

Report

Assignment 7 : Tree Set

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(Tree Set) ประกอบด้วย methods ได้แก่
 - Method ReadFile เอาไว้อ่านไฟล์แล้วเพิ่มข้อมูลเข้าไปใน ArrayList
 - Method AddTreeNode เอาไว้เช็คเวลาเพิ่มข้อมูลเข้าไปใน tree
 - Method PrintStats เอาไว้นับจำนวนคำที่มี keyword เหมือนกันอย่างเดียว
 - Method Find Print KeyWord เอาไว้หา keyword ที่ผู้ใช้ต้องการหา
2. สร้าง class ในการทำงานเกี่ยวกับ Structure(โครงสร้างข้อมูล) โดยเฉพาะและสร้าง method ได้แก่
 - method ที่ใช้ Compare และส่งค่าให้ Library ไป Search และ contains ข้อมูล
 - constructor(มีข้อมูล) แบ่งข้อมูลใน structure เป็นส่วนๆ

- Source code

- Class DictTree(main)

- method Read_File

```

10 import java.io.BufferedReader;
8
9 public class DictTree {
10     static TreeSet<Dnode> dict = new TreeSet<Dnode> (); // สร้างต้นไม้ Array list แบบ global มาเก็บข้อมูลคำความหมาย
11     static int count1=0;
12     public static void Read_File() { // ฟังก์ชันในการอ่านไฟล์
13         String str;
14         try { // try catch เพื่อจับ error!
15             FileInputStream fcsv = new FileInputStream("C:\\Users\\LENOVO\\Desktop"
16                 + "\\CPE-YEAR#1\\CPE SUBJECT\\CPE111 DATA STRUCTURE\\Assignment 6\\UTF8_Lexitron.csv"); // สร้างไฟล์เก็บคำกับคำความหมาย Encoding
17             InputStreamReader utf = new InputStreamReader(fcsv,"UTF-8"); // รับ Encoding
18             BufferedReader buff = new BufferedReader(utf); // class bufferreader อ่านข้อมูล BOM ที่
19             buff.read(); // อ่าน BOM ที่ (BOM มีขนาดเท่ากับ char เลยใช้ .read() แทนอ่าน char )
20             while((str = buff.readLine()) != null) { // จนจบอ่านไฟล์
21                 AddTreeNode(str); // check node have been ?
22                 count1++; // นับจำนวน
23             }
24             fcsv.close(); // ปิดไฟล์ที่ได้อ่านเสร็จ
25         }
26         catch(Exception e){ // ถ้าเกิด Error!
27             System.out.printf("Error! Can't Read File %n"); // แสดงข้อความ
28         }
29     }

```

- method AddTreeNode & print_stats

```

30     public static void AddTreeNode(String str){
31         Dnode x = new Dnode(str);
32         if(dict.contains(x)) { // use compareTo function to contains word only
33             TreeSet<Dnode> z = (TreeSet<Dnode>) dict.subSet(x, true, x, true); // search from x to x and get x too (mean just 1 node)
34             if(!z.first().mean.contains(x.mean.get(0))) // check repeat meaning
35                 z.first().mean.add(x.mean.get(0)); // add new meaning
36         }
37         else // if didn't have x node in tree before add x to new node
38             dict.add(x);
39     }
40     public static void print_stats() { // show every stats of tutorial
41         int sum = 0,max = 0,i=0;
42         Dnode x = new Dnode();
43         System.out.printf("Keyword in TreeSet : %d \n",dict.size()); // show keyword
44         for(Dnode itr : dict) { // loop to count meaning
45             sum += itr.mean.size(); // sum every meaning
46             if(max < itr.mean.size()) { // find maximum meaning in Tree
47                 max = itr.mean.size(); // maximum
48                 x = itr; // remember node that have maximum meaning
49             }
50         }
51         System.out.println("Meaning in TreeSet : "+ sum); // show amount of meaning
52         System.out.println("Most Keyword Found : "+x.word+" \ncount : "+max); // show most keyword
53         for(String show : x.mean) { // loop to printf different meaning
54             System.out.printf("\t\t%d) %s " + show + " \n",++i,x.word);
55         }
56     }

```

➤ Print_token

```
70 public static void Find_print_keyword(String str) { // find keyword that users needs
71     Dnode x = new Dnode();
72     int i=0;
73     x.word = str; // put string from users in Node to search node in Tree
74     if(dict.contains(x)) { // check String is in Tree?
75         TreeSet<Dnode> z = (TreeSet<Dnode>) dict.subSet(x,true,x,true); // if string is in Tree
76         System.out.printf("Ans > Found %s in BinaryTree & Have %d meaning\n",str,z.first().mean.size()); // show users
77         for(String print : z.first().mean) { // loop to print different meaning
78             System.out.printf("\t\t%d) %s %s \n",++i,str,print);
79         }
80     }
81     else System.out.printf("Not Found %s in BinaryTree !!\n",str); // if string isn't in in tree
82 }
```

➤ Main (method ที่ใช้รัน)

```
70 public static void main(String[] args) { // class use to run
71     String str;
72     Scanner in = new Scanner(System.in);
73     Read_File(); // อ่านข้อมูล
74     System.out.println("Total Read = "+count1+" records."); // show total read of File
75     print_stats(); // show every stats
76     do { // loop to find keywords that users need
77         System.out.printf("Enter token > ");
78         str = in.nextLine(); // อ่านข้อมูล
79         str = str.trim().replaceAll("\\s+", " "); // เปลี่ยนเว้นวรรคให้เป็นตัวว่างด้วยวิธีที่รวดเร็วด้วย
80         Find_print_keyword(str); // send string to search in tree
81     }while(!str.equalsIgnoreCase("end")); // ทำซ้ำ ถ้า str ไม่ใช่ end
82     System.out.println("End Program.");
83     System.out.printf("This program is written by Sorathorn Kaewhotchuangkul 63070501067 CPE/1");
84 }
85 }
86 }
```

- Class ที่ทำงานเกี่ยวกับข้อมูล (Dnode)

```
1 import java.util.ArrayList;
2
3 public class Dnode implements Comparable<Dnode>{
4     String word;
5     ArrayList<String> mean;
6     public Dnode() { // constructor แบบ node เปล่า
7         word = "";
8         mean = new ArrayList<String>();
9         mean.add("");
10    }
11
12    public int compareTo(Dnode x) { // Comparable ของ Sort และ binary search
13        return (int) this.word.compareToIgnoreCase(x.word); // รีটারน์เป็น int
14    }
15
16    public Dnode(String buff) { // constructor แบบมีข้อมูล
17        String meanscan;
18        buff = buff.trim().replaceAll("\\s+", " "); // ลบ white space หน้า หลัง ตรงกลางที่เป็น white space ยาวๆก็ให้เหลืออันเดียว
19        String [] str = buff.split(","); // split มันจาก ,
20        word = str[0];
21        meanscan = str[1] + "(" + str[2] + ")"; // gather meaning and type
22        mean = new ArrayList<String>(); // reserve a memory of mean
23        mean.add(meanscan); // add mean + type in ArrayList
24    }
25 }
```

Note! อธิบายโค้ด comment อยู่ในตัวโค้ด

- Test case & อธิบาย

```
Total Read = 74233 records.
Keyword in TreeSet : 45921
Meaning in TreeSet : 73981
Most Keyword Found : get off
count : 35

1) get off เริ่ม (PHRV)
2) get off เริ่ม (บางสิ่ง) ได้ดีหรือไม่ดี (PHRV)
3) get off เริ่มไม่จริง (PHRV)
4) get off เริ่มไม่ได้ (PHRV)
5) get off เริ่มทำงาน (PHRV)
6) get off เริ่มแล้ว (PHRV)
7) get off เลิกเมื่อไหร่ (PHRV)
8) get off เลิกงาน (PHRV)
9) get off เลิกทำใช้เงินจำนวนมาก (PHRV)
10) get off เอาออกไป (PHRV)
11) get off ผิดงาน (PHRV)
12) get off ไปถึงมาก (PHRV)
13) get off ไปไหน (PHRV)
14) get off ไม่เชื่อหรือ (PHRV)
15) get off ไม่ได้อะไรเลย (PHRV)
16) get off ช่วยเหลือให้ออกมาจาก (เรื่องที่กำลัง) (PHRV)
17) get off ตื่นเต้น (PHRV)
18) get off ทำความสะอาด (PHRV)
19) get off ทำผิด (PHRV)
20) get off บอกให้เลิกทำหรือเลิกทำ (PHRV)
21) get off ปิดความลับของคุณ (PHRV)
22) get off ทำ (PHRV)
23) get off มีวันหยุด (PHRV)
24) get off บอกมา (PHRV)
25) get off บอกฉัน (การควบคุม) (PHRV)
26) get off รอฉัน (PHRV)
27) get off รอฉันแล้ว (PHRV)
28) get off ลงมาจาก (PHRV)
29) get off สัม (PHRV)
30) get off ส่ง (PHRV)
31) get off หนีออกจาก (PHRV)
32) get off หลับ (PHRV)
33) get off ออกเดินทาง (PHRV)
34) get off ออกจาก (รถ) (PHRV)
35) get off ออกจาก (PHRV)
```

เมื่อเริ่มโปรแกรมจะเริ่มอ่านไฟล์และนับจำนวนที่อ่านได้ จำนวน **Word** ใน **Tree** ความหมายใน **Tree** และ **word** ที่มี **meaning** เยอะที่สุดใน **tree** และแสดง **meaning** ออกมาให้เห็นด้วย

```
Enter token > a
Ans > Found a in BinaryTree & Have 1 meaning
1) a อักษรตัวแรกในภาษาอังกฤษ (N)
Enter token > zymurgy
Ans > Found zymurgy in BinaryTree & Have 1 meaning
1) zymurgy การหมักสุรา (N)
Enter token > Gamine
Ans > Found Gamine in BinaryTree & Have 2 meaning
1) Gamine (เด็กหญิง) ซึ่งเล่นตุ๊กตาแบบเด็กชาย (ADJ)
2) Gamine เด็กหญิงที่ชอบเล่นตุ๊กตาแบบเด็กชาย (N)
Enter token > CROON
Ans > Found CROON in BinaryTree & Have 3 meaning
1) CROON การอิมพอง (N)
2) CROON อิมพอง (VI)
3) CROON อิมพอง (VT)
Enter token > favorite
Ans > Found favorite in BinaryTree & Have 4 meaning
1) favorite ซึ่งเป็นที่โปรดปราน (ADJ)
2) favorite คนโปรด (N)
3) favorite ความนิยมชื่นชอบ (N)
4) favorite ตัวถึง (N)
```

เมื่อแสดงสถิติต่างๆไปตอนเริ่มแล้วก็จะให้ผู้ใช้ใส่ **token** ที่ต้องการค้นหาใน **Tree** หลังจากนั้นโปรแกรมจะนำไปค้นหาและแสดงความหมายของ **token** ตัวนั้นออกมาด้วยเหมือนกับ **testcase** นี้

```

Enter token > acid rain
Ans > Found acid rain in BinaryTree & Have 1 meaning
1) acid rain (N)
Enter token > get off
Ans > Found get off in BinaryTree & Have 35 meaning
1) get off (PHRV)
2) get off (บางสิ่ง) (PHRV)
3) get off (เริ่มเป็นจริง) (PHRV)
4) get off (เริ่มดื่ม) (PHRV)
5) get off (เริ่มทำงาน) (PHRV)
6) get off (เริ่มรู้) (PHRV)
7) get off (เลิกยุ่ง) (PHRV)
8) get off (เลิกงาน) (PHRV)
9) get off (เลิกทำโน้มน้าวใจ) (PHRV)
10) get off (เอาออกไป) (PHRV)
11) get off (แต่งงาน) (PHRV)
12) get off (ไม่สนใจ) (PHRV)
13) get off (ไปไหน) (PHRV)
14) get off (ไม่เชื่อ) (PHRV)
15) get off (ไม่ได้เลย) (PHRV)
16) get off (ช่วยเหลือให้ออกมาจาก (เรื่องที่กำลัง) (PHRV)
17) get off (ดื่ม) (PHRV)
18) get off (ทำความผิด) (PHRV)
19) get off (ทำผิด) (PHRV)
20) get off (บอกให้เลิกทำหรือเลิก) (PHRV)
21) get off (ข้อความลับ) (PHRV)
22) get off (พัก) (PHRV)
23) get off (มีเหตุผล) (PHRV)
24) get off (ยกเลิก) (PHRV)
25) get off (ยอมรับ (การควบคุม) (PHRV)
26) get off (รอดพ้น) (PHRV)
27) get off (รอดพ้นอันตราย) (PHRV)
28) get off (ลงมาจาก) (PHRV)
29) get off (ดื่ม) (PHRV)
30) get off (ส่ง) (PHRV)
31) get off (หนีออกจาก) (PHRV)
32) get off (หลับ) (PHRV)
33) get off (ออกเดินทาง) (PHRV)
34) get off (ออกจาก (รถ) (PHRV)
35) get off (ออกจากรถ) (PHRV)

```

Testcase นี้ก็ไม่มีอะไรต่างจากเดิมมากเพียงแต่มีการ
เพิ่มเว้นวรรคเข้ามาตอนใส่ **token** แต่โปรแกรมก็
สามารถตัดเว้นวรรคที่ไม่เกี่ยวข้องออกได้จากการ
trim() และ **replaceAll()** ทำให้สามารถค้นหาใน
Tree ได้ตามปกติเหมือนเดิม

Testcase นี้ก็ไม่สามารถค้นหาเจอใน Tree
เนื่องจาก **contains** ไม่พบตัวที่ต้องการค้นหาทำให้
โปรแกรมแสดงข้อความว่า ไม่เจอ **token** นั้น

```

Enter token > cpe
Not Found cpe in BinaryTree !!

```

```

Enter token > end
Ans > Found end in BinaryTree & Have 9 meaning
1) end (เข้าพบ) (N)
2) end (ขอบเขต) (N)
3) end (ความหมาย) (N)
4) end (ตอนจบ) (N)
5) end (ส่วนที่เหลือ) (N)
6) end (ส่วนปลายของวัตถุ) (N)
7) end (ทำให้อันตร) (VI)
8) end (มีผลสรุป) (VI)
9) end (ทำให้อันตร) (VT)

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

```

Testcase นี้ก็ค้นหาตามปกติแล้วเจอจึงแสดง
meaning ออกมาและหลังจากนั้นก็จบโปรแกรม
เนื่องจาก **Loop** นั้นหยุดทำงานเนื่องจากเงื่อนไขของ
String ที่รับมานั้นเป็นคำว่า **"end"**

- สรุปความเข้าใจของตนเอง

ใน Assignment นี้ต้องฝึกการใช้ TreeSet ในการจัดการกับข้อมูลโดยอ่านไฟล์เข้ามาและนำไปเก็บ โดยสร้าง Class มาหนึ่ง class ไว้เป็น Structure ของข้อมูลแล้วนำข้อไปผูกในไปวิเคราะห์จัดเก็บใน Tree, หา keyword ที่ซ้ำกันมากที่สุด, และแสดงผล และหา keyword ที่ผู้ใช้ต้องการหาและแสดงผลออกมา

- ผลการประเมินตนเอง

ให้ตนเองอยู่ที่ระดับ 80 เพราะสามารถทำงานได้ด้วยตัวเองแต่ก็ไม่ได้ทั้งหมดยังมีบางจุดที่สงสัยและยังไม่เข้าใจจึงถามพี่ TA และถามเพื่อนๆบ้างบางจุดจึงเข้าใจและสามารถทำงานต่อได้จนงานเสร็จและยังเข้าใจเนื้อหาของ Tree พอสมควรทำให้สามารถทำ Assignment นี้เสร็จส่งได้ทันเวลา



63070501067
SORATHORN
KAEWCHOTCHUANGKUL

Grading Rubric

[View Full Rubric](#)

Criterion 1



80

โครงสร้างข้อมูลแบบต้นไม้



80