



ค้นหารายชื่อผู้มีสิทธิ์เข้าสอบจากฐานข้อมูล
(618240)

จัดทำโดย

นางสาวอมรรัตน์ หาญเนื่องนิตย์ รหัสนักศึกษา 640910640
นายณชพล อุบล รหัสนักศึกษา 640911019

เสนอ

ผู้ช่วยศาสตราจารย์ ดร.ยุพธนา เจริญดา

รายงานเล่มนี้เป็นส่วนหนึ่งของวิชา 618240 โครงสร้างข้อมูลและอัลกอริทึม
คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม ภาควิชาไฟฟ้า
ภาคเรียนที่ 1 ปีการศึกษา 2565
มหาวิทยาลัยศิลปากร วิทยาเขตพระราชวังสนามจันทร์

คำนำ

รายงานเล่มนี้จัดทำขึ้นเพื่อเป็นส่วนหนึ่งของวิชาโครงสร้างข้อมูลและอัลกอริทึม เพื่อให้ได้ศึกษาหาความรู้ในเรื่องราวของการเขียนโปรแกรมเพื่อเอาไปใช้ประโยชน์หรือทำให้เกิดความสะดวกสบายในการหาข้อมูลต่าง ๆ โดยได้ศึกษาผ่านแหล่งความรู้ต่าง ๆ อาทิเช่น ตำรา หนังสือ หรือสื่อการเรียนการสอน โดยรายงานเล่มนี้ต้องมีเนื้อหาเกี่ยวกับการเขียนโปรแกรมเอาไว้ใช้เรียงลำดับการเข้าเรียนจากมากไปน้อยและดูการเข้าเรียนในชั้นปีการศึกษานั้นว่าเข้าเรียนทั้งหมดกี่ครั้ง ถ้าเกิดว่ามีจำนวนไม่ครบพอรันโปรแกรมก็จะไม่ขึ้นรายชื่อของนักศึกษา

ผู้จัดทำคาดหวังเป็นอย่างยิ่งว่าการจัดทำเอกสารฉบับนี้จะมีข้อมูลที่เป็นประโยชน์ต่อผู้สนใจที่จะศึกษาโปรแกรมจาวาเป็นอย่างดี

นักศึกษาชั้นปีที่ 2

15 ตุลาคม 2565

สารบัญ

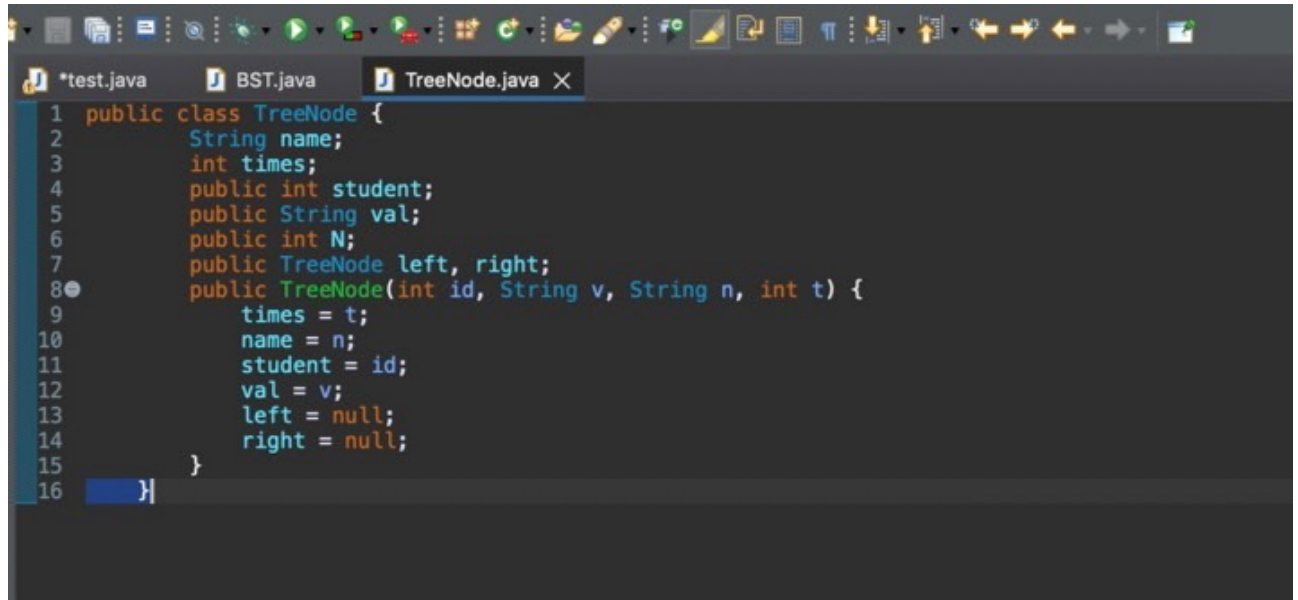
ที่มาของ Project code	1
TreeNode	2
Binary search Tree (BST)	3
Test	5
ผลการรัน	7

ที่มาของ Project code

Project code ได้อิงมาจาก Binary search Tree (BST) หรือชื่อภาษาไทยเรียกว่า ต้นไม้ทวิภาค เป็นการจัดเก็บข้อมูลรูปแบบหนึ่งที่มีประสิทธิภาพโดยเฉพาะการเพิ่ม ลบ ค้นหา หาตัวมากที่สุด หรือตัวที่น้อยที่สุด มีลักษณะการเก็บข้อมูลเป็นโหนด คล้ายกับ Linked List แต่ไม่ได้เก็บลักษณะเป็นเส้นตรงเหมือนกัน เพราะ BST จะเก็บข้อมูลในลักษณะที่ว่าข้อมูลทางขวาจะมากกว่าตัวตรงกลาง และทางซ้าย แต่ละโหนดมีลูกได้ไม่เกิน 2 โหนด การเรียงในลักษณะนี้ทำให้ BST ได้เปรียบในเรื่องของการค้นหาข้อมูล เพราะข้อมูลที่เก็บไว้ได้อยู่ในรูปแบบง่ายต่อการค้นหาอยู่แล้ว

Code Java

TreeNode



```
1 public class TreeNode {
2     String name;
3     int times;
4     public int student;
5     public String val;
6     public int N;
7     public TreeNode left, right;
8     public TreeNode(int id, String v, String n, int t) {
9         times = t;
10        name = n;
11        student = id;
12        val = v;
13        left = null;
14        right = null;
15    }
16 }
```

คำอธิบาย

บรรทัดที่ 1 : สร้างคลาสโหนดสำหรับต้นไม้ ตั้งชื่อว่า TreeNode

บรรทัดที่ 2 : ตัวแปรของชื่อนักศึกษา

บรรทัดที่ 3 : ตัวที่แสดงจำนวนตัวเลขของครั้งที่เข้าเรียนในภาคการศึกษานั้น ๆ

บรรทัดที่ 4 : สร้างตัวแปร student เก็บข้อมูลเป็นจำนวนเต็ม

บรรทัดที่ 9 : เป็นการแทนค่าตัวแปร โดยที่ times เราจะแทนเป็นจำนวนครั้งที่
เข้าเรียนในภาคปีการศึกษานั้น ๆ

บรรทัดที่ 10 : เป็นการแทนค่าตัวแปร โดยที่ name เราจะแทนเป็นชื่อของนักศึกษา

บรรทัดที่ 11 : เป็นการแทนค่าตัวแปร โดยที่ student เราจะแทนเป็นรหัสนักศึกษา

บรรทัดที่ 13 : ตัวแปร left ใส่ค่าเริ่มต้นเป็น null

บรรทัดที่ 14 : ตัวแปร right ใส่ค่าเริ่มต้นเป็น null

Binary search Tree BST

```
1 public class BST {
2     private TreeNode root;
3     BST() { root = null; }
4     private int size(TreeNode x) {
5         if(x == null) return 0;
6         else return x.N;
7     }
8     private TreeNode put(TreeNode x, int key, String name, int times) {
9         if(x == null) return new TreeNode(key, name, name, times);
10        if(key < x.student) x.right = put(x.right, key, name, times);
11        if(key > x.student) x.left = put(x.left, key, name, times);
12        x.N = size(x.left) + size(x.right) + 1;
13        return x;
14    }
15    public void put(int key, String name, int times) {
16        root = put(root, key, name, times);
17    }
18    public void inorder() { inorder(root); }
19    public void inorder(TreeNode x) {
20        if(x != null) {
21            inorder(x.left);
22            System.out.println(x.student + " " + x.name + " " + x.times);
23            inorder(x.right);
24        }
25    }
26    private TreeNode deleteMin(TreeNode x) {
27        if (x.right == null) return x.left;
28        x.right = deleteMin(x.right);
29        x.N = size(x.right) + size(x.left) + 1;
30        return x;
31    }
32    public void deleteMin() {
33        root = deleteMin(root);
34    }
35 }
```

คำอธิบาย

บรรทัดที่ 1 : สร้างคลาสโหนดสำหรับต้นไม้ ตั้งชื่อว่า BST

บรรทัดที่ 2 : สร้างโหนดรากชื่อ root ขึ้นมาเป็นข้อมูลชนิด BST

บรรทัดที่ 5 : ถ้าตัวแปร x เท่ากับ null หมายถึง ยังไม่มีรากก็เท่ากับยังไม่มีข้อมูล
ตัวอื่น ๆ ทั้งหมด ให้คืนค่าเป็น 0

บรรทัดที่ 8 : เป็นการกำหนดการแสดงผลข้อมูลตัวแปรหน้าโปรแกรม ซึ่งจะมี int key

บรรทัดที่ 9 : ถ้าตัวแปร x เท่ากับ null หมายถึง ยังไม่มีรากก็เท่ากับยังไม่มี
ข้อมูลตัวอื่น ๆ ทั้งหมด ให้คืนค่าใหม่เป็น TreeNode ที่แสดง
จำนวนตัวเลขนักศึกษาชื่อจริงและจำนวนครั้งที่เข้าเรียน

บรรทัดที่ 10 – 11 : จะแสดงข้อมูลหน้าโปรแกรมว่าเราจะหาค่าจำนวนครั้งที่เข้าเรียน
จากน้อยไปมาก แต่ถ้าหากว่ามีจำนวนการเข้าเรียนไม่ครบ
พอแสดงค่าข้อมูลจะไม่ขึ้นข้อมูลของนักศึกษา

บรรทัดที่ 15 : เป็นการเรียงข้อมูลและป้อนค่าแป้นพิมพ์ตัวแปรที่เราได้กำหนดไว้ใน
class ของ TreeNode

บรรทัดที่ 18 : เป็นการเรียงข้อมูลของ TreeNode

บรรทัดที่ 22 : เป็นการแสดงข้อมูลจำนวนครั้งที่เข้าเรียน ชื่อนักศึกษาและรหัสนักศึกษา

บรรทัดที่ 26 : เป็นการกำหนดแสดงข้อมูลการลบค่าน้อยที่สุดออก

บรรทัดที่ 27 : ถ้าค่าทางขวาเท่ากับ null หมายถึง ยังไม่มีรากก็เท่ากับยังไม่มีข้อมูล
ตัวอื่น ๆ จะเป็นตัวทางซ้ายทั้งหมด

บรรทัดที่ 32 : เป็นการเรียงข้อมูลของ deleteMin

Test

```
1 import java.util.Scanner;
2 public class test {
3     public static void main(String[] args) {
4         System.out.println("Subject java Section 1 :");
5         System.out.println("Times, Name Student Id Student : \n");
6         BST t1 = new BST();
7         t1.put(51, "Amonrat Hannueangnit", 640910640);
8         t1.put(50, "Nachapol Ubon", 640911019);
9         t1.put(53, "Kantinan Kimkitiklvilai", 640910268);
10        t1.put(48, "Jetawat Koosuan", 640910623);
11        t1.put(54, "Jaronrat Rattanawichai", 640910484);
12        t1.put(56, "Teekatat Aunjitti", 640910288);
13        t1.deleteMin();
14        t1.inorder();
15        System.out.println("\n");
16        System.out.println("Subject java Section 2 :");
17        System.out.println("Times, Name Student Id Student : \n");
18        BST t2 = new BST();
19        t2.put(51, "Kumpon Yawawong", 640910269);
20        t2.put(56, "Jessada Mali", 640910624);
21        t2.put(50, "Sukonta Nanthawisod", 640310321);
22        t2.put(49, "Pitchporn Supol", 640910303);
23        t2.put(54, "Akanut Puengchaipat", 640910328);
24        t2.put(59, "Methawi Jaisue", 640910311);
25        t2.deleteMin();
26        t2.inorder();
27        System.out.println("\n");
28        System.out.println("Subject java Section 3 :");
29        System.out.println("Times, Name Student Id Student : \n");
30        BST t3 = new BST();
31        t3.put(53, "Thanatcha Phengleng", 640910631);
32        t3.put(51, "Puttipong Senaprom", 640911050);
33        t3.put(40, "Sasivimon Mokluang", 640910635);
34        t3.put(52, "Chanikarn Kannamee", 640910625);
35        t3.put(50, "Theewaraporn Thepsuwan", 640911033);
36        t3.put(59, "Thanawat Srisawang", 640910490);
37        t3.deleteMin();
38        t3.inorder();
39        System.out.println("\n");
40        Scanner input = new Scanner(System.in);
41        System.out.print("Enter Section 1 :");
42        double a = input.nextDouble();
43        System.out.print("Enter Section 2 :");
44        double b = input.nextDouble();
45        System.out.print("Enter Section 3 :");
46        double c = input.nextDouble();
47        int area = (int)(a+b+c);
48        System.out.println(area);
49    }
50 }
51
```

คำอธิบาย

บรรทัดที่ 4 : คือการแสดงผลข้อมูลเพื่อให้รู้ว่าเป็นรายวิชาอะไรและSection อะไร

บรรทัดที่ 5 : คือการแสดงผลข้อมูลเพื่อให้รู้ถึงจำนวนครั้งที่เข้าเรียน,ชื่อนักศึกษาและรหัสนักศึกษา

บรรทัดที่ 6 : ใส่ t1 เพื่อเป็นการจัดกลุ่มให้กับ BST และ t1,t2,t3 หมายถึง section1,2,3

บรรทัดที่ 13 : เป็นการลบจำนวนน้อยที่สุดออก

บรรทัดที่ 14 : เป็นการเรียงสมาชิกใน section ที่ 1

บรรทัดที่ 15 : คือการรับค่าตัวเลขจากแป้นพิมพ์

บรรทัดที่ 41 : รับค่าตัวเลขเพื่อที่จะได้ป้อนข้อมูลจำนวนนักศึกษาที่ยังมีอยู่ในระบบ
ในแต่ละ section

บรรทัดที่ 42 : รับค่าของบรรทัดที่ 41 เพื่อไปการหาค่าของ section ต่อไป

บรรทัดที่ 47 : การโชว์ผลลัพธ์ของสมการที่อยากรู้ค่าทั้งหมดของทั้ง 3 Section

บรรทัดที่ 48 : โชว์ค่าผลลัพธ์ของ Section ทั้งหมด

ผลการรัน

```
<terminated> test [Java Application] /Users/orn/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64.17.0.4.v2022
Subject java Section 1 :
Times, Name Student Id Student :

56 Teekatat Aunjitti      640910288
54 Jaroonrat Rattanawichai 640910484
53 Kantinan Kimkitiklilai 640910268
51 Amonrat Hannueangnit   640910640
50 Nachapol Ubon          640911019

Subject java Section 2 :
Times, Name Student Id Student :

59 Methawi Jaisue        640910311
56 Jessada Mali          640910624
54 Akanut Puengchaipat   640910328
51 Kumpon Yawawong       640910269
50 Sukonta Nanthawisod   640310321

Subject java Section 3 :
Times, Name Student Id Student :

59 Thanawat Srisawang    640910490
53 Thanatcha Phengleng   640910631
52 Chanikarn Kannamee    640910625
51 Puttipong Senaprom    640911050
50 Theewaraporn Thepsuwan 640911033

Enter Section 1 :7
Enter Section 2 :9
Enter Section 3 :10
26
```

คำอธิบาย

เป็นการแสดงข้อมูลว่าในแต่ละ section จำนวนนักศึกษาที่แต่ละ section มีทั้งหมดกี่คนที่มีจำนวนครบ 50 ครั้งต่อ 1 ภาคเรียน โดยที่เราจะกำหนดว่าใน 1 ภาคเรียนจะต้องเข้าเรียนมากกว่า 50 ครั้ง หากไม่ครบตามจำนวนที่กำหนด พอแสดงผลข้อมูลจะไม่พบข้อมูลของนักศึกษาดังกล่าว และนำจำนวนทั้งหมดของแต่ละ section มาบวกกัน จะได้ข้อมูลของนักศึกษาทั้งหมดที่ได้เข้าสอบในรายวิชานั้น ๆ โค้ดนี้ทำขึ้นมาเพื่อประหยัดเวลา และมีความแม่นยำค่อนข้างสูง และสะดวกสบายต่ออาจารย์ผู้สอน

บรรณานุกรม

https://expert-programming-tutor.com/tutorial/data_structure/06_1_%E0%B9%84%E0%B8%9A%E0%B8%99%E0%B8%B2%E0%B8%A3%E0%B8%B5%E0%B8%97%E0%B8%A3%E0%B8%B5.php