

# **Report**

## **Assignment 6 : Array List**

จัดทำโดย

นายสรรธ แก้วโชติช่วงกุล 63070501067 CPE REGULAR

รายงานนี้เป็นส่วนหนึ่งของวิชา CPE111

Programming With Data Structures

King Mongkut's University of Technology Thonburi

## ● สิ่งที่ทำใน Assignment

1. สร้าง class ในการทำงานเกี่ยวกับ Dictionary(Array List) ประกอบด้วย methods ได้แก่
  - Method ReadFile เอาไว้อ่านไฟล์แล้วเพิ่มข้อมูลเข้าไปใน ArrayList
  - Method DeleteRepeatWord เอาไว้ลบคำที่เหมือนกันทุกอย่าง
  - Method count keyword เอาไว้นับจำนวนคำที่มี keyword เหมือนกันอย่างเดียว
  - Method Find Print KeyWord เอาไว้หา Index ของ keyword ที่ต้องการหาและส่งข้อมูลไปปรี้น
  - Method Print token ใช้ในการรับข้อมูลมาและแสดงผล
2. สร้าง class ในการทำงานเกี่ยวกับ Structure(โครงสร้างข้อมูล) โดยเฉพาะและสร้าง method ได้แก่
  - method ที่ใช้ Compare และส่งค่าให้ Library ไป Sort และ Search ข้อมูล (binary search)
  - method CompareRepeatWord เช็คคำที่มี keyword meaning และ type เหมือนกันทั้งสาม
  - method CompareWord เช็คคำที่มี keyword เหมือนกันแต่ type และ meaning ต่างกัน

- Source code
- Class Dictionary(main)

### ➤ method Read\_File & DeleteRepeateWord

```

10 import java.io.BufferedReader;
11
12 public class DictArray {
13     static ArrayList<Dnode> dict = new ArrayList<Dnode> (); // สร้างตัวแปร Array list แบบ global มาเก็บข้อมูลที่ผ่านมา
14     static int count1=0;
15     static String x = null; // เก็บค่า String ที่หาเจอครั้งแรก
16     public static void Read_File() { // ฟังก์ชันในการอ่านไฟล์
17         String str;
18         try { // ใช้ try catch เพื่อ catch error!
19             FileInputStream fcsv = new FileInputStream("C:\\Users\\LENOVO\\Desktop"
20                 + "\\CPE-YEAR1\\CPE SUBJECT\\CPE111 DATA STRUCTURE\\Assignment 6\\UTF8_Lexitron.csv"); // สร้างไฟล์ขึ้นมาเก็บที่ตัวแปร fcsv Endcoding
21             InputStreamReader utf = new InputStreamReader(fcsv, "UTF-8"); // ระบุ Endcoding
22             BufferedReader buff = new BufferedReader(utf); // ใช้ class bufferreader อ่านข้อมูล BOM
23             buff.read(); // อ่าน BOM ตัว (BOM ตัวแรกเก็บ char เหนือ char) แล้ว buff.read() เพื่ออ่าน char
24             while((str = buff.readLine()) != null) { // จนจบอ่านไฟล์
25                 Dnode x = new Dnode(str); // จะ memories มาเก็บข้อมูลที่ผ่านมาเป็น Structure
26                 dict.add(x); // เก็บข้อมูล (ใช้ Library ArrayList)
27                 count1++; // นับจำนวน
28             }
29             fcsv.close(); // ปิดไฟล์ที่มีอ่านเสร็จ
30         } catch (Exception e) { // ถ้าเกิด Error!
31             System.out.printf("Error! Can't Read File %d"); // แสดงข้อความ
32         }
33     }
34     public static int Delete_RepeatWord() { // method ลบคำซ้ำ
35         int i, j = 0;
36         for(i=0; i<dict.size()-1; i++) { // วนรอบทุกตัวมาจับคู่
37             if(dict.get(i).CompareRepeatWord(dict.get(i+1))) { // ใช้ CompareRepeatWord ใน Class Dnode เพื่อเช็คว่ามีคำเหมือนกันไหม
38                 dict.remove(i+1); // ลบคำที่ซ้ำ
39                 j++; // นับจำนวนที่ซ้ำ
40             }
41         }
42         count1 = count1 - j; // อัปเดต count
43         return j; // คืนค่าจำนวนที่ลบ
44     }
45 }

```

### ➤ method count\_keyword & Find\_print\_keyword

```

43 public static int count_keyword() { // method นับคำที่มี word เหมือนกันแต่ type และ meaning ต่างกัน
44     int i, j=1, count=0; // j เริ่มที่ 1 เพราะนับตัวแปรเทียบกับตัวที่ 0 มันจะไม่นับตัวที่ 0
45     for(i=0; i<dict.size()-1; i++) { // วนรอบเพื่อเช็ค
46         if(dict.get(i).CompareKeyword(dict.get(i+1))) { // ใช้ CompareKeyword ใน Class Dnode (เพราะ get ข้อมูลออกมาแล้วมันเป็น Dnode)
47             j++; // นับจำนวนคำที่มี Keyword เหมือนกัน
48         }
49         else { // แต่ถ้ามันไม่เหมือนกัน
50             if(count < j) { // เช็คก่อนว่า count หรือจำนวนของตัวก่อนหน้ามันมากกว่าตัวนี้หรือป่าวเพราะเราต้องการตัวที่ซ้ำมากที่สุด
51                 count = j; // จำค่า j
52                 x = dict.get(i).word; // จำ keyword
53             }
54             j=1; // รีเซ็ต j เมื่อเจอตัวที่ keyword ไม่เหมือนกันแล้ว
55         }
56     }
57     return count; // ได้ค่ากลับมาที่มีคำ
58 }
59 public static void Find_print_keyword(String str) { // methods ในการหา keyword ที่ต้องการ
60     int index;
61     Dnode x = new Dnode(); // สร้าง memories ของ Dnode มาหนึ่งตัว
62     x.word = str; // ให้ word ของตัวนี้เป็นตัวที่เราต้องการเปรียบเทียบ
63     //System.out.printf("Your add : %s\n", x.word);
64     index = Collections.binarySearch(dict, x); // ใช้ Library Binary search ในการค้นหา word นั้นแล้วให้ Index ออกมา
65     //System.out.printf("Index Found : %d\n", index);
66
67     if(index > 0) { // ถ้าไม่ได้เจอตัวแรก
68         while(index >= 0 && dict.get(index).word.equalsIgnoreCase(str)) { // ให้วนหาตัวแรกของ word นั้นเพราะ Index ที่ได้อาจไม่ใช่ตัวแรก
69             index--;
70         }
71         if(index < 0) { // เจอ index ที่ตำแหน่งแรก index = 0 -> index-- = -1 หลุดจาก Loop
72             print_token(0, str); // จะได้อ่านตำแหน่งแรกคือ 0
73         } else { // ไม่ใช่ตำแหน่งแรกมันจะหลุดออกมาก่อนตัวแรก
74             print_token(index+1, str); // ต้องบวกไป 1 ตำแหน่งจะได้ตำแหน่งแรก
75         }
76     } else if(index == 0) { // ถ้า Index เป็นตำแหน่งแรกตั้งแต่ Search ครั้งแรก
77         print_token(0, str); // ส่งไปปรี้นเลย
78     } else { // น้อยกว่า 0 หรือไม่เจอ word นั้น
79         System.out.printf("Not Found token "+str+" in Dictionary.%d");
80     }
81 }

```

## ➤ Print\_token

```
81 public static void print_token(int FirstIndex,String str) { // methods ในการปรับข้อมูล
82     int count=0,i,last;
83     i = FirstIndex; // กำหนด i เป็น Index ตัวแรกที่ปรับ
84     //System.out.printf("Dict size : %d",dict.size());
85     while(i < dict.size() && dict.get(i).word.equalsIgnoreCase(str)) { // วนรอบจนกว่าตัวท้ายก่อนหน้าจะไม่เหมือนจะเอาไป Compare ก่อนเมื่อมันเลยตัวสุดท้ายไป
86         count++; //นับ
87         i++; // เลื่อนไปเรื่อยๆ
88     }
89     last = FirstIndex+count-1; // หากตำแหน่งสุดท้าย
90     System.out.printf("\t\tFound "+count+" token at "+FirstIndex+" - %d %n",last);
91     for(i=1;i<=count;i++) { // วนรอบแสดงข้อมูล
92         System.out.printf("\t\t\t%d) %s %s(%s)%n",i,dict.get(FirstIndex).word,dict.get(FirstIndex).mean,dict.get(FirstIndex).type);
93         FirstIndex++;
94     }
95 }
```

## ➤ Main (method ที่ใช้รัน)

```
96 public static void main(String[] args) {
97     int i,repeatwords = 0,keywords = 0;
98     String str;
99     Scanner in = new Scanner(System.in);
100     Read_File(); // อ่าน
101     System.out.println("Total Read = "+count1+" records."); // แสดงผลตอนแรก
102     Collections.sort(dict); // เรียงลำดับ
103     repeatwords = Delete_RepeatWord(); // หากซ้ำ + ลบซ้ำ
104     System.out.println("Repeat Words= "+repeatwords+" records."); // แสดงจำนวนตัวซ้ำ
105     System.out.println("Total Read Now = "+count1+" records."); // แสดงจำนวนที่เหลือหลังจากลบไป
106     keywords = count_keyword(); // หากจำนวนที่มี word ซ้ำมากที่สุด(type meaning ต่างกัน)
107     System.out.println("Maximum meaning : "+ x +" have "+keywords+" meaning."); // แสดงจำนวนที่ word เหมือนกันแต่ความหมายต่างกัน
108     Find_print_keyword(x); // ปรับIndex และ ข้อมูลของตัวนั้นออกมา
109     do {
110         System.out.printf("Enter token > ");
111         str = in.nextLine(); // อ่านข้อมูล
112         //System.out.println("Your token : "+str+"");
113         str = str.trim().replaceAll("\\s+", " "); // เช็คว่ามีวรรคทั้งหมดแล้วแทนด้วยเว้นวรรคตัวเดียว
114         //System.out.println("Your token after trim : "+str+"");
115         Find_print_keyword(str); // ส่ง token ที่ปรับ Index และปรับออกมา
116     }while(!str.equalsIgnoreCase("end")); // หากค่า str ไม่ใช่ end
117     System.out.println("End Program.");
118     System.out.printf("This program is written by Sorathorn Kaewhotchuankul 63070501067 CPE/1");
119 }
120 }
121 }
122 }
```

- Class ที่ทำงานเกี่ยวกับข้อมูล (Dnode)

```
1 import java.util.ArrayList;
2
3 public class Dnode implements Comparable<Dnode>{
4     String word;
5     String mean;
6     String type;
7     public Dnode() { // constructor แบบ node เปล่า
8         word = "";
9         mean = "";
10        type = "";
11    }
12
13    public int compareTo(Dnode x) { // Comparable ของ Sort และ binary search
14        return (int) this.word.compareToIgnoreCase(x.word); // รีটারน์เป็น int
15    }
16
17    public Dnode(String buff) { // constructor แบบมีข้อมูล
18        buff = buff.trim().replaceAll("\\s+", " "); // ลบ white space หน้า หลัง ตรงกลางที่เป็น white space ยาวๆทิ้งให้เหลืออันเดียว
19        String [] str = buff.split(","); // split มันจาก ,
20        word = str[0];
21        mean = str[1];
22        type = str[2];
23    }
24
25    boolean CompareRepeatWord(Dnode x) { // methods เปรียบเทียบคำที่ซ้ำกันทุกอย่าง
26        if(this.word.equalsIgnoreCase(x.word) && this.mean.equalsIgnoreCase(x.mean) && this.type.equalsIgnoreCase(x.type))
27            return true;
28        else
29            return false;
30    }
31
32    boolean CompareKeyword(Dnode x) { // methods เปรียบเทียบคำที่ซ้ำกันเฉพาะ word
33        if(this.word.equalsIgnoreCase(x.word) && !(this.mean.equalsIgnoreCase(x.mean)))
34            return true;
35        else return false;
36    }
37 }
```

**Note!** อธิบายโค้ด comment อยู่ในตัวโค้ด

## ● Test case & อธิบาย

เมื่อเริ่มโปรแกรมจะเริ่มอ่านไฟล์และนับจำนวนที่อ่านได้ และหาคำที่ซ้ำกันแล้วลบออก หลังจากนั้นก็แสดงคำศัพท์ที่เหลืออยู่หลังจากลบ

Total Read = 74233 records.  
Repeat Words= 252 records.  
Total Read Now = 73981 records.

Maximum meaning : get off have 35 meaning.  
Found 35 token at 23963 - 23997

- 1) get off เริ่ม (PHRV)
- 2) get off เริ่ม (บางสิ่ง) ได้ดีหรือไม่ดี (PHRV)
- 3) get off เริ่มไปเลย (PHRV)
- 4) get off เริ่มใหม่ (PHRV)
- 5) get off เริ่มทำงาน (PHRV)
- 6) get off เริ่มรู้ (PHRV)
- 7) get off เลิกมือหนึ่ง (PHRV)
- 8) get off เลิกงาน (PHRV)
- 9) get off เลิกทำให้ฉันรำคาญ (PHRV)
- 10) get off เอาออกไป (PHRV)
- 11) get off มุ่งงาน (PHRV)
- 12) get off ไปมา (PHRV)
- 13) get off ไปให้ทัน (PHRV)
- 14) get off ไม่เอื้อเฟื้อ (PHRV)
- 15) get off ไม่โดนใคร (PHRV)
- 16) get off ช่วยเหลือให้ไกลมาจาก (เรื่องที่น่ารังเกียจ) (PHRV)
- 17) get off ตื่นเต้น (PHRV)
- 18) get off ทำความสะอาด (PHRV)
- 19) get off ทำผล (PHRV)
- 20) get off บอกให้เลิกทำหรือเลิกทำ (PHRV)
- 21) get off ปิดความลับของคุณ (PHRV)
- 22) get off ทัก (PHRV)
- 23) get off มีวันหยุด (PHRV)
- 24) get off ขอดมา (PHRV)
- 25) get off ขอบรับ (การควบคุม) (PHRV)
- 26) get off รอดพ้น (PHRV)
- 27) get off รอดพ้นอันตราย (PHRV)
- 28) get off ลงมาจาก (PHRV)
- 29) get off สัม (PHRV)
- 30) get off ส่ง (PHRV)
- 31) get off หนีรอดจาก (PHRV)
- 32) get off หลับ (PHRV)
- 33) get off ออกเดินทาง (PHRV)
- 34) get off ออกจาก (รถ) (PHRV)
- 35) get off ออกจากรถ (PHRV)

โปรแกรมจะหาคำที่มี Keyword ซ้ำกันเพียงอย่างเดียวและบอกจำนวนว่าเจอที่คำ ที่ Index ไหน และ แสดงผลออกมาให้ดูทุกตัว

Enter token > a  
Found 1 token at 0 - 0  
1) A อักษรตัวแรกในภาษาอังกฤษ (N)

Enter token > zymurgy  
Found 1 token at 73980 - 73980  
1) zymurgy การหมักสุรา (N)

Enter token > Gamine  
Found 2 token at 23413 - 23414  
1) gamine (เด็กหญิง) ซึ่งเล่นฟุตบอลแบบเด็กชาย (ADJ)  
2) gamine เด็กหญิงที่ชอบเล่นฟุตบอลแบบเด็กชาย (N)

Enter token > CROON  
Found 3 token at 13684 - 13686  
1) croon การอิมเพรส (N)  
2) croon อิมเพรส (VI)  
3) croon อิมเพรส (VT)

Enter token > favorite  
Found 4 token at 20582 - 20585  
1) favorite ซึ่งเป็นที่โปรดปราน (ADJ)  
2) favorite คนโปรด (N)  
3) favorite ความนิยมชอบ (N)  
4) favorite ด้านข้าง (N)

หลังจากนั้นจะให้ผู้ใช้ป้อนคำศัพท์ที่ต้องการหา และโปรแกรมจะนำไปค้นหาใน Array List และบอกว่าเจอที่คำ ที่ Index ไหนและ แสดงผลออกมาให้ดู

```

Enter token > acid rain
Found 1 token at 475 - 475
1) acid rain พจนานุกรม(N)
Enter token > get off
Found 35 token at 23963 - 23997
1) get off สัม(PhRV)
2) get off สัม (บางสิ่ง) ใดดีหรือไม่ดี(PhRV)
3) get off สัมเป็นจริง(PhRV)
4) get off สัมพิเศษ(PhRV)
5) get off สัมทำงาน(PhRV)
6) get off สัมรู้(PhRV)
7) get off เลิกเมื่อไหร่(PhRV)
8) get off เลิกงาน(PhRV)
9) get off เลิกทำให้ฉันรำคาญ(PhRV)
10) get off เอาออกไป(PhRV)
11) get off แต่งงาน(PhRV)
12) get off รับผิดชอบ(PhRV)
13) get off ไปให้พ้น(PhRV)
14) get off ไม่เชื่อหรือ(PhRV)
15) get off ไม่โดนเลี้ยงดู(PhRV)
16) get off ช่วยเหลือให้ออกมาจาก (สิ่งที่กำลังจะ)(PhRV)
17) get off ตื่นเต้น(PhRV)
18) get off ทำความสะอาด(PhRV)
19) get off ทำผล(PhRV)
20) get off บอกให้เลิกทำหรือไม่ทำ(PhRV)
21) get off ปิดความลับของคุณ(PhRV)
22) get off ทัก(PhRV)
23) get off มีวิเศษ(PhRV)
24) get off ขอลงมา(PhRV)
25) get off ขอรับ (การควบคุม)(PhRV)
26) get off รอดพ้น(PhRV)
27) get off รอดพ้นอันตราย(PhRV)
28) get off ลงมาจาก(PhRV)
29) get off สัม(PhRV)
30) get off ส่ง(PhRV)
31) get off หนีรอดจาก(PhRV)
32) get off หลับ(PhRV)
33) get off ออกเดินทาง(PhRV)
34) get off ออกจาก (รถ)(PhRV)
35) get off ออกจากรถ(PhRV)

```

กรณีนี้เมื่อป้อนเว้นวรรคมากกว่า 1 ตัวทั้งหน้า  
หลังและตรงกลางนั้น ด้านหน้าและด้านหลังจะ  
ถูก trim() ออกไปและ ตรงกลางจะถูก replace  
ด้วยเว้นวรรคเพียงตัวเดียวทำให้จาก  
( acid rain )-> (acid rain)  
แล้วก็ไปค้นหาใน ArrayList ตามปกติเหมือน  
กรณีด้านบน

กรณีนี้ไม่เจอใน ArrayList โปรแกรมจึงบอกไม่เจอ

```

Enter token > cpe
Not Found token cpe in Dictionary.
Enter token > end
Found 9 token at 18763 - 18771
1) end เป้าหมาย(N)
2) end ขอบเขต(N)
3) end ความตาย(N)
4) end ตอนจบ(N)
5) end ส่วนที่เหลือ(N)
6) end ส่วนปลายของวัตถุ(N)
7) end ทำให้สิ้นสุด(VI)
8) end มีผลสรุป(VI)
9) end ให้เริ่มสุด(VT)

```

End Program.  
This program is written by Sorathorn Kaewchotchuankul 63070501067 CPE/1

กรณีนี้ก็หาคำว่า end ใน Array List ตามปกติเมื่อหาเสร็จแล้วมันจะไปเช็คเงื่อนไข  
ของ loop Do{}While แล้วจะพบว่า String ที่รับมานั้นคือ end ทำให้โปรแกรม  
หลุดจาก loop และจบการทำงาน

- สรุปความเข้าใจของตนเอง

ใน Assignment นี้ต้องฝึกการใช้ ArrayList ในการจัดการกับข้อมูลโดยอ่านไฟล์เข้ามาและนำไปเก็บ โดยสร้าง Class มาหนึ่ง class ไว้เป็น Structure ของข้อมูลแล้วนำข้อมูลไปวิเคราะห์ หาคำซ้ำ ลบคำซ้ำ เรียงลำดับ หา keyword ที่ซ้ำกันมากที่สุดและแสดงผล และหา keyword ที่ผู้ใช้ต้องการหา และแสดงผลออกมา

- ผลการประเมินตนเอง

ให้ตนเองอยู่ที่ระดับ 80 เพราะสามารถทำงานได้ด้วยตัวเองแต่ก็ไม่ได้ทั้งหมดยังมีบางจุดที่สงสัยและยังไม่เข้าใจจึงถามพี่ TA และถามเพื่อนๆบ้างบางจุดจึงเข้าใจและสามารถทำงานต่อได้จนงานเสร็จและสามารถส่งได้ทันเวลา



63070501067  
SORATHORN  
KAEWCHOTCHUANGKUL

### Grading Rubric

[View Full Rubric](#)

#### Criterion 1

20	40	60	80	100
----	----	----	----	-----

80

#### โครงสร้างข้อมูลเชิงเส้น การจัดการข้อมูลหน่วยความจำแบบ...

20	40	60	80	100
----	----	----	----	-----

80