EmployeeAgeMap.java

import java.util.HashMap;

import java.util.Map;

public class EmployeeAgeMap {

public static void main(String[] args) {

// Create a HashMap to store employee names and ages

Map<String, Integer> employeeAges = new HashMap<>();

// Add employee names and ages to the map

employeeAges.put("John", 25);

employeeAges.put("Alice", 30);

employeeAges.put("Bob", 35);

employeeAges.put("Emily", 28);

employeeAges.put("David", 40);

// Display the employee ages map

System.***out***.println("Employee Ages:");

System.***out***.println(employeeAges);

// Associate a new age with a specified employee name

String newEmployee = "Sarah";

int newAge = 32;

employeeAges.put(newEmployee, newAge);

// Display the updated employee ages map

System.***out***.println("\nUpdated Employee Ages:");

System.***out***.println(employeeAges);

}

}

Output

Employee Ages:

{Emily=28, Bob=35, Alice=30, John=25, David=40}

Updated Employee Ages:

{Emily=28, Bob=35, Sarah=32, Alice=30, John=25, David=40}

CricketerScores.java

import java.util.HashMap;

import java.util.Map;

public class CricketerScores {

public static void main(String[] args) {

// Create a Map to store cricketer names and their scores

Map<String, Integer> cricketerScores = new HashMap<>();

// Add cricketers and their scores to the map

cricketerScores.put("Virat Kohli", 120);

cricketerScores.put("Rohit Sharma", 98);

cricketerScores.put("MS Dhoni", 45);

cricketerScores.put("David Warner", 85);

cricketerScores.put("Joe Root", 110);

// Search for a batsman's name and display his score

String batsmanName = "Virat Kohli";

if (cricketerScores.containsKey(batsmanName)) {

int score = cricketerScores.get(batsmanName);

System.***out***.println("Score of " + batsmanName + ": " + score);

} else {

System.***out***.println("Batsman '" + batsmanName + "' not found in the records.");

}

}

}

Output

Score of Virat Kohli: 120

DictionaryApp.java

import java.util.\*;

public class DictionaryApp {

public static void main(String[] args) {

// TreeMap to store word-definition pairs

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

// Scanner object to read user input

Scanner scanner = new Scanner(System.***in***);

// Add word-definition pairs to the dictionary

System.***out***.println("Enter word-definition pairs (type 'done' to finish):");

String word, definition;

while (true) {

System.***out***.print("Word: ");

word = scanner.nextLine();

if (word.equalsIgnoreCase("done")) {

break;

}

System.***out***.print("Definition: ");

definition = scanner.nextLine();

dictionary.put(word, definition);

}

// Retrieve and display the definition of a specific word

System.***out***.print("\nEnter a word to get its definition: ");

String searchWord = scanner.nextLine();

if (dictionary.containsKey(searchWord)) {

System.***out***.println("Definition: " + dictionary.get(searchWord));

} else {

System.***out***.println("Word not found in the dictionary.");

}

// Display all word-definition pairs in alphabetical order

System.***out***.println("\nWord-Definition Pairs (in alphabetical order):");

for (Map.Entry<String, String> entry : dictionary.entrySet()) {

System.***out***.println(entry.getKey() + ": " + entry.getValue());

}

}

}

Output

Word: good

Definition: nice

Word: done

Enter a word to get its definition: good

Definition: nice

Word-Definition Pairs (in alphabetical order):

good: nice