

Exercises for Programming Practice 2

Maps are useful in situations where a key can be viewed as a unique identifier for the object. It stores key-value (k, v) pairs. There cannot be duplicate keys. It is convenient to use "HashMap" to handle Maps. Java HashMap is described in the document "HashMap".

After understanding the document "HashMap", please challenge Exercise 15 to 17.

Note:

- ✓ Do not create "module-info.java", when you create a Java Project.
- ✓ Do not set Package name in the window "New Java Class".
- ✓ Do not use the title of exercise for "contents of the program".
Think about "contents of the program" yourself.

The deadline for submitting the programs is 18:00 on June 25th, 2020.

The purpose of Exercise 15 - 17 is complete the Java program "TableMap.java" that is implementation of Map system using Java hashMap.

Use MapTest class to check the behavior of TableMap class.

The classes "MapTest.java" can be obtained from Week 8 of Resource page, "Programming Practice 2" course, manaba+R. Fig. 1 shows the source code of "MapTest.java".

The getData method creates a map of four key-value pairs and returns it. The key of this map is a String, and the value is an Integer.

In the ex16 method, the hashmap obtained by getData method is used as an argument in the constructor of TableMap to initialize the TableMap object "tm". Finally, the method showMap of "tm" is called to print out the contents of hashmap.

In the ex17, after setting hashmap in TableMap object "tm", the length of hashmap is output. The pair of Key "A" and value 7 is set to "tm", and finally the pair of key "K" and value 5 is set to "tm". The hasmap is printed out after each instruction execution.

In the ex18, hashmap is set in TableMap object "tm" like other methods. The value of key "U" is displayed and then the pair of key "U" is deleted. The hashmap is printed out after each command is executed.

```

import java.util.HashMap;

public class MapTest {
    HashMap<String, Integer> hashmap = new HashMap<String, Integer>();

    public static void main(String[] args) {
        MapTest mt = new MapTest();
        mt.ex15();
        mt.ex16();
        mt.ex17();
    }

    public void ex15() {
        hashmap = getData();
        TableMap tm = new TableMap(hashmap);
        tm.showMap();
    }

    public void ex16() {
        hashmap = getData();
        TableMap tm = new TableMap(hashmap);
        System.out.println("the size is " + tm.len());
        tm.showMap();
        tm.set("A", 7);
        tm.showMap();
        tm.set("K", 5);
        tm.showMap();
    }

    public void ex17() {
        hashmap = getData();
        TableMap tm = new TableMap(hashmap);
        tm.showMap();
        System.out.println("the value associated with key U is " + tm.get("U"));
        tm.showMap();
        System.out.println("the value popped with key U is " + tm.pop("U"));
        tm.showMap();
    }

    public HashMap<String, Integer> getData(){
        HashMap<String, Integer> hashmap = new HashMap<String, Integer>();

        hashmap.put("K", 2);
        hashmap.put("B", 4);
        hashmap.put("U", 3);
        hashmap.put("V", 8);
        return hashmap;
    }
}

```

Fig. 1 The source code of "MapTest.java".

The outline of "TableMap.java" is shown in Fig. 2.

```
/* comments */

import java.util.HashMap;
import java.util.Iterator;
import java.util.Set;

public class TableMap {
    HashMap<String, Integer> hashmap = new HashMap<String, Integer>();

    public TableMap(HashMap<String, Integer> hashmap_) {
        hashmap = hashmap_;
    }

    public void showMap() {
        String key;
        // Complete this part
    }

    public int len() {
        return // Complete this part
    }

    public void set(String key, int value) {
        // Complete this part
    }

    public int get(String key) {
        return // Complete this part
    }

    public void pop(String key) {
        return // Complete this part
    }
}
```

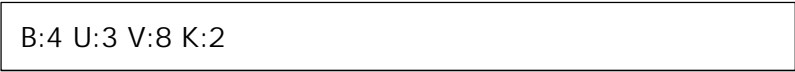
Fig. 2 The outline of "TableMap.java"

Exercise 15 (file name "TableMap.java")

Create the following methods of "TableMap.java":

- ✓ public void showMap() that display the contents of hashmap (HashMap<String, Integer>) with keys and value separated by colon. Use "Iterator" by referring to the example of HashMap. The result of executing method ex15 of MapTest class is shown in Fig. 3. When using the ex15 method, uncomment "//mt.ex15()" in the main method of MapTest.java.

Submit "TableMap.java" that satisfies Exercise 15.



```
B:4 U:3 V:8 K:2
```

Fig. 3 The execution result of method ex15 of "MapTest.java".


Exercise 16 (file name "TableMap.java")

Create the following methods of "TableMap.java":

- ✓ public int len() that returns the size of map.
- ✓ public void set(String key, int value)
If the map contains a mapping for the key, replace value.
Otherwise, associate the value with the key in this map.

Use the method ex16 of MapTest class, and the execution result of ex16 is shown in Fig. 4. When using the ex16 method, uncomment "//mt.ex16()" in the main method of MapTest.java.

Submit "TableMap.java" that satisfies Exercise 16.



```
the size is 4  
B:4 U:3 V:8 K:2  
A:7 B:4 U:3 V:8 K:2  
A:7 B:4 U:3 V:8 K:5
```

Fig. 4 The execution result of method ex16 of "MapTest.java".

Exercise 17 (file name "TableMap.java")

Create the following methods of "HeapPriorityQueue.java":

- ✓ public int get(String key) that returns the value to which the specified key is mapped, but does not removed it.
- ✓ public int pop(String key) that removes the mapping for the specified key from this map if present, and returns the value.

Use the method ex17 of MapTest class and the execution result of ex17 is shown in Fig. 6. When using the ex17 method, uncomment "//mt.ex17()" in the main method of MapTest.java.

Submit "TableMap.java" that satisfies Exercise 17.

```
B:4 U:3 V:8 K:2
the value associated with key U is 3
B:4 U:3 V:8 K:2
the value popped with key U is 3
B:4 V:8 K:2
```

Fig. 6 The execution result of method ex17 of "MapTest.java".