

แบบฝึกปฏิบัติครั้งที่ 5

Fundamental Data Structures (Single Linked List)

คำสั่ง จงสร้างคลาส SLinkStack เพื่อใช้ในการจัดการ stack บนโครงสร้าง Single Linked List

คลาส Node จะเป็นต้นแบบของแต่ละโหนด โดยในตัวอย่างจะเก็บกำหนดรูปแบบของโหนดให้เก็บข้อมูลประเภท String หากต้องการเปลี่ยนโครงสร้างของโหนดให้เก็บข้อมูลประเภทอื่นก็สามารถกำหนดได้ในส่วนของชนิดข้อมูลของตัวแปร item และเมธอดต่าง ๆ ที่เกี่ยวข้อง

```
public class Node {  
    private String item;  
    private Node link;  
    // constructors  
    // head node  
    public Node() {  
        link = null;  
    }  
    // last node  
    public Node(String newItem) {  
        item = newItem;  
        link = null;  
    }  
    public Node(String newItem, Node nextNode) {  
        item = newItem;  
        link = nextNode;  
    }  
    // begin method  
    public void setItem(String newItem) {  
        item = newItem;  
    }  
}
```

```
public String getItem() {  
    return item;  
}  
  
public void setLink(Node nextNode) {  
    link = nextNode;  
}  
  
public Node getLink() {  
    return link;  
}  
}
```

คลาส SingleLinkedList จะประกอบด้วยเมธอดต่าง ๆ ในการจัดการลิงค์ลิสต์ทางเดียว โดยจะมีการเรียกใช้เมธอดในคลาส Node เพื่อช่วยจัดการข้อมูลในระดับโหนด ดังนั้นในการเก็บข้อมูลในลิงค์ลิสต์จะเป็นข้อมูลประเภท String เช่นเดียวกับคลาส Node

```
public class SingleLinkedList {  
    Node newNode;  
    Node head;  
    Node curr;  
    Node prev;  
    // การสร้างลิงค์ลิสต์ทางเดียว  
    public void add (String newItem) {  
        if (head == null) {  
            newNode = new Node(newItem);  
        } else {  
            newNode = new Node(newItem, head);  
        }  
        head = newNode;  
    }  
}
```

```
// การค้นหาข้อมูลบนลิงค์ลิสต์ทางเดียว
public boolean searchData(String item) {
    curr = new Node();
    prev = new Node();
    curr = head;
    prev = null;
    while (curr != null) {
        if (curr.getItem().equalsIgnoreCase(item)) {
            return true;
        } else {
            prev = curr;
            curr = curr.getLink();
        }
    }
    return false;
}

// การลบโหนดในลิงค์ลิสต์ทางเดียว
public void deleteNode(String item) {
    // ทำการเรียกใช้เมธอด searchData เพื่อค้นหาตำแหน่งที่ต้องการลบ
    if (searchData(item)) {
        //การลบโหนดที่ตำแหน่งแรกของลิงค์ลิสต์
        if (head == curr) {
            head = curr.getLink();
        } else if (curr.getLink() == null) {
            // การลบโหนดที่ตำแหน่งสุดท้ายของลิงค์ลิสต์
            prev.setLink(null);
        } else {
            // การลบโหนดที่อยู่ระหว่างโหนดในลิงค์ลิสต์
            prev.setLink(curr.getLink());
        }
    }
}
```

```
// การแทรกโหนดเข้าลิงค์ลิสต์ทางเดียว
public void insertNode(String iteminsert, String newItem) {

    // ทำการสร้างโหนดใหม่ที่จะทำการแทรก
    newNode = new Node(newItem);

    // เรียกเมธอด searchData เพื่อค้นหาตำแหน่งที่ต้องการแทรกโหนด
    if (searchData(iteminsert)) {

        // ทำการแทรกโหนดที่ตำแหน่งแรก
        if (head == curr) {
            newNode.setLink(curr);
            head = newNode;
        } else {
            // ทำการแทรกโหนดเข้าระหว่างโหนดในลิงค์ลิสต์
            newNode.setLink(curr);
            prev.setLink(newNode);
        }
    }
}

// การแทรกโหนดเข้าต่อท้ายลิงค์ลิสต์ทางเดียว
public void insertLastNode(String newItem) {
    // ทำการสร้างโหนดใหม่ที่จะทำการแทรก
    newNode = new Node(newItem);
    // ทำการค้นหาตำแหน่งสุดท้ายของลิงค์ลิสต์
    curr = new Node();
    prev = new Node();
    curr = head;
    prev = null;
```

```
        while (curr != null) {
            prev = curr;
            curr = curr.getLink();
        }

        // ทำการอัปเดตค่า Link เพื่อปรับการเชื่อมโยง
        prev.setLink(newNode);
    }

    // การแสดงข้อมูลในลิงค์ลิสต์ทางเดียว
    public void showdata() {

        // ทำการเดินทางไปบนลิงค์ลิสต์ทีละโหนด จนถึงโหนดสุดท้าย
        curr = new Node();
        prev = new Node();
        curr = head;
        prev = null;
        while (curr != null) {

            // เมื่อถึงแต่ละโหนดจะทำการอ่านค่าข้อมูลที่เก็บในโหนดมาแสดง
            System.out.print(curr.getItem() + "\t");
            prev = curr;
            curr = curr.getLink();
        }
    }
}
```

โจทย์

1. ทำการสร้างคลาส SLinkStack ซึ่งภายในประกอบด้วยเมธอดในการจัดการ Stack บนโครงสร้างลิงค์ลิสต์ทางเดียว
 - 1.1 เมธอด push
 - 1.2 เมธอด pop
 - 1.3 เมธอด top : จะให้ค่าข้อมูลที่ตำแหน่ง top
 - 1.4 เมธอด empty : ตรวจสอบว่า stack มีข้อมูลหรือไม่
 - 1.5 เมธอด show : แสดงข้อมูลทั้งหมดที่มีใน stack ไล่จากตำแหน่ง top ถึงตำแหน่งแรก
2. ทำการสร้างคลาส testStack ภายในคลาสเมื่อรันโปรแกรมจะปรากฏเมนูให้เลือกดังนี้
Main Menu
 1. create stack
 2. push
 3. pop
 4. top
 5. show
 6. exitPlease chose your choice :

การทำงานจะวนไปเรื่อยๆ จนกระทั่งผู้ใช้งานทำการเลือกเมนู exit จึงจะจบการทำงาน ซึ่งจะแสดงข้อมูลที่มี stack ตอนนั้นก่อนหยุดการทำงาน

วิธีการส่งงาน

1. ให้นักศึกษาทำการ export project โดยตั้งชื่อเป็น รหัสนักศึกษา
2. ทำการอัปโหลดไฟล์ส่งทาง google classroom โดยนักศึกษาต้องส่งให้ถูก section ไม่เช่นนั้น จะไม่ได้คะแนน
3. เขียนคอมเม้น แจ้งชื่อและรหัสนักศึกษา