

Project Array Part I

COS2103

Data structure and Algorithm

1 / 2568

Present to

Asst.Prof.DR.Chouvalit Khancome

By Krittaya Tantichaiyakul

6705500269

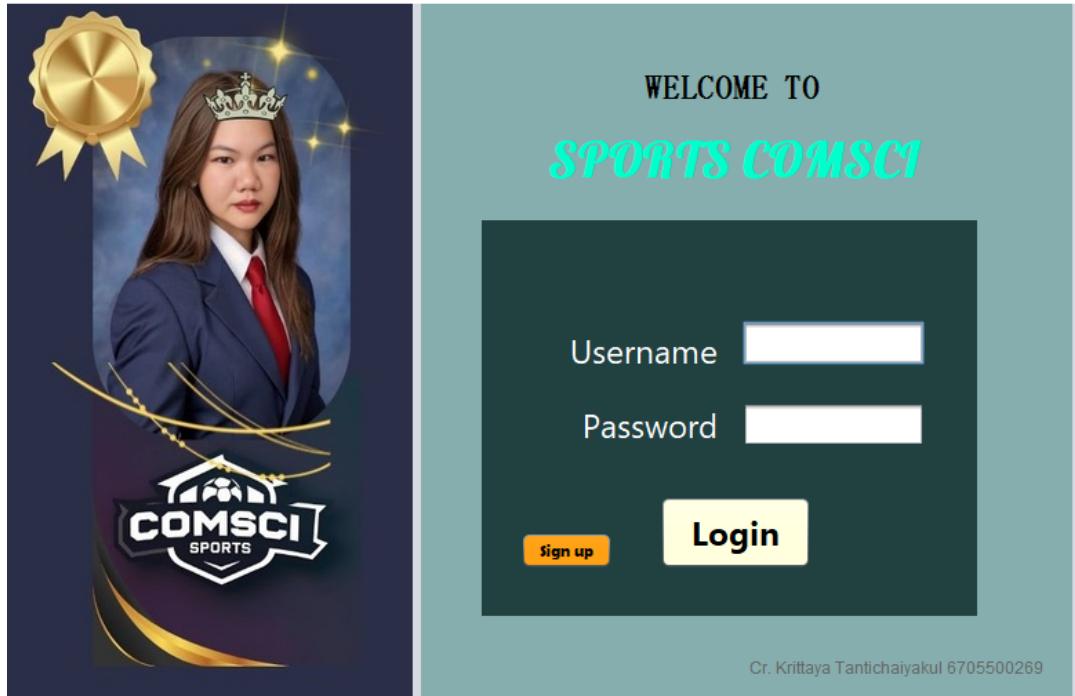
หัวข้อ

อุปกรณ์กีฬา



Program Interface Preview

1. Login page - Login



2. Category of array - Menu

The menu interface has a dark background. On the left, there's a sidebar with categories: Sneaker, Soccer, Golf, Racket, and Ball. The central area shows a product detail for an Adidas sneaker: Product code: 10233, Name: Adidas sneaker, Type: 1, Price: 4600.0 baht. Navigation buttons "Back" and "Next" are at the bottom. On the right, there's a profile picture of a person, the text "Host : Krittaya001", a "Manage data" button, and a "Stock" section showing Storage: 15, Amount: 13, and List no.: 1. At the bottom, there's a footer with the letters [K][R][I][T][T][A] and numbers 0, 1, 2, 3, 4, 5, and the text "Krittaya Tantichaiyakul 6705500269".

3. list of products – Manage

No.	Code	Name	Price	Type	Address
1	10233	Adidas sne...	4600.0	1	/sc/sneake...
2	10472	Nike sneaker	4200.0	1	/sc/sneake...
3	10785	Skecher sn...	2500.0	1	/sc/ske.jpg
4	20096	Umbro Glove	800.0	2	/sc/ub.jpg
5	20504	Nike footba...	1500.0	2	/sc/shirtNik...
6	30319	Ping stick g...	1300.0	3	/sc/golfstick...
7	30873	Anyday golf ...	5000.0	3	/sc/golfbag....
8	40152	Yonex strike	2400.0	4	/sc/yon.png
9	40197	Kawasaki s...	1700.0	4	/sc/kawa.jpg
10	40601	Victor shuttle...	200.0	4	/sc/shuttle.j...
11	50290	Puma footb...	800.0	5	/sc/puma.jpg
12	50475	Mikasa voll...	1200.0	5	/sc/volley.jpg
13	50778	Wilson bas...	1500.0	5	/sc/wilson.j...

4. ไฟล์คลาสจัดเก็บข้อมูล – DataStore

```

11  public class DataStore {
12      public static int n=13;
13      final int UB=15;
14      int LB=0;
15      public static class Products {
16          public int code;
17          public String name;
18          public int type;
19          public float price;
20          public String pic;
21
22          public void setAll(int a, String b, int c, float d, String e){
23              code = a;
24              name = b;
25              type = c;
26              price = d;
27              pic = e;
28          }
29          static int getn(){
30              return n;
31          }
32      }
33  }

35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58

```

```

public static Products[] list = new Products[15];
static {
    for (int i = 0; i < list.length; i++) {
        list[i] = new Products();
    }
}

// สำหรับนับจำนวนรายการ
list[0].setAll(10233,"Adidas sneaker",1,4600,"/sc/sneakerAdi.jpg");
list[1].setAll(10472,"Nike sneaker",1,4200,"/sc/sneakerNike.jpg");
list[2].setAll(10785,"Skecher sneaker",1,2500,"/sc/ske.jpg");
list[3].setAll(20096,"Umbro Glove",2,800,"/sc/ub.jpg");
list[4].setAll(20504,"Nike football shirt",2,1500,"/sc/shirtNike.jpg");
list[5].setAll(30319,"Ping stick golf",3,1300,"/sc/golfstick.jpg");
list[6].setAll(30873,"Anyday golf bag",3,5000,"/sc/golfbag.jpg");
list[7].setAll(40152,"Yonex strike",4,2400,"/sc/yon.png");
list[8].setAll(40197,"Kawasaki strike",4,1700,"/sc/kawa.jpg");
list[9].setAll(40601,"Victor shuttlecock",4,200,"/sc/shuttle.jpg");
list[10].setAll(50290,"Puma football",5,800,"/sc/puma.jpg");
list[11].setAll(50475,"Mikasa volleyball",5,1200,"/sc/volley.jpg");
list[12].setAll(50778,"Wilson basketball",5,1500,"/sc/wilson.jpg");
}

```

การทำงานของอัลกอริทึม

1. สร้างไฟล์ DataStore เป็นศูนย์กลางสำหรับจัดเก็บอาร์เรย์ หากไฟล์อื่นต้องการเข้าถึงอาร์เรย์ จำเป็นต้องใช้ผ่านไฟล์นี้ เนื่องจากชุดข้อมูลที่อาจารย์กำหนดมี data type ที่แตกต่างกัน จึงต้องสร้าง Class Products เพื่อจัดเก็บ Attribute ทั้ง 5 สร้างอาร์เรย์ list ขึ้นมาขนาด 15 ช่อง พร้อมกำหนดค่าข้อมูล 13 ช่องคือ 0-12 และสร้างตัวแปร n สำหรับเก็บจำนวนสมาชิกในอาร์เรย์แบบ static โดยกำหนดให้ค่าเริ่มต้นเท่ากับจำนวนที่ใส่ข้อมูลไปคือ 13 ส่วน index ตำแหน่งที่ 13-14 ยังเป็นช่องว่าง เพื่อให้เกิดความหลากหลายในการเล่นกับอัลกอริทึมค่ะ

```
public class DataStore {
    public static int n=13; // 13. ลักษณะ Max
    final int UB=15;
    int LB=0; // Start
    public static class Products {
        public int code;
        public String name;
        public int type;
        public float price;
        public String pic;
        public void setAll(int a, String b, int c, float d, String e) {
            code = a;
            name = b;
            type = c;
            price = d;
            pic = e;
        }
        static int getN() {
            return n;
        }
    }
}
```

```
public static Products[] list = new Products[15];
static {
    for (int i = 0; i < list.length; i++) {
        list[i] = new Products();
    }
}
// กำหนดค่าบุคคลต่อเรขา
list[0].setAll(10233, "Adidas sneaker", 1,4600, "/sc/sneakerAdi.jpg");
list[1].setAll(10472, "Nike sneaker", 1,4200, "/sc/sneakerNike.jpg");
list[2].setAll(10785, "Skecher sneaker", 1,2500, "/sc/ske.jpg");
list[3].setAll(20096, "Umbro Glove", 2,800, "/sc/ub.jpg");
list[4].setAll(20504, "Nike football shirt", 2,1500, "/sc/shirtNike.jpg");
list[5].setAll(30319, "Ping stick golf", 3,1300, "/sc/golfstick.jpg");
list[6].setAll(30873, "Anyday golf bag", 3,5000, "/sc/golfbag.jpg");
list[7].setAll(40152, "Yonex strike", 4,2400, "/sc/yon.png");
list[8].setAll(40197, "Kawasaki strike", 4,1700, "/sc/kawa.jpg");
list[9].setAll(40601, "Victor shuttlecock", 4,200, "/sc/shuttle.jpg");
list[10].setAll(50290, "Puma football", 5,800, "/sc/puma.jpg");
list[11].setAll(50475, "Mikasa volleyball", 5,1200, "/sc/volley.jpg");
list[12].setAll(50778, "Wilson basketball", 5,1500, "/sc/wilson.jpg");
}

list[13] and list[14] = Empty
```

2. เริ่มต้นระบบจะพามาที่หน้า Login ท่านสามารถใส่ username และ password เป็น 1234 ได้เลย ซึ่งจริงๆแล้วหน้ามีภาพในหัวอยู่แล้วว่า สามารถตั้งให้ผู้ใช้ sign in ใหม่ได้ โดยการจัดเก็บรหัสผ่านใหม่ไว้ที่อาร์เรย์ชุดนึง แล้วจึงตรวจสอบที่ปุ่ม Login หาก string ตรงกันกับชุดข้อมูล ก็จะอนุญาตให้เข้าหน้าถัดไปได้ แต่ถ้าทำแบบนั้นก็จะเหมือนว่าเป็น user ไม่ใช่เจ้าของที่จะไปสามารถจัดการชุดข้อมูลข้างในได้ เลยขอเก็บไว้ทำในโปรเจกหลังๆค่ะ 😊

WELCOME TO
SPORTS COMSCI

```

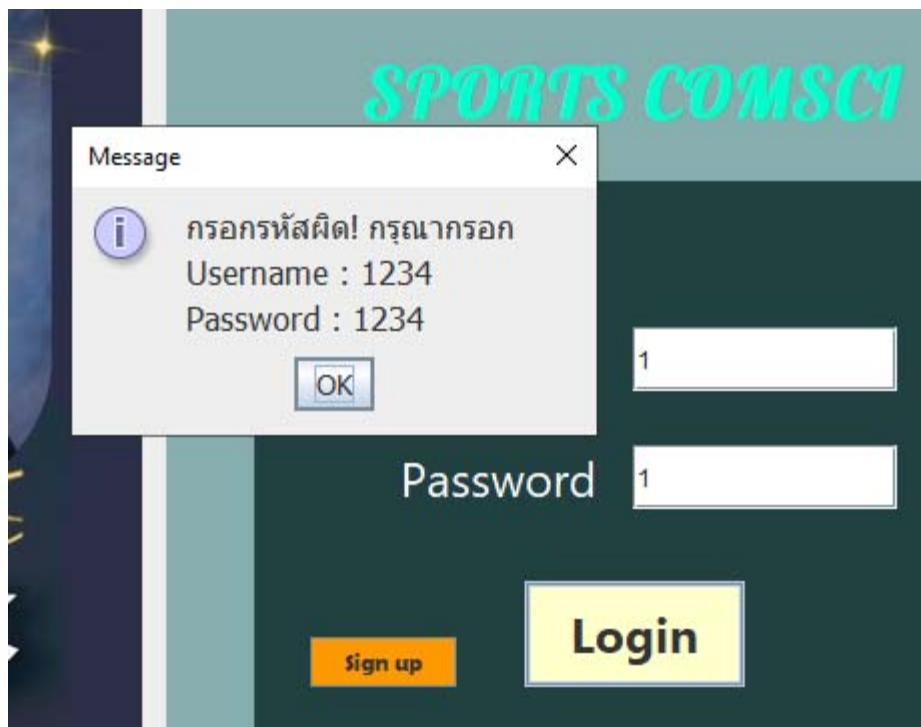
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    int user=Integer.parseInt(jTextField1.getText());
    int pass=Integer.parseInt(jTextField2.getText());
    if(user==1234 && pass==1234){
        System.out.println("Login complete!");
        JOptionPane.showMessageDialog(this, "ล็อกอินสำเร็จ!");
        Menu f2 = new Menu(); // สร้างหน้าจอเมนู
        f2.setVisible(true); // แสดงหน้าจอ
        this.dispose();
    }else{
        JOptionPane.showMessageDialog(this, "กรอกรหัสผิด! กรุณากรอก \n Username : 1234\n Password : 1234");
    }
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    UIManager.put("OptionPane.messageFont", new Font("Tahoma", Font.PLAIN, 16));
    UIManager.put("OptionPane.buttonFont", new Font("Tahoma", Font.PLAIN, 14));
    JOptionPane.showMessageDialog(this, "ข้อมูลถูกต้อง ยังเป็นปีนี้อยู่ ระบบจะดำเนินการ \n Username : 1234\n Pass
}

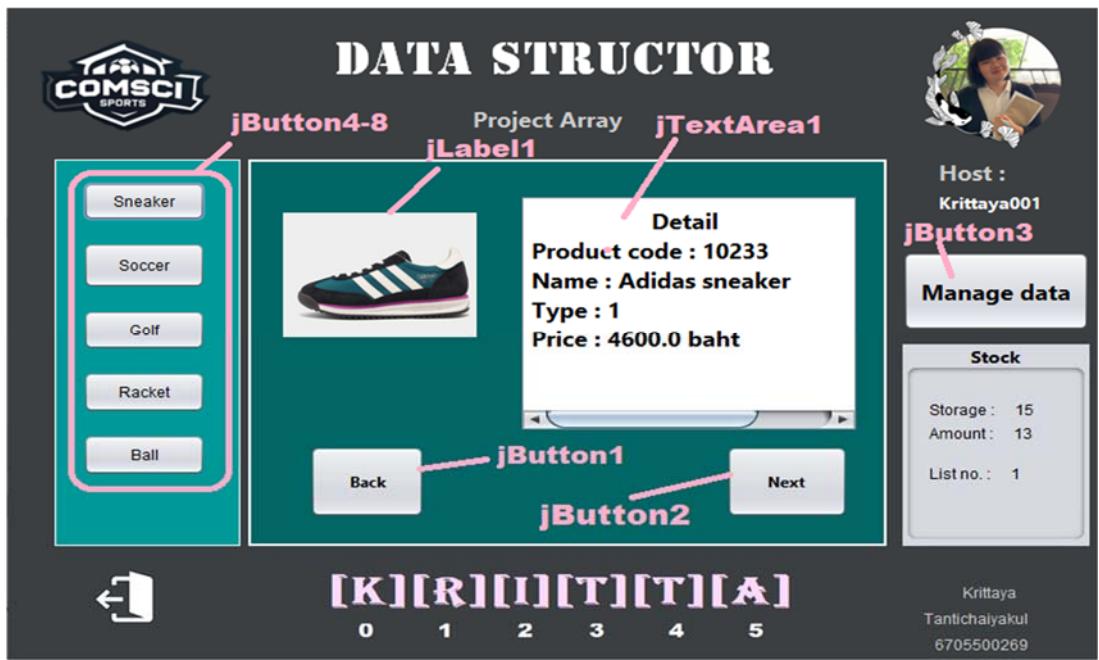
```

PCP UP
บันทึก

หลังจากที่กด Login จะเข้าฟังก์ชัน jButton1 ซึ่งจะดึงข้อมูลจาก jTextField1 และ jTextField2 แปลงมาเป็น int และเข้าคำสั่ง if ว่าทั้ง 2 texts นั้นเท่ากับ 1234 หรือไม่ ถ้าตรงก็อนุญาตให้เข้าหน้าถัดไป แต่ถ้ายังไม่ตรง ระบบก็จะ popup ขึ้นข้อความแสดงยูสเซอร์และรหัสผ่านค่ะ



3. หน้า Menu จะเป็นหน้าแสดงผลข้อมูลอาร์เรย์ที่ละรายการพร้อมรูปภาพประกอบ จะมีตัวแปรที่ชื่อ current ทำหน้าที่เก็บตำแหน่งปัจจุบันที่แสดงผล ดังนั้น jTextArea1 จะแสดงรายละเอียดข้อมูลและ jLabel1 จะแสดงรูปภาพของอาร์เรย์ตำแหน่งที่ current แบบ realtime กำหนดค่าเริ่มต้นให้ current = -1 เนื่องจากว่า หากตั้งเท่ากับ 0 เวลาผู้ใช้เข้ามาแล้วกดปุ่ม next เลย จะทำให้ค่า current+1 หรือเท่ากับตำแหน่งที่ 1 ทันที จะทำให้ข้ามข้อมูลตัวแรกไป



```

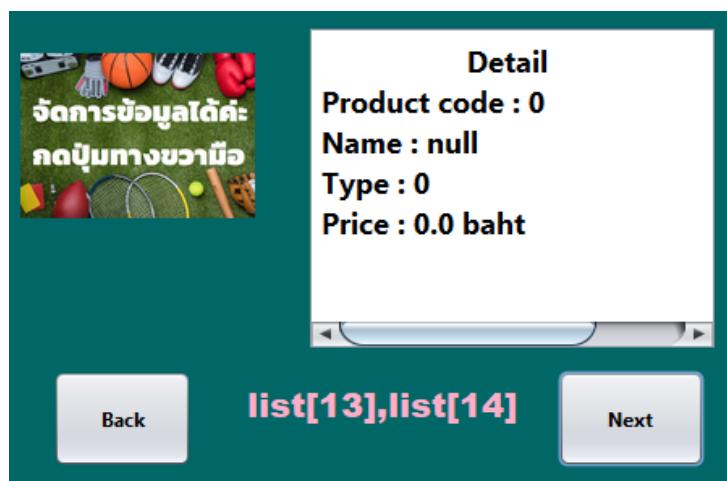
564 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
565     if(current<(DataStore.n-1)){
566         current++;
567         jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource(DataStore.list[current].pic)));
568     } else if(current<(UB-1)){
569         current++;
570         jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/sc/home.png")));
571     } else{
572         JOptionPane.showMessageDialog(this, "Overflow!");
573         System.out.println("Overflow");
574     }
575     jLabel19.setText(""+(current+1));
576     jTextArea1.setText("Detail" +
577                         "\nProduct code : "+DataStore.list[current].code+
578                         "\nName : "+DataStore.list[current].name+
579                         "\nType : "+DataStore.list[current].type+
580                         "\nPrice : "+DataStore.list[current].price+" baht");
581 }
582 }
583 }

584 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
585     if(current>DataStore.n){
586         current--;
587         jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/sc/home.png")));
588     } else if(current<LB){
589         current--;
590         jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource(DataStore.list[current].pic)));
591     } else{
592         JOptionPane.showMessageDialog(this, "Underflow!");
593         System.out.println("Underflow");
594     }
595 }
596 }
597 
```

Annotations on the code:

- Line 568:** A purple oval highlights the line `jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/sc/home.png")));` with the handwritten note "LB - UB - 1".
- Line 570:** A red oval highlights the line `jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/sc/home.png")));` with the handwritten note "Set รูป default".
- Line 576:** A blue brace groups the code from `jLabel19.setText` to the end of the block, with the handwritten note "show in textarea".
- Line 594:** A green box highlights the line `System.out.println("Underflow");` with the handwritten note "0...14".

กำหนดให้ปุ่ม next และ back แสดงข้อมูลถัดไปและก่อนหน้า โดยกำหนดสเกลไว้ที่ 0-UB โดยการใช้ if else หากอยู่ในเงื่อนไข จึงอนุญาตให้ + - current ได้ จะทำให้เวลา click next back จะไม่เกินขอบเขตที่มีอยู่ แล้วจึงแสดงผลผ่าน TextArea1 และ jLabel1 อีกครั้ง และเนื่องจากว่าภายในกล่องอารย์เรย์ยังมีข้อมูลที่ยังไม่ได้ใส่ข้อมูล ก็คือ index ที่ 13 14 เวลา click next มาถึงตำแหน่งนี้จะเกิด error เนื่องจากปกติแล้ว jLabel1 จะดึงรูปภาพจาก address ที่อยู่ในแอฟท ริบบิ้ง string pic ทุกครั้ง แต่ในตำแหน่งที่ 13 14 เป็นช่องว่างไม่มี address กด next จะขึ้นข้อความ แดง ดังนั้นเราควรหารูป default ที่ซึ่งไปรูปนั้นๆ เวลาไม่มีข้อมูลด้วย



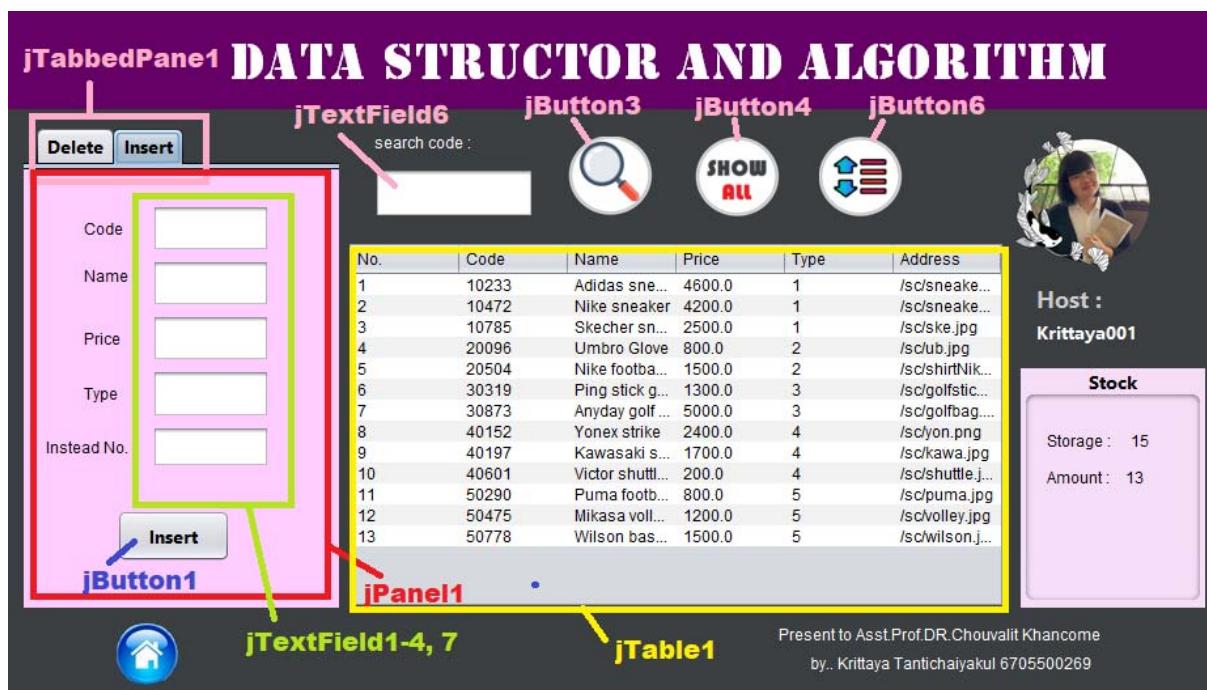
แบบปุ่ม Category ทางซ้ายมือ คือหมวดหมู่ของรายการสินค้า หากท่านกดที่ปุ่มหมวดใดระบบจะพาท่านไปยังรายการแรกของหมวดนั้นๆ คล้ายๆ สารบัญ ตัวอย่าง หมวด Soccer จะมี type คือ 2 เวลา click ระบบจะเข้าฟังก์ชัน jButton5 สร้างตัวแปร i ขึ้นมาในการท่องลูป เนื่องจากเราต้องการให้พาไปรายการแรกของหมวดนั้นๆ จะต้องเริ่มต้นที่ 0 ไปถึง n-1 หากเจอ type ที่ตรงกันตัวแรก ก็แค่กำหนด current เท่ากับ i ได้เลย พร้อมอัพเดทข้อมูลลง Area1 และรูปภาพลง Label1 ภายในฟังก์ชันนั้นๆ

```

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    int i;
    for(i=LB;i<DataStore.n;i++){
        if(DataStore.list[i].type==2){
            current=i;
            break;
        }
    }
    jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource(DataStore.list[current].pic)));
    jLabel19.setText(""+(current+1));
    jTextArea1.setText("Detail"+'+ DataStore.list[current].code +'+ DataStore.list[current].name +'+ DataStore.list[current].type +'+ DataStore.list[current].price+" baht");
}

```

4. การจัดการข้อมูล จำเป็นต้องเข้าไปยังหน้าไฟล์ Manage หรือทำการกดปุ่ม jButton3 จากหน้า Menu นั้นเอง หน้านี้จะสร้าง jTable1 ขนาด 15 ช่องตามขนาดของอาร์rays ท่านจะเห็นตารางจะแสดงข้อมูลสินค้าทั้งหมด 13 ชิ้น ซึ่งต่างจากหน้า Menu ที่จะแสดงข้อมูลทีละรายการ อีกทั้งตารางนี้มีการเชื่อมไว้ในทุกๆบุ๊มหรือทุกๆฟังก์ชันว่า เวลาที่มีการจัดการข้อมูลในรูปแบบต่างๆ ตารางนี้ก็จะอัพเดทแบบ realtime ด้วย



4.1 Insert

การแทรกข้อมูลใหม่ลงไปในกล่องอาร์เรย์ ท่านจำเป็นต้องกรอกรายละเอียดลงบน jTextField1-4 และ 7 ก่อน ค่อยกดปุ่ม Insert ระบบจะเข้าฟังก์ชัน jButton1 เริ่มต้นหนูกำหนดตัวแปรขึ้นมา 2 ตัวเพื่อรับค่าที่จำเป็นสำหรับการแทรกข้อมูลก็คือ โค้ดสินค้าและตำแหน่งที่ต้องการแทรก (jTextField1 & jTextField7) นำมาแปลงจาก string เป็น int แล้วทำการลบค่าตำแหน่งลง 1 เพื่อนำมาใช้ในอาร์เรย์ หลังจากนั้นจะเข้าลูปตรวจสอบว่า โค้ด(jTextField1) ที่รับค่ามานั้น มีเลขนำหน้าอยู่ในอาร์เรย์อยู่แล้วหรือไม่ ถ้าตรงกันจะให้เด้ง popup และแสดงข้อความว่า มีเลข โค้ดนี้อยู่แล้ว และ return เพื่อออกจากฟังก์ชัน แล้วให้ผู้ใช้ไปแก้ไขเปลี่ยนเลขอื่นแทน แต่ถ้าไม่มีเลขซ้ำ จะเข้าคำสั่ง if ถัดไปเพื่อตรวจสอบว่า บนนี้ชุดข้อมูลมากกว่าหรือเท่ากับ UB หรือไม่ ซึ่งเราเซ็ตไว้ห่างบนแล้วให้เท่ากับ 15 หากเป็นจริงแสดงว่าช่องอาร์เรย์เต็มเรียบร้อย ทำการ return ออก แต่ถ้ายังไม่เต็ม ก็จะเข้า if ที่ 2 และ 3 เพื่อตรวจสอบว่าตำแหน่งที่ผู้ใช้ส่งมา(jTextField7) นั้น น้อยกว่า 0 หรือมากกว่า UB ใหม่ตามลำดับ พุดง่ายๆคือ อยู่ภายนอกขอบเขตของอาร์เรย์ก่อนนี้ใหม่ หากเป็นจริงก็จะเด้งข้อความ Underflow และ Overflow ตามลำดับ พร้อม return ออก หากผ่านประตุ if ทั้ง 4 มาได้แล้ว นั่นหมายความว่า การแทรกข้อมูลตรงตามเงื่อนไขทุกอย่าง พร้อมทั้งพาเข้าลูปข้อนหลัง การเข้าลูปข้อนหลังคือการเริ่มจาก ตัวท้ายสุดมายังตัวที่กำหนด เนื่องจากว่าการแทรก ก็คือการเพิ่มจำนวน เราต้องถอยตัวสุดท้ายออกไป 1 ช่อง และตัวที่เหลือก็เบี่ยงตามกันมา ดังนั้นตัวที่ใกล้ช่องว่างมากที่สุดก็คือ ตัวสุดท้ายนั้นเอง ข้อมูลที่อยู่หลังจาก insertIndex จะมีค่าตำแหน่ง +1 ทุกตัวและข้อมูลใหม่ที่ได้จาก jTextField2-4 ก็จะถูกเข้ามาจัดเก็บลงในอาร์เรย์ช่องที่ insertIndex ทันที หลังจากนั้นจึงอพเดทค่า n+1 ตารางและกระดานสต็อกค่ะ

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    int insertCode = Integer.parseInt(jTextField1.getText());
    int insertIndex = Integer.parseInt(jTextField7.getText());
    insertIndex -= 1;
    for(int i=0;i<DataStore.r;i++){
        if(insertCode==DataStore.list[i].code){
            System.out.println("This code exist already!");
            JOptionPane.showMessageDialog(this, "This code exist already!");
            return;
        }
    }
    if(DataStore.r>=UB){
        System.out.println("Array full!");
        JOptionPane.showMessageDialog(this, "Array full!");
        return;
    }
    if(insertIndex<0){
        System.out.println("Underflow!");
        JOptionPane.showMessageDialog(this, "Underflow!");
        return;
    }
    if(insertIndex>=(UB-1)){
        System.out.println("Overflow!");
        JOptionPane.showMessageDialog(this, "Overflow!");
        return;
    }
    for(int j=(DataStore.r-1);j>=insertIndex;j--){
        DataStore.Products temp = new DataStore.Products();
        temp.code = DataStore.list[j].code;
        temp.name = DataStore.list[j].name;
        temp.price = DataStore.list[j].price;
        temp.type = DataStore.list[j].type;
        temp.pic = DataStore.list[j].pic;
        DataStore.list[j+1]=temp;
    }
    DataStore.list[insertIndex].code=Integer.parseInt(jTextField1.getText());
    DataStore.list[insertIndex].name=jTextField2.getText();
    DataStore.list[insertIndex].price=Integer.parseInt(jTextField3.getText());
    DataStore.list[insertIndex].type=Integer.parseInt(jTextField4.getText());
    DataStore.list[insertIndex].pic="/sc/home.png";
    DataStore.r++; - update n
    System.out.println("Status : insert complete!");
    jLabel21.setText("Amount : "+DataStore.r);
    loadTableData(); table ns: m/s
}

```

לכטת נתונים
לאבוקס

MAX
stack Ü

min index
אנו מושגחים

Ex temp = 3

temp

table ns: m/s

Insert Function

4.2 Search

สำหรับการค้นหาข้อมูล ผู้ใช้จะกรอกค่าเพียง 1 ค่า คือ โค้ดสินค้า แล้วจึงทำการสร้างตัวแปร index เพื่อคอยรับค่าเริ่มต้นกลับ ที่ส่งไปยังฟังก์ชัน searchData พร้อมกับแบบอาวิเม้นไปนั้นก็คือโค้ดที่เราเพิ่งแปลงมาจาก jTextField6 ภายในฟังก์ชันนั้น เราสร้างตัวแปรเก็บผลลัพธ์ LOC = -1 และตัวท่องอารเรย์ I=0 ปกติของอาจารย์จะเริ่มนับกลับเป็น Boolean แต่ของหนูขออนุญาตดัดแปลงนิดนึงคือ ขอส่งกลับเป็น int นะครับ 🙏 เพื่อจะส่ง index ที่ต้องการค้นหากลับไปด้วย โดยที่ทำการวนลูปหา หากโค้ดสินค้าตรงกันที่ I ได ก็กำหนดค่าให้ LOC เท่ากับ I ได้เลยและส่ง LOC กลับ แต่ถ้าไม่เจอ ก็ยังคงค่อนเข้าไปเดิมคือ ส่งค่า LOC ที่ไม่ได้มีการเปลี่ยนแปลงค่า ซึ่งก็คือ -1 กลับไปนั้นเอง

```

public int searchData(int ITEM){
    int LOC=-1; if = -1 =false
    int I=0;
    while(I<DataStore.n){
        if(ITEM==DataStore.list[I].code){
            LOC=I; =Index !=false
            System.out.println("Found at index : "+I);
            return LOC;
        }else{
            I++;
        }
    }
    System.out.println("Not found");
    return LOC;
}

```

Parameter = Code Products

Loop invariant: LOC = Index

Condition: still false -1

หลังจากกลับมาจากการคลิกปุ่ม SearchData ก็จะเข้า if ตรวจสอบสถานะว่าเจอกหรือไม่ โดยที่ค่า LOC หาก < 0 หรือจะตั้ง $= -1$ ก็ได้เหมือนกัน หากใช้จะแจ้งเตือนและรีเทิร์นออก แต่ถ้าผ่าน if นี้ นั่นหมายความว่าเราจะได้ค่า LOC ที่เป็น index กลับมา และทำการเช็คตารางให้ใช้ว่าแค่ข้อมูลในตำแหน่งที่ index เท่านั้น

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    int SearchCode= Integer.parseInt(jTextField6.getText());  
    int index=SearchData(SearchCode);  
    if(index<0){ หาก LOC หัวข้อ int ของค่า index หัวข้อ  
        JOptionPane.showMessageDialog(this, "not found!"); ตัวไม่พบข้อหา  
        return; -1 ใน if นี้  
    } Reset Table show ค่า ,  
โดยหาค่ามาจากหัวข้อ  
ที่สุ่ม  
    DefaultTableModel model = (DefaultTableModel) jTable1.getModel();  
    model.setRowCount(0);  
    model.addRow(new Object[]{index+1,  
        DataStore.list[index].code,  
        DataStore.list[index].name,  
        DataStore.list[index].price,  
        DataStore.list[index].type,  
        DataStore.list[index].pic  
    });
```

4.3 Delete

สำหรับการลบข้อมูล จะใช้ข้อมูลจากผู้ใช้คือ โค้ดสินค้า ใช้วิธีเดียวกันกับการ Search เลย เพราะเราจำเป็นต้องหา ก่อนว่า โค้ดสินค้าที่ต้องการจะลบ ตรงกับชุดข้อมูลในอารเรย์ในตำแหน่งที่เท่าไร โดยการ โอนค่า โค้ดสินค้าลงบนฟังก์ชัน searchData

หลังจากกลับมาจากการคลิก โค้ด searchData ก็จะเข้า if ตรวจสอบสถานะว่า เชื่อมต่อหรือไม่ เมื่อเชื่อมต่อแล้ว ก็ทำการสร้างลูป ให้เริ่มต้นตั้งแต่ index ที่ได้รับมา เราจะเอาช่องนั้นให้มีค่าเท่ากับอารเรย์ช่องถัดไป ซึ่งข้อมูลเดิมตำแหน่งที่ index จะหายไปแล้ว ทำงานไปเรื่อยๆ จนครบลูป ในตอนนี้ เราจะได้ข้อมูลอารเรย์ที่มีจำนวนเท่าเดิม แต่จะมีข้อมูลที่เหมือนกัน 2 ตัว อยู่ท้ายແຕງ เราจะต้องลบข้อมูลตัวสุดท้ายออก โดยการกำหนดให้ตัวสุดท้ายเท่ากับ null พิริ่อมทั้งอพเดท n-1 ตารางและกระดาษสต็อกด้วย

```
587 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
588     int delCode = Integer.parseInt(jTextField5.getText());  
589     int index=SearchData(delCode);  
590     if(index<0){  
591         System.out.println("Status : not found");  
592         jLabel11.setText("Status : not found..");  
593         return;  
594     }  
595     jLabel12.setText("Code : "+DataStore.list[index].code);  
596     jLabel13.setText("Name : "+DataStore.list[index].name);  
597     jLabel14.setText("Price : "+DataStore.list[index].price);  
598     jLabel15.setText("Type : "+DataStore.list[index].type);  
599     for(int i=index;i<(DataStore.n-1);i++){  
600         DataStore.list[i]=DataStore.list[i+1];  
601     }  
602     DataStore.list[DataStore.n-1]=null;  
603     DataStore.n--;  
604     System.out.println("Status : delete complete!");  
605     jLabel11.setText("Status : delete complete!");  
606     jLabel21.setText("Amount : "+DataStore.n);  
607     loadTableData();  
608 }
```

4.4 Sorting

หากผู้ใช้มีการแทรกข้อมูลเข้าไปในอารเรย์ โดยข้อมูลไม่ได้เรียงกันถูกต้อง ท่านสามารถกด sort ได้ จะพาท่านไปยังฟังก์ชั่น JButton6 อัลกอริทึมของอาจารย์จะเป็นการเรียงลำดับแบบ bubble sort โดยที่รอบแรกจะได้ตัวมากสุดอยู่ในลำดับสุดท้ายและไม่มาเรื่อยๆจนถึงค่าน้อยที่สุด ตัวแรก เราจะสร้าง 2 ลูป เพราะเราจะต้องໄล่เทียบ 2 ตัวแล้ว ตัวแรกคือ K หรือตัวที่ 2 ในลูปแรก เริ่มต้นที่ 0 และ PTR ตัวเทียบ ไว้ในลูปที่สอง ค่อยໄล่สมาชิกทุกตัวมาเทียบกับตัวที่ K ในทุกรอบ ในแต่ละรอบจะกำหนดให้ $PTR = 0$ เป็นอ เพื่อให้เริ่มวิ่งตั้งแต่ตัวแรกทุกครั้ง จนถึงตัวที่ $n-K$ สาเหตุที่ต้องลบ K ออกในลูปที่ 2 เนื่องจาก ในแต่ละรอบ เราจะได้ตัวที่มากที่สุดคุณท้ายแล้วเรียบร้อยแล้ว ไม่ว่าจะเป็นรอบที่ 2 หรือ 3 หรือรอบที่ใดๆก็ตาม เราจะได้ตัวที่มากที่สุดจำนวน K ตัว(รอบ)คุณท้ายอาจารย์ของเรา ดังนั้น เราไม่จำเป็นต้องวนเทียบทุกอีลิเม้นในรอบต่อๆมา ก็ได้ ซึ่งตรงนี้เป็นหัวใจสำคัญในการเขียนอัลกอริทึมที่ดี

```
private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
    int PTR; // จัดการข้อมูล
    for(int K=0;K<(DataStore.n-1);K++){
        PTR=0; → Reset ตัวแปรPTR ให้เป็น 0
        while(PTR<(DataStore.n-1)-K){
            if(DataStore.list[PTR].code>DataStore.list[PTR+1].code){
                DataStore.Products temp = new DataStore.Products();
                temp=DataStore.list[PTR];
                DataStore.list[PTR]=DataStore.list[PTR+1];
                DataStore.list[PTR+1]=temp;
            }
            PTR++; // เพิ่มตัวแปร PTR 1 ขึ้นไป
        }
    }
    System.out.println("Status : data sorted");
    loadTableData();
}
```

กราฟที่แสดงการทำงานของขั้นตอนนี้

↑ temp(9)

↑ temp(9)

ขอบคุณค่ะ