-2반]

2405 [C반]