

[HashMap구현 실습문제1] 다음과 같은 조건을 만족하는 프로그램을 작성 하시오

Java.util.HashMap의 작동원리를 알기위해 모방한 클래스이다. 구현 클래스다이어그램과 클래스 구조를 참고하여 Snack데이터를 손쉽게 추가, 수정, 삭제등 을 할 수 있는 클래스를 생성하는 프로젝트를 완성하시오.

* Snack클래스는 com.kh.chap03_map.part01_HashMap.model.vo.Snack을 활용

* 프로젝트 명 : 13_Collections_Homework1

1. 구현 클래스 다이어그램 (Class Diagram)

<i>com.kh.practice1.model.vo.HashMap</i>
- table: Entry []
- capacity: int
- size: int
+ HashMap(capacity: int)
+ main(String args[]) : void
+ put(key: String, value: Snack): void
+ get(key: String): Snack
+ containsKey(key: String): boolean
+ remove(key: String): void
+ size(): int

<i>com.kh.practice1.model.vo.Entry</i>
~ key : String
~ value : Snack
~ next : Entry
+ Entry(key : String, value : Snack , next : Entry)

2. 구현 클래스 설명

Package명	Class명	Method	설명
com.kh.practice1. model.vo.	HashMap	+ HashMap(capacity: int)	전달받은 매개변수를 활용하여 capacity값을 초기화하고, capacity만큼의 크기를 가진 객체배열을 table에 할당하는 생성자
		+ put(key: String, value: Snack): void	전달받은 매개변수를 활용하여 table에 값을 저장하는 함수
		+ get(key: String): Snack	전달받은 매개변수를 활용하여 table에 저장된 데이터를 반환하는 함수
		+ containsKey(key: String): boolean	전달받은 매개변수를 활용하여 table에 key값이 존재하는지 확인하는 함수
		+ remove(key: String): void	전달받은 매개변수를 활용하여 table에 key값에 존재하는 데이터를 삭제하는 함수.
		+ size(): int	size필드를 반환하는 함수.
		+ getIndex(key :String) : int	전달받은 매개변수를 활용하여 table에 저장시킬 index값을 반환하는 함수
		+ main(String args[]) : void	실행용 함수.
	Entry	+ Entry(key : String, value : Snack , next : Entry)	전달받은 매개변수를 활용하여 각 필드에 값을 초기화 시키는 생성자

* 작성순서

1. 매개변수 있는 생성자를 통해 HashMap객체를 생성한다.
2. put함수를 이용하여 데이터를 추가한다.
3. get함수를 통해 데이터를 반환받은후 출력한다.
4. containsKey함수를 통해 key값이 있는지 검사한다.
5. HashMap의 사이즈를 측정한다.
6. remove함수 호출하여 데이터를 삭제한다.
7. 삭제 후 get과 size함수를 호출하여 내가 찾던 값이 제거되었는지 확인한다.

3. class 구조

```
public class HashMap {  
  
    public int getIndex(String key) {  
  
        // 매개변수로 전달받은 key값을 hashCode()로 변환시킨후  
  
        // 객체의 총 저장공간 수(capacity)만큼 나눈 나머지 결과값을 반환하는 함수.  
  
    }  
  
    public void put(String key, Snack value) {  
  
        // getIndex함수 호출후 반환값으로 table의 index에 접근  
  
        // 접근시 내부의 값이 null이라면 Entry매개변수 있는 생성자를 이용하여  
  
        // 객체생성후 비어있는 저장공간에 Entry객체 추가하기.  
  
        // ex) 최초 Entry 객체 생성시 : new Entry(key, value , null)  
  
        // 값이 null이 아니라면 , entry에 저장된 key값과 새롭게 추가하려는 entry의  
  
        // key값이 동일한지 검사후 동일하다면 , 중복값이므로 value값만 덮어씌우고  
  
        // key값이 다르다면 현재 entry의 next요소에 entry객체 저장.  
  
        // 만약 next에도 이미 값이 존재하는경우, next가 없는 entry를 찾을때까지 반복  
  
        // 중복값을 제외하고, 값이 새롭게 추가된경우 size 증가  
  
    }  
  
    public Snack get(String key) {  
  
        // getIndex함수 호출후 반환값으로 table의 index에 접근  
  
        // 접근시 내부의 값이 null이라면 null값 반환  
  
        // 객체가 있다면 저장된 객체의 key값과 매개변수로 전달받은 key값 확인  
  
        // 동일하다면 객체에 저장된 value값 반환.  
  
        // 동일하지 않다면 노드의 next 값을 찾아 다시한번 검사 (next가 존재하지  
  
        // 않을때까지 반복). 다음 노드들 중에서 key값이 일치하는 값이 존재한다면  
  
        // 노드 내부의 value값 반환. 일치하는 key값이 없었다면 null값 반환  
  
    }  

```

```
public boolean containsKey(String key) {  
    // getIndex함수 호출후 반환값으로 table의 index에 접근  
    // 접근시 내부의 값이 null이라면 null값 반환  
    // 객체가 있다면 저장된 객체의 key값과 매개변수로 전달받은 key값 확인  
    // 동일하다면 true반환, 일치하지 않는다면 node의 next값을 찾아 일치하는 것을  
    // 찾을때까지 검사. 일치하는 값을 찾았다면 true/ 찾지 못했다다면 false 반환  
}  
  
public void remove(String key) {  
    // getIndex함수 호출후 반환값으로 table의 index에 접근  
    // 접근시 내부의 값이 null이라면 메소드 종료;  
    // 객체가 있다면 저장된 객체의 key값과 매개변수로 전달받은 key값 확인  
    // 동일하다면 현재 Node삭제.  
}  
  
public int size(){  
    //size필드반환  
}  
}
```

```
public static void main(String[] args) {  
  
    HashMap hm = new HashMap (10);  
  
    hm.put("다이제", new Snack("초코맛", 1500));  
  
    hm.put("칸초", new Snack("단맛", 600));  
  
    hm.put("새우깡", new Snack("짠맛", 500));  
  
    hm.put("포테이토칩", new Snack("짠맛", 500));  
  
  
    System.out.println(hm.get("새우깡")); // 출력 : 짠맛, 500원  
  
    hm.put("새우깡", new Snack("매운맛", 700));  
  
    System.out.println(hm.get("새우깡")); // 출력 : 매운맛, 700원  
  
  
    System.out.println(map.containsKey("새우깡 ")); // 출력 : true  
    System.out.println(map.size()); // 출력 : 4  
  
  
    map.remove("포테이토칩");  
  
    System.out.println(map.containsKey("포테이토칩 ")); // 출력 : false  
    System.out.println(map.size()); // 출력 : 3  
  
}
```