LAB16_ANP_C6339_HASHMAP

Due date: Friday, 24 November 2023, 5:30 AM
Maximum number of files: 1
Type of work: Individual work
Lingabathula Thapaswi[AF0339471]

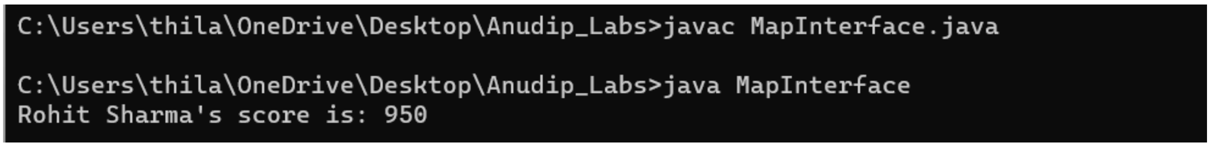
Assignment-1.

Write a program in Java to create a Map Interface where we can store the cricketer name in it along with his scores and search for the batsman name and display his score. [Hint: use containsKey() method to search batsman name]

Solution:-

```
import java.util.*;


 public class MapInterface {


  public static void main(String[] args) {


   // Creating a HashMap to store name and score

   HashMap<String, Integer> cricketers = new HashMap<String, Integer>();


   // Adding elements to the Map

   cricketers.put("Virat Kohli", 1000);

   cricketers.put("Rohit Sharma", 950);

   cricketers.put("MS Dhoni", 900);

   cricketers.put("Steve Smith", 800);


   // Searching for a batsman name

   String searchName = "Rohit Sharma";


   // Checking if the name exists using containsKey()

   if(cricketers.containsKey(searchName)) {


    // Displaying the score using get() method

    System.out.println(searchName + "'s score is: " +

             cricketers.get(searchName));
   }
   else {
    System.out.println("Name not found");
   }
  }
 }
```

Output:



```
C:\Users\thila\OneDrive\Desktop\Anudip_Labs>javac MapInterface.java

C:\Users\thila\OneDrive\Desktop\Anudip_Labs>java MapInterface
Rohit Sharma's score is: 950
```

Assignment-2.

Write a Java program that demonstrates the functionality of this dictionary application using a TreeMap.

Your program should include the following features:

i)A TreeMap named dictionary to store word-definition pairs.

ii)A way to input word-definition pairs and add them to the dictionary.

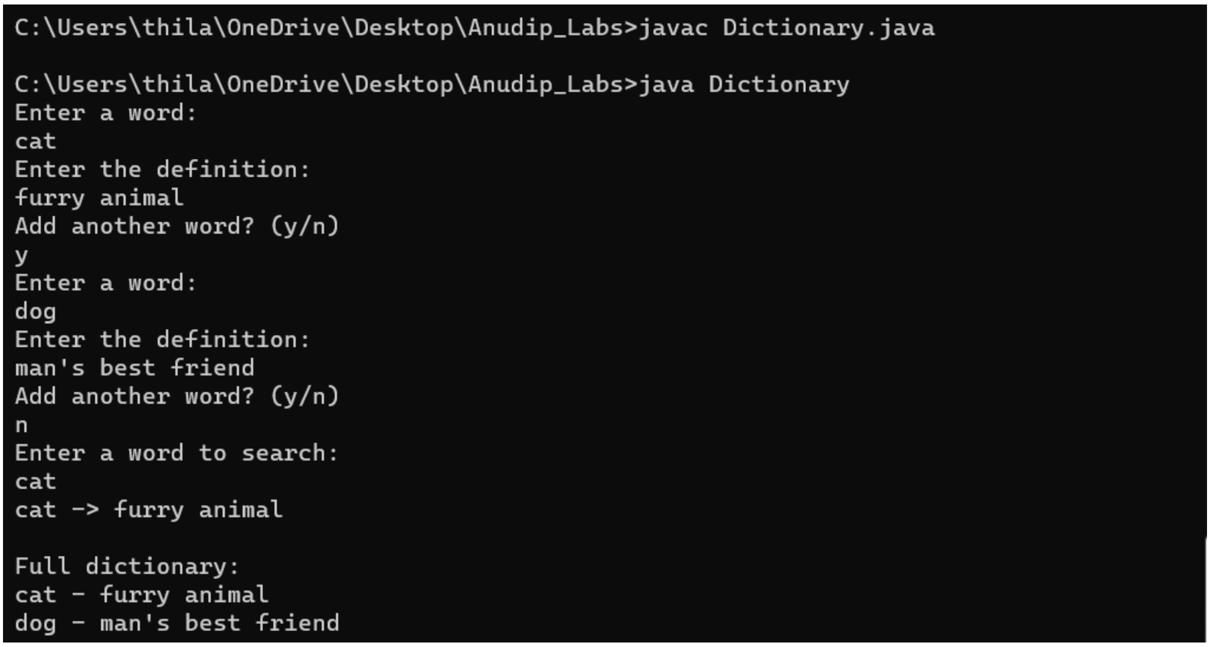
iii)A way to retrieve and display the definition of a specific word.

iv)An iteration through the dictionary to display all word-definition pairs in alphabetical order based on words

Solution:-

```
import java.util.Map;

import java.util.Scanner;

import java.util.TreeMap;


public class Dictionary {


    public static void main(String[] args) {


        // Create a TreeMap to store word-definition pairs

        TreeMap<String, String> dictionary = new TreeMap<>();


        Scanner sc = new Scanner(System.in);
        // Loop to add word-definition pairs to the dictionary
        while(true) {
            System.out.println("Enter a word: ");
            String word = sc.nextLine();

            System.out.println("Enter the definition: ");
            String definition = sc.nextLine();

            dictionary.put(word, definition);

            System.out.println("Add another word? (y/n)");
            String choice = sc.nextLine();
            if(choice.equals("n")) break;
        }

        // Retrieve and display the definition of a specific word
        System.out.println("Enter a word to search: ");
        String searchWord = sc.nextLine();
        System.out.println(searchWord + " -> " + dictionary.get(searchWord));

        // Iterate through the dictionary and print all entries
        System.out.println("\nFull dictionary:");
        for (Map.Entry<String, String> entry : dictionary.entrySet()) {
            System.out.println(entry.getKey() + " - " + entry.getValue());
        }
    }
}
```

Output:



```
C:\Users\thila\OneDrive\Desktop\Anudip_Labs>javac Dictionary.java

C:\Users\thila\OneDrive\Desktop\Anudip_Labs>java Dictionary
Enter a word:
cat
Enter the definition:
furry animal
Add another word? (y/n)
y
Enter a word:
dog
Enter the definition:
man's best friend
Add another word? (y/n)
n
Enter a word to search:
cat
cat -> furry animal

Full dictionary:
cat - furry animal
dog - man's best friend
```