Java Essentials Module 5 Case Study 1

Please run jar file to test the program.

Command: java -jar Module5CaseStudy1.jar

Notes:

* Student records hashmap created for each exercise/function
* Student records hashmap in 2 forms <Integer, String> and <String, Integer>
* Please enter names for comparison correctly or it will create an error
  + Try/catch surrounds the comparison

# StudentRecords Class (includes main):

**import** java.util.Collections;

**import** java.util.Comparator;

**import** java.util.HashMap;

**import** java.util.LinkedHashMap;

**import** java.util.LinkedList;

**import** java.util.Map;

**import** java.util.Map.Entry;

**import** java.util.Scanner;

**import** java.util.\*;

**public** **class** StudentRecords {

**static** Scanner *adminInput* = **new** Scanner(System.***in***);

**static** HashMap<Integer, String> createStudentRecords() {

HashMap<Integer, String> studentRecords = **new** HashMap<Integer, String>();

studentRecords.put(1, "Francis");

studentRecords.put(2, "Mabel");

studentRecords.put(3, "Arnold");

studentRecords.put(4, "Donald");

studentRecords.put(5, "Raymond");

**return** studentRecords;

}

**static** HashMap<String, Integer> createStudentRecordsForSorting() {

HashMap<String, Integer> studentRecords = **new** HashMap<String, Integer>();

studentRecords.put("Francis", 3);

studentRecords.put("Mabel", 2);

studentRecords.put("Arnold", 5);

studentRecords.put("Donald", 1);

studentRecords.put("Raymond", 4);

**return** studentRecords;

}

**static** **void** showStudentRecords(HashMap<Integer, String> studentRecords) {

System.***out***.println("\nStudent records: ");

System.***out***.println("\nID: \tName: ");

**for** (Integer i : studentRecords.keySet()) {

System.***out***.println(i + "\t" + studentRecords.get(i));

}

}

**static** **void** showStudentRecords2(HashMap<String, Integer> studentRecords) {

System.***out***.println("\nStudent records: ");

System.***out***.println("\nName: \tID: ");

**for** (String i : studentRecords.keySet()) {

System.***out***.println(i + "\t" + studentRecords.get(i));

}

}

**static** **void** showStudentRecords3(Map<String, Integer> studentRecords) {

System.***out***.println("\nStudent records: ");

System.***out***.println("\nName: \tID: ");

**for** (String i : studentRecords.keySet()) {

System.***out***.println(i + "\t" + studentRecords.get(i));

}

}

**static** **void** addStudentRecords() {

HashMap<Integer, String> studentRecords = **new** HashMap<Integer, String>();

System.***out***.println("Add student records: ");

**for** (Integer i = 1; i <=5; i++) {

System.***out***.println("\nPlease enter student name: ");

String studentName = *adminInput*.nextLine();

studentRecords.put(i, studentName);

}

*showStudentRecords*(studentRecords);

}

**static** **void** deleteStudentRecords() {

HashMap<Integer, String> studentRecords = *createStudentRecords*();

*showStudentRecords*(studentRecords);

System.***out***.println("\nStudents Mabel (ID 2) and Donald (ID 4) are expelled for kidnapping a pig.");

System.***out***.println("They are now removed from the student records. ");

studentRecords.remove(2);

studentRecords.remove(4);

*showStudentRecords*(studentRecords);

}

**static** **void** editStudentRecords() {

HashMap<Integer, String> studentRecords = *createStudentRecords*();

*showStudentRecords*(studentRecords);

System.***out***.println("\nSome student records are incorrect and are edited.");

System.***out***.println("\nStudent Arnold (ID 3) is edited to Arnoldia.");

System.***out***.println("Student Raymond (ID 3) is edited to Raymondo. ");

studentRecords.replace(3, "Arnoldia");

studentRecords.replace(5, "Raymondo");

*showStudentRecords*(studentRecords);

}

**static** **int** compareTo(Integer studentID1, Integer studentID2) {

**if**(studentID1 < studentID2) {

**return** -1;

} **else** **if** (studentID1 > studentID2) {

**return** 1;

} **else** {

**return** 0;

}

}

**static** **void** compareStudentID (){

System.***out***.println("\nCompare student ID:");

HashMap<String, Integer> studentRecords = *createStudentRecordsForSorting*();

*showStudentRecords2*(studentRecords);

System.***out***.println("\nEnter student 1 name:");

String studentName1 = *adminInput*.nextLine();

Integer studentID1 = studentRecords.get(studentName1);

System.***out***.println("\nEnter student 2 name:");

String studentName2 = *adminInput*.nextLine();

Integer studentID2 = studentRecords.get(studentName2);

**int** compareResult = *compareTo*(studentID1, studentID2);

**if** (compareResult == -1) {

System.***out***.println("\n" + studentName1 + " has ID less than " + studentName2 + ".");

System.***out***.println(studentID1 + " < " + studentID2);

} **else** **if** (compareResult == 1) {

System.***out***.println("\n" + studentName1 + " has ID more than " + studentName2 + ".");

System.***out***.println(studentID1 + " < " + studentID2);

} **else** {

System.***out***.println("\n" + studentName1 + " has the same ID as " + studentName2 + ".");

System.***out***.println(studentID1 + " = " + studentID2);

}

}

**public** **static** **void** sortStudents() {

HashMap<String, Integer> studentRecords = *createStudentRecordsForSorting*();

System.***out***.println("\nPre sorting");

*showStudentRecords2*(studentRecords);

Map<String, Integer> studentRecordsSortedByName = **new** TreeMap<String, Integer>(studentRecords);

System.***out***.println("\nPost sorting by name");

*showStudentRecords3*(studentRecordsSortedByName);

// Create a list from elements of HashMap

LinkedList<Entry<String, Integer>> list = **new** LinkedList<Map.Entry<String, Integer> >(studentRecords.entrySet());

// Sort the list

Collections.*sort*(list, **new** Comparator<Map.Entry<String, Integer> >() {

**public** **int** compare(Map.Entry<String, Integer> student1,

Map.Entry<String, Integer> student2)

{

**return** (student1.getValue()).compareTo(student2.getValue());

}

});

HashMap<String, Integer> studentRecordsSortedByID = **new** LinkedHashMap<String, Integer>();

**for** (Map.Entry<String, Integer> student: list) {

studentRecordsSortedByID.put(student.getKey(), student.getValue());

}

System.***out***.println("\nPost sorting by ID");

*showStudentRecords2*(studentRecordsSortedByID);

}

**public** **static** **void** main(String[] args) {

*addStudentRecords*();

*deleteStudentRecords*();

*editStudentRecords*();

*compareStudentID* ();

*sortStudents*();

}

}

# Output:

Add student records:

Please enter student name:

Ralph

Please enter student name:

Frank

Please enter student name:

Phillip

Please enter student name:

Margaret

Please enter student name:

Evaline

Student records:

ID: Name:

1 Ralph

2 Frank

3 Phillip

4 Margaret

5 Evaline

Student records:

ID: Name:

1 Francis

2 Mabel

3 Arnold

4 Donald

5 Raymond

Students Mabel (ID 2) and Donald (ID 4) are expelled for kidnapping a pig.

They are now removed from the student records.

Student records:

ID: Name:

1 Francis

3 Arnold

5 Raymond

Student records:

ID: Name:

1 Francis

2 Mabel

3 Arnold

4 Donald

5 Raymond

Some student records are incorrect and are edited.

Student Arnold (ID 3) is edited to Arnoldia.

Student Raymond (ID 3) is edited to Raymondo.

Student records:

ID: Name:

1 Francis

2 Mabel

3 Arnoldia

4 Donald

5 Raymondo

Compare student ID:

Student records:

Name: ID:

Francis 3

Raymond 4

Mabel 2

Arnold 5

Donald 1

Enter student 1 name:

Francis

Enter student 2 name:

Mabel

Francis has ID more than Mabel.

3 < 2

Pre sorting

Student records:

Name: ID:

Francis 3

Raymond 4

Mabel 2

Arnold 5

Donald 1

Post sorting by name

Student records:

Name: ID:

Arnold 5

Donald 1

Francis 3

Mabel 2

Raymond 4

Post sorting by ID

Student records:

Name: ID:

Donald 1

Mabel 2

Francis 3

Raymond 4

Arnold 5