Project Array Part II

COS2103

Data structure and Algorithm

1 / 2568

Present to

Asst.Prof.DR.Chouvalit Khancome

By Krittaya Tantichaiyakul

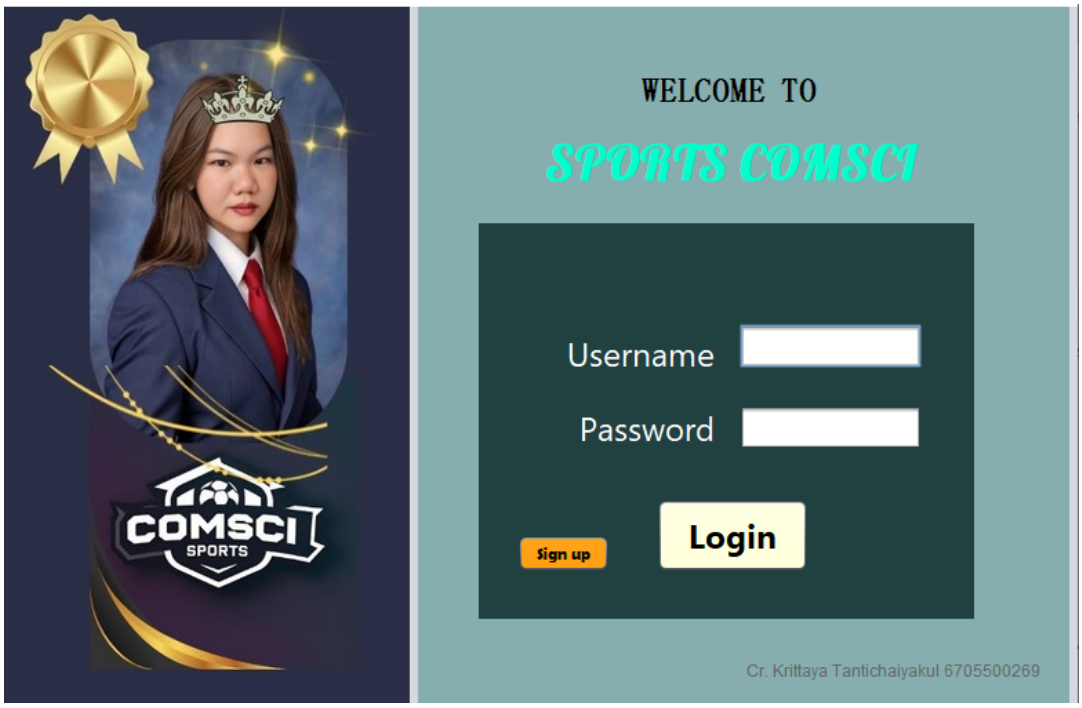
6705500269

หัวข้อ

อุปกรณ์กีฬา

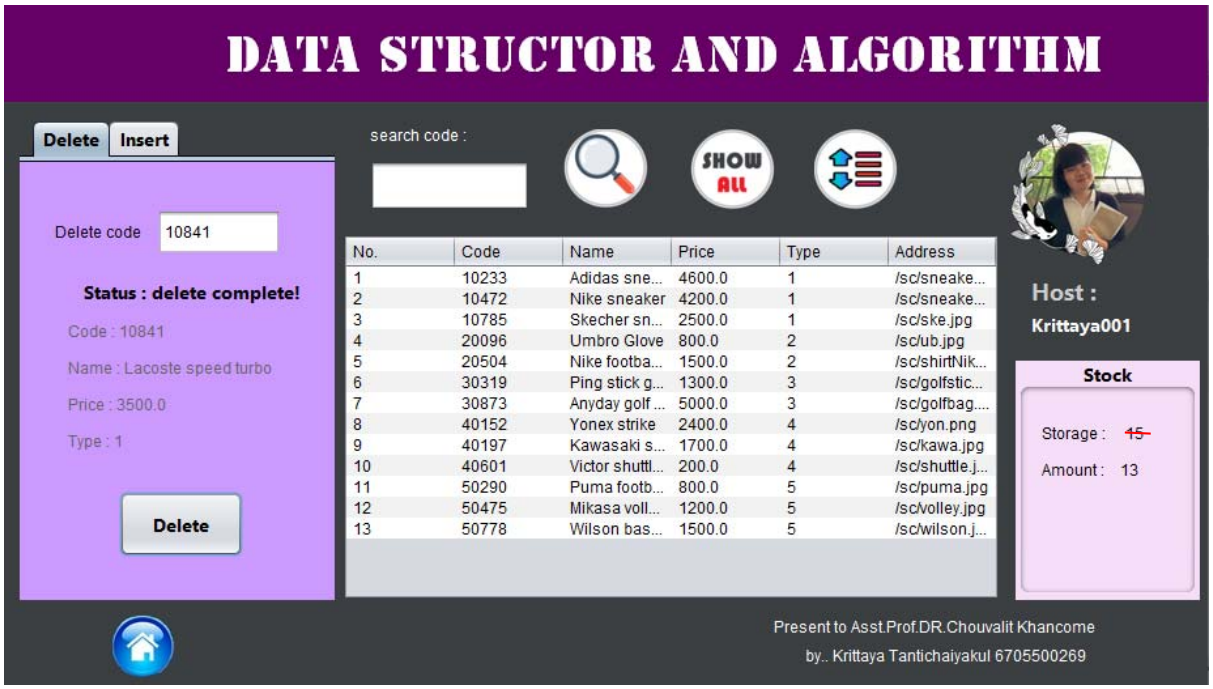
Program Interface Preview

1. Login page - Login



2. Category of array - Menu

3. list of products – Manage



4. ไฟล์กลางจัดเก็บอาร์เรย์ – 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      }
```

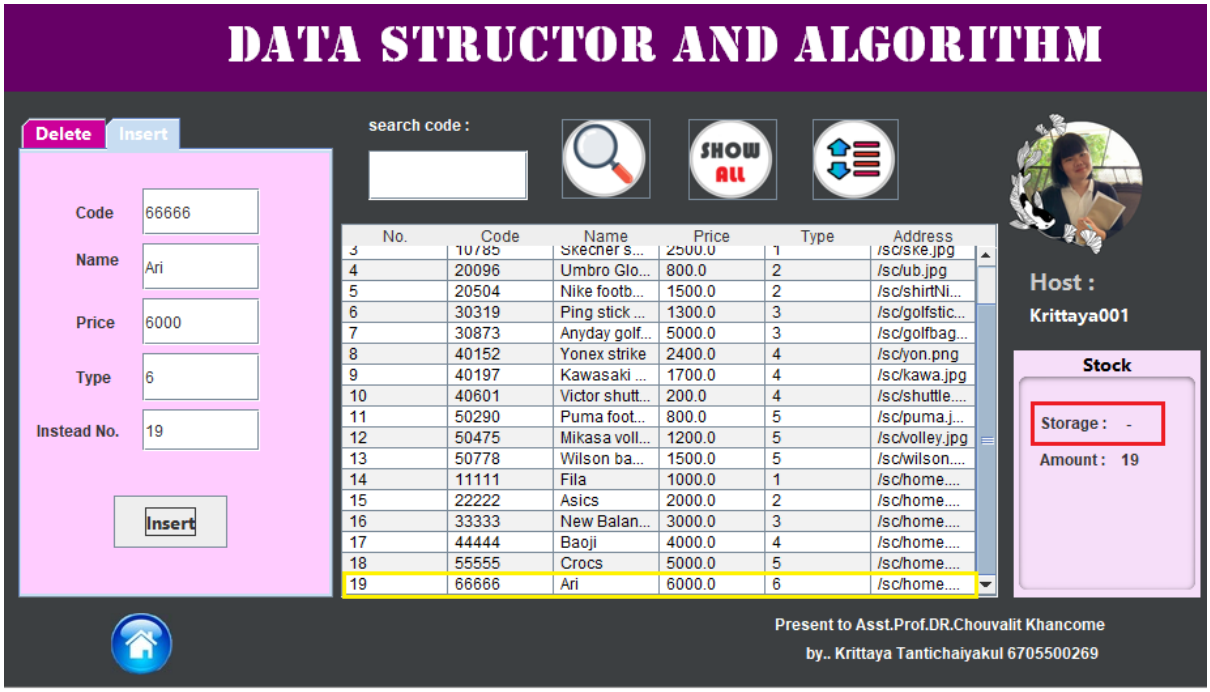
```
35      public static Products[] list = new Products[15];
36
37      static {
39          for (int i = 0; i < list.length; i++) {
40              list[i] = new Products();
41          }
42          // กำหนดข้อมูลตั้งแต่แรก
          list[0].setAll(10233,"Adidas sneaker",1,4600,"/sc/sneakerAdi.jpg");
          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");
56      }
57  }
58
```

ความแตกต่างระหว่าง

Part I และ Part II

1. สำหรับ Part I จะมีการใช้งานอาเรย์แบบปกติ array[] และมีการใช้งานคู่กับอัลกอริทึมค่อนข้างมาก ส่วน Part II จะมีการนำเข้า Java.util.ArrayList มาใช้งาน ซึ่งแพ็กเกจนี้จะอำนวยความสะดวกในการใช้งานมากขึ้น มีการใช้งานอัลกอริทึมที่น้อยลง สั้นลง รวดเร็วขึ้น ทำให้ผู้ใช้เขียนโค้ดได้ง่าย และโปรแกรมทำงานได้ดียิ่งขึ้น

2. การสร้างอาเรย์แบบปกติ จะบังคับให้ผู้ใช้เก็บข้อมูลได้ตามขนาดของมัน หากเต็มแล้วต้องการจะเพิ่มข้อมูลอีก จำเป็นต้องลบหรือสร้างอาเรย์ขึ้นมาใหม่ที่มีขนาดใหญ่กว่าเดิม ซึ่งเป็นปัญหาสำหรับใครหลายคน แพ็กเกจ Java.util.ArrayList นี้เองจึงถูกสร้างมาเพื่อแก้ปัญหาดังกล่าว จะทำให้อาเรย์ของเรามีความยืดหยุ่นมากขึ้น กล่าวคือ สามารถเพิ่มข้อมูลได้เรื่อยๆ นั่นเอง



3. เนื่องจากอาเรย์ชนิด arraylist สามารถยืดหยุ่นได้ และมีคำสั่ง size() เพื่อตรวจสอบขนาดของอาเรย์ลิสได้ เราจึงไม่จำเป็นต้องใช้ตัวแปร n อีกต่อไป เพียงแต่คำสั่งอาจพิมพ์ยาวนิดนึง เนื่องจากเราประกาศ arraylist ไว้ในไฟล์กลาง DataStore จึงต้องมีการเข้าถึง ก็คือ DataStore.list.size() แต่แลกกับการที่เวลาเรา insert delete หรืออัพเดทกระดานสต็อก เราไม่จำเป็นต้องกังวลเรื่องความถูกต้องของ n++ หรือ n— อีกแล้ว

```
int insertCode=Integer.parseInt(jTextField1.getText());
int insertIndex=Integer.parseInt(jTextField7.getText());
insertIndex-=1;
for(int i=0;i<DataStore.list.size();i++){
    if(insertCode==DataStore.list.get(i).code){
        System.out.println("This code exist already!");
        JOptionPane.showMessageDialog(this, "This code exist alread
        return;
    }
}
if(insertIndex<0){
```

4. ปกติเราจะสร้างตัวแปร UB ซึ่งเป็นขนาดของอาเรย์ แต่การนำแพ็กเกจมาใช้ อาเรย์ลิสของเราจะไม่มีวันเต็ม ดังนั้นตัวแปร UB จึงไม่จำเป็นสำหรับการใช้อัลกอริทึมอีกต่อไป เวลากำหนดสเกล เราก็สามารถตั้งว่า 0-DataStore.list.size() หรือพูดง่ายๆก็คือ ตั้งแต่ 0 ถึง จำนวนสมาชิกทั้งหมดได้เลย

5. ในหน้า Menu คำสั่งปุ่ม next back ปกติเราจะต้องสร้างเงื่อนไขแบบเลือกอย่างใดอย่างหนึ่ง คือ ค่า current อยู่ภายในจำนวนสินค้าที่มีอยู่ไหม เพื่อโชว์ข้อมูลภายในสินค้าที่มีอยู่ ต่อมาค่า current อยู่ภายในขนาดของอาเรย์ไหม เพื่อโชว์ข้อมูลช่องว่างที่เหลือด้วย แต่สำหรับการนำแพ็กเกจมาใช้ เราจะไม่มีเพดาน ดังนั้นไม่จำเป็นต้องโชว์ตำแหน่งข้อมูลที่ว่างแล้ว สามารถทำให้ current ไม่เกินจำนวนสินค้าหรือ .size() ได้เลย

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent
    if(current<DataStore.list.size()-1){
        current++;
        jLabel1.setIcon(new javax.swing.ImageIcon(
getClass().getResource(DataStore.list.get(current).pic)
));

    }else{
        JOptionPane.showMessageDialog(this, "Overflow!");
        System.out.println("Overflow");
    }
```

6. สำหรับการ insert ถือเป็นหัวข้อที่เห็นความแตกต่างระหว่างอาเรย์แบบปกติกับอาเรย์ลิสมากที่สุด ปกติการที่เราจะแทรกข้อมูลลงอาเรย์แบบปกติ เราจะใช้อัลกอเยอะมาก เริ่มตั้งแต่ตรวจสอบว่าโค้ดซ้ำไหม อาเรย์เต็มหรือยังและตำแหน่งที่ต้องการแทรกอยู่ภายในอาเรย์ไหม ซึ่งสำหรับ arraylist เราไม่มีเพดาน จึงไม่จำเป็นต้องใช้ if ตรวจสอบว่าอาเรย์เต็มหรือยัง แทรกข้อมูลเข้าไปได้เลย ต่อมา if ตรวจสอบว่าตำแหน่งที่ต้องการแทรกเกินอาเรย์ลิสไหม จากเดิม insertIndex>=(UB-1) เราสามารถแทรกลงไปตรงไปตำแหน่งไหนของอาเรย์ก็ได้ รวมถึงช่องว่างด้วย แต่ครั้งนี้เราจะไม่มีช่องว่างแล้ว จึงต้องเปลี่ยน UB เป็น size() และที่สำคัญ ผู้ใช้ยังมีโอกาสที่จะแทรกข้อมูลเข้าไปเป็นตัวสุดท้ายได้ด้วย เราจะต้องขยายสเกล +1 โดยเปลี่ยนจาก >= เป็น > หรือให้ size()+1 ก็ได้เหมือนกัน

```
530         int insertCode=Integer.parseInt(jTextField1.getText());
531         int insertIndex=Integer.parseInt(jTextField7.getText());
532         insertIndex-=1;
533 1       for(int i=0;i<DataStore.list.size();i++){
534             if(insertCode==DataStore.list.get(i).code){
535                 System.out.println("This code exist already!");
536                 JOptionPane.showMessageDialog(this, "This code exist already!");
537                 return;
538             }
539         }
540 2       if(insertIndex<0){
541             System.out.println("Underflow!");
542             JOptionPane.showMessageDialog(this, "Underflow!");
543             return;
544         }
545
546 3       if(insertIndex>(DataStore.list.size()) ){
547             System.out.println("Overflow!");
548             JOptionPane.showMessageDialog(this, "Overflow!");
549             return;
550         }
551         DataStore.Products obj2 = new DataStore.Products();
552         obj2.code=Integer.parseInt(jTextField1.getText());
553         obj2.name=jTextField2.getText();
554         obj2.price=Integer.parseInt(jTextField3.getText());
555         obj2.type=Integer.parseInt(jTextField4.getText());
556         obj2.pic="/sc/home.png";
557         DataStore.list.add(insertIndex,obj2);
558         System.out.println("Status : insert complete!");
559         jLabel21.setText("Amount :    "+DataStore.list.size());
560         loadTableData();
561     }
```

} Temp

การใช้ฟังก์ชั่น add เป็นคำสั่งที่จะเพิ่มข้อมูลลงอาเรย์ลิสอัตโนมัติ โดยที่เราไม่จำเป็นต้องขยับอีลิเม้นตัวข้างหลังออกทุกตัวอีกแล้ว ซึ่งตรงนี้ถือว่าเป็นการลดขั้นตอนที่ดีมาก ตอบโจทย์วิชานี้ก็คือ การออกแบบอัลกอลิทึมอย่างมีประสิทธิภาพ ทำให้จำนวน Big O ลดลง แต่คำสั่งนี้มีเงื่อนไขว่า ต้องแนบอากิวเม้นตัวที่ 2 เป็นอ็อบเจ็กที่เป็นข้อมูลใหม่ลงไปด้วย เราจะต้องสร้างอ็อบเจ็กใหม่สำหรับเซ็ตค่าที่ได้รับจาก TextField ไว้ก่อน หลังจากที่อ็อบเจ็กตัวนั้นมีหน้าตาของมันเรียบร้อยแล้ว แล้วจึงสามารถใส่ลงไปบนฟังก์ชั่น add(int, object) เพื่อเพิ่มข้อมูลได้แล้ว

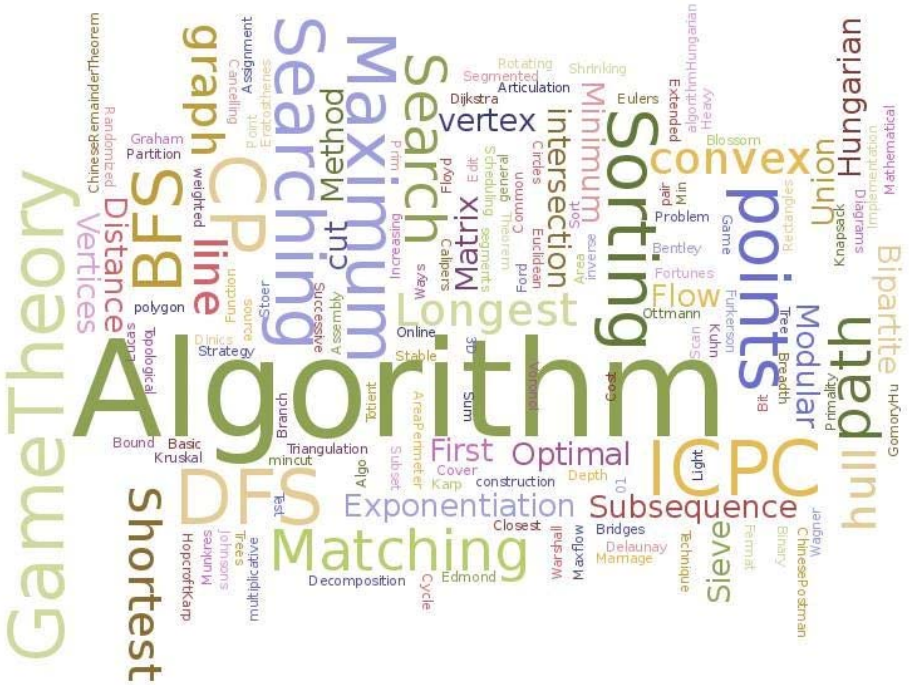
2. สำหรับการ delete จะมีคำสั่งที่อำนวยความสะดวกก็คือ remove() มาพร้อมใช้งาน เราไม่จำเป็นต้องใช้อัลกอ for แล้วขยับตัวหลังมาแทนตัวหน้าอีกเช่นกัน ใช้คำสั่ง remove() คือจบ แต่คำสั่งนี้มีเงื่อนไขคือ ต้องใส่อากิวเม้นเป็นค่า index เพื่อลบตำแหน่งนั้นด้วย ดังนั้น เราจึงต้อง search หาตำแหน่งดังกล่าวก่อน แล้วจึงนำค่านั้นใส่ลงบนอากิวเม้นได้เลย เช่น remove(index)

```
569  private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
570      int delCode = Integer.parseInt(jTextField5.getText());
571      int index=SearchData(delCode);
572      if(index<0){
573              System.out.println("Status : not found");
574              jLabel11.setText("Status : not found..");
575              return;
576      }
577      jLabel12.setText("Code : "+DataStore.list.get(index).code);
578      jLabel13.setText("Name : "+DataStore.list.get(index).name);
579      jLabel14.setText("Price : "+DataStore.list.get(index).price);
580      jLabel15.setText("Type : "+DataStore.list.get(index).type);
581
582      DataStore.list.remove(index);
583      System.out.println("Status : delete complete!");
584      jLabel11.setText("Status : delete complete!");
585      jLabel21.setText("Amount :    "+DataStore.list.size());
586      loadTableData();
587
588  }
```

3. สำหรับการค้นหา search นั้นยังไม่มีคำสั่งที่ค้นหาตำแหน่งโดยตรง จะมีที่ใช้แทนกันได้บ้างก็คือ contains() แต่คำสั่งนี้จะให้ค่า Boolean ว่าเจอหรือไม่เจอแค่นั้น แต่โปรแกรมของเราจำเป็นต้องใช้คำสั่งที่ค้นหาตำแหน่งด้วย เพื่อนำมาประยุกต์ใช้กับการลบและแทรกข้อมูล หนูจึงยังคงต้องใช้อัลกอเดิมในการค้นหาเดต้าอยู่ค่ะ

รวมถึงการ sort ด้วยเช่นกัน แพ็คเกจอาเรย์ลิสยังไม่มีคำสั่งในการจัดเรียงข้อมูลค่ะ

ดังนั้นแล้ว จุดประสงค์ของอาจารย์ สำหรับพาร์ทนี้ก็คือ ให้นักศึกษาเรียนรู้โครงสร้างข้อมูลและอัลกอริทึมหลายๆรูปแบบ เพื่อทำให้เข้าใจโครงสร้างข้อมุลของอาเรย์ที่แท้จริง และเอกลักษณ์ของอัลกอแต่ละแพ็คเกจว่าทำงานอย่างไร โด่ดเด่นในด้านไหน แบบไหนทำให้ทำงานได้รวดเร็วหรือช่วยลดช่องโหว่ที่อาจทำให้เกิดความผิดพลาดได้มากกว่า ส่งผลให้นักศึกษาสามารถเขียนโปรแกรมได้หลากหลายรูปแบบมากขึ้น บางอันใช้แทนกันได้ บางอันใช้แทนกันไม่ได้ ทำให้เราสามารถวิเคราะห์ได้ว่า โจทย์ปัญหาต่างๆ ควรเลือกอัลกอแบบใดที่เหมาะสมที่สุดมาใช้งาน



ขอบคุณค่ะ