



ภาษาการเขียนโปรแกรม

WEEK7-8

02-212-213

อ.ธิดาวรรณ คล้ายศรี



- 1.1 An Introduction to Programming (บทนาการเขียนโปรแกรม)
- 1.2 Traditional Programming Languages (ภาษาการเขียนโปรแกรมแบบดั้งเดิม)
- 1.3 Syntax of the Programming Languages (ไวยากรณ์ของภาษาเขียนโปรแกรม)
 - 1.3.1 Control Block (การควบคุมบล็อก)
 - 1.3.2 Block Structure (โครงสร้างบล็อก)
 - 1.3.3 Recursive (การทำงานแบบเรียกตัวเอง)
- 1.4 Review Data Structure (ทบทวนโครงสร้างข้อมูล)
- 1.5 Sorting and Searching Algorithms (อัลกอริทึมจัดลำดับและสืบค้นข้อมูล)
- 1.6 Testing Programs (การทดสอบโค้ด)
 - 1.6.1 Compiler
 - 1.6.2 Running Time and Debugging
 - 1.6.3 Creating Test Cases

บทเรียนสัปดาห์ที่แล้ว Using Java Objects-comparator

- ใ้รับข้อมูลคะแนนโดยอ่านมาจาก Text File แล้วทำการจัดเรียงข้อมูล

Std_Names	Scores
Somchai	65
Suthida	72
Bonsong	83
Virat	85
Ratree	71
Kampon	60
Taworn	55

```
Arrays.sort(pairs, new ComparingDataByScore());
```

```
Arrays.sort(pairs, new ComparingDataByName());
```



1.5 Basic Algorithms

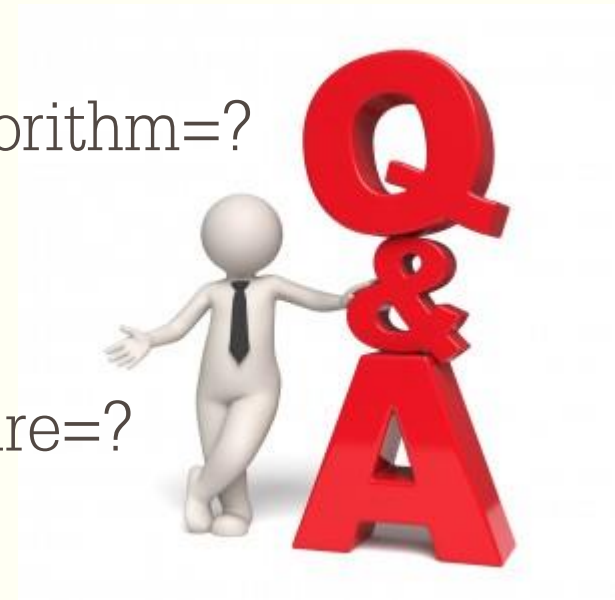
อัลกอริธึมพื้นฐานในการเขียนโปรแกรม

Searching Data (การค้นหาข้อมูล)

Std_Names	Scores
Somchai	65
Suthida	72
Bonsong	83
Virat	85
Ratree	71
Kampon	60
Taworn	55

Algorithm=?

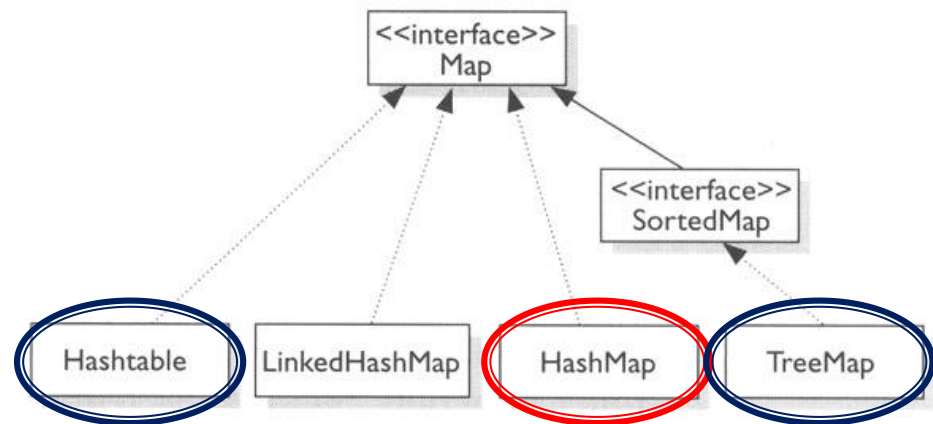
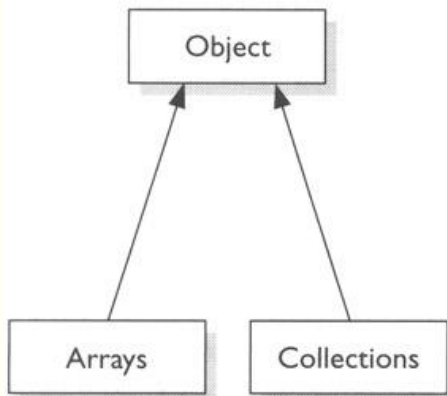
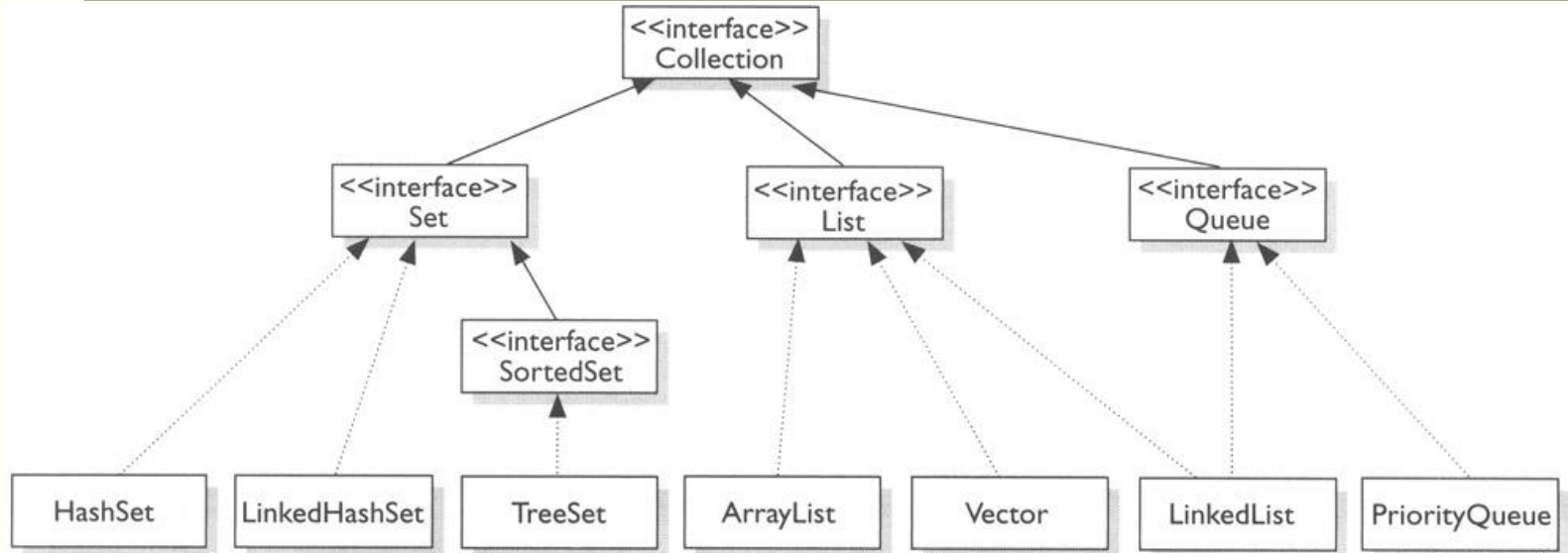
Data Structure=?



Data Structures for Searching Data

(โครงสร้างข้อมูลในการค้นหาข้อมูล)

Collections/Hash



Hash Map/ Hash Table

```
HashMap< ArrayList<Integer>, Integer> pairs = new HashMap<>();
```

```
HashMap< ArrayList<String>, Integer > pairs = new HashMap<>();
```

Searching Data Using Java Hash Map

- 1) ให้รับข้อมูลคะแนนโดยอ่านมาจาก Text File แล้วทำการจัดเก็บไว้ใน hash map
- 2) ให้แสดงผลข้อมูลจากตัวแปร โครงสร้างข้อมูลแบบ map ออกมา
- 3) “ค้นหาคะแนนของนศ. คนหนึ่ง”
→ ดึงข้อมูลคะแนนที่เก็บไว้ใน map โดยค้นหาด้วยชื่อ
- 4) “ค้นหา นศ. ที่มีคะแนน N”
→ ดึงข้อมูลชื่อนศ. ที่เก็บไว้ใน map โดยค้นหาด้วยคะแนน

Std_Names	Scores
Somchai	65
Suthida	72
Bonsong	83
Virat	85
Ratree	71
Kampon	60
Taworn	55



ฝึกปฏิบัติ

Projects x Files Services —

SearchingScore

- Source Packages
 - PackageSearchingsScores
 - ConstructingMapNameScore.java
 - SearchingScore.java
- Test Packages
 - <default package>
 - Test_SearchingNamesScores.java
- Libraries
- Test Libraries

ConstructingMapNameScore.java x SearchingScore.java x Test_SearchingNamesScores.java x

Source History

```
1  /*
2  * This proram is to search students' scores
3  * Algorithm:
4  *   1) Reading data score from a text file into HashMap
5  *   2) Searching student's scoe by entry their name
6  *   3) Displaying the score
7  */
8  package PackageSearchingsScores;
9
10 import ...7 lines
17
18 /**...4 lines */
22 public class ConstructingMapNameScore {
23
24     private final HashMap<String, Integer> nameScore = new HashMap<>();
25
26     /** reads data from a text file into a Hashmap ...5 lines */
31 public void inputNameScoreFromTextFileToMap(String fName) throws FileNotFoundException {
32     Scanner fIn = new Scanner(new FileReader(fName));
33     do {
34         nameScore.put(fIn.next(), fIn.nextInt());
35     } while (fIn.hasNext());
36 }
37
```

Projects x Files Services

ConstructingMapNameScore.java x SearchingScore.java x Test_SearchingNamesScores.java x

Source History

37

38 `/**displays names and scores of students from hashmap.`

39 `*`

40 `*/`

41 `public void printNameScoreMap() {`

42 `Iterator< Map.Entry<String, Integer>> itr = nameScore.entrySet().iterator();`

43

44 `System.out.println("\n-----");`

45 `System.out.println("Names\tScores");`

46 `System.out.println ("-----");`

47 `while (itr.hasNext()) {`

48 `Map.Entry<String, Integer> entryC = itr.next();`

49 `System.out.println(entryC.getKey() + "\t" + entryC.getValue());`

50 `}`

51 `}`

52

53 `/** writes names-scores containing sets of student's names and scores from ...7 lines */`

60 `public void writeNameScoreMapToTextFile(String file) throws FileNotFoundException {`

61 `Iterator< Map.Entry<String, Integer>> itr = nameScore.entrySet().iterator();`

62 `try (PrintWriter fOut = new PrintWriter(new PrintWriter(file))) {`

63 `while (itr.hasNext()) {`

64 `Map.Entry<String, Integer> entryC = itr.next();`

65 `fOut.println(entryC.getKey() + " " + entryC.getValue());`

66 `}`

67 `}`

68 `}`

SearchingScore

Source Packages

PackageSearchingsScores

ConstructingMapNameScore.java

SearchingScore.java

Test Packages

<default package>

Test_SearchingNamesScores.java

Libraries

Test Libraries

Projects x Files Services

- SearchingScore
 - Source Packages
 - PackageSearchingsScores
 - ConstructingMapNameScore.java
 - SearchingScore.java
 - Test Packages
 - <default package>
 - Test_SearchingNamesScores.java
 - Libraries
 - Test Libraries

SearchingScore.java x ConstructingMapNameScore.java x

Source History

```
20 public void searchStudentScoreByName(ConstructingMapNameScore S, String name) {
21     Iterator< Entry<String, Integer>> itrC = S.getNameScore().entrySet().iterator();
22     while (itrC.hasNext()) {
23         Entry<String, Integer> entryN = itrC.next();
24         if (entryN.getKey().contains(name)) {
25             System.out.println("\n\t" + entryN.getKey() + " " + entryN.getValue());
26         }
27     }
28 }
29
30 public void searchStudentNameByScore(ConstructingMapNameScore S, Integer score) {
31     Iterator< Entry<String, Integer>> itrC = S.getNameScore().entrySet().iterator();
32     while (itrC.hasNext()) {
33         Entry<String, Integer> entryN = itrC.next();
34         if (entryN.getValue().equals(score)) {
35             System.out.println("\n\t" + entryN.getKey() + " " + entryN.getValue());
36         }
37     }
38 }
39 }
```

Projects X Files Services — SearchingScore.java X ConstructingMapNameScore.java X Test_SearchingNamesScores.java X

Source History

1 |
2 | import ...4 lines
6 |
7 | /**...4 lines */
11 | public class Test_SearchingNamesScores {
12 |
13 | public Test_SearchingNamesScores () {
14 | }
15 |
16 | /*
17 | * This prorame is to search for students' score for a given name
18 | * Algorithm:
19 | * 1) Reading data score from a text file into Array
20 | * 2) Sorting data using Array.sort by ascending score then name
21 | * 3) Displaying the sorted data or writing to another text file
22 | */
23 | @Test
24 | public void mainTesting() throws FileNotFoundException {
25 |
26 | ConstructingMapNameScore J = new ConstructingMapNameScore();
27 | J.inputNameScoreFromTextFileToMap("NameScore.txt");//1) Reading data score
28 | System.out.println("Original Names and Scores:");
29 | J.printNameScoreMap();
30 | System.out.println("--Total Students = " + J.getNameScore().size()+"--");
31 |
32 | /*Searching for studen't score from a given name*/
33 | SearchingScore S = new SearchingScore();
34 | S.searchStudentNameByScore(J, 83);
35 | S.searchStudentScoreByName(J, "Virat");
36 | }
37 | }

Hash Map/ Hash Table

```
HashMap< ArrayList<String>, Integer> nameScore = new HashMap<>();
```

```
/**adds names and score into hashmap  
 *  
 * @param n name- arrayList of String  
 * @param s score-integer  
 */  
public void addNamesScores(ArrayList<String> n, Integer s) {  
    nameScore.put(new ArrayList<>(n), s);  
}
```

```
/**displays names and scores of students from hashmap.  
 *  
 */  


---

  
public void printNameScoreMap() {  
    Iterator< Map.Entry<ArrayList<String>, Integer>> itr = nameScore.entrySet().iterator();  
  
    while (itr.hasNext()) {  
        Map.Entry<ArrayList<String>, Integer> entryC = itr.next();  
        System.out.println(entryC.getKey() + " \t " + entryC.getValue());  
    }  
}
```

```
public void searchStudentScore(ConstructingMapNameScore S, String name) {  


---

  
    Iterator< Entry<ArrayList<String>, Integer>> itrC = S.getNameScore().entrySet().iterator();  
    while (itrC.hasNext()) {  
        Entry<ArrayList<String>, Integer> entryN = itrC.next();  
        if (entryN.getKey().contains(name)) {  
            System.out.println("\n\t" + entryN.getKey() + " " + entryN.getValue());  
        }  
    }  
}
```