

คลาส **LinkedList** ที่เรียนมาเป็นโครงสร้างข้อมูลอีกแบบที่ใช้เก็บชุดของข้อมูลที่ไม่สนใจลำดับ แบบฝึกหัดนี้เป็นการฝึก

- เปรียบเทียบคลาส **ArrayCollection** กับ **LinkedList**,
- สร้าง method เพิ่มในคลาส **LinkedList** และ
- นำคลาส **LinkedList** ไปสร้างโครงสร้างข้อมูลใหม่

แบบฝึกหัดบางส่วนมาจาก “โครงสร้างข้อมูล : ฉบับวจาจาจา” (<https://www.cp.eng.chula.ac.th/books/ds-vjiv/>) โดย รศ.ดร.สมชาย ประสิทธิ์จูตระกูล

เปรียบเทียบคลาส **ArrayCollection** กับ **LinkedList**

เขียนโปรแกรมจับเวลาเพื่อเปรียบเทียบ running time สำหรับ operation ต่างๆ สำหรับคลาส **ArrayCollection** กับคลาส **LinkedList** โดยหาเวลาทำงานเฉลี่ย (average running time) จากการทำงานกับค่าที่สุ่มขึ้นมา 10000 ครั้ง สำหรับ Collection ที่มีขนาดต่างกัน (เช่น 10000, 100000) เพื่อนำมาวาดกราฟเปรียบเทียบ running time ของ แต่ละ operation บน collection ขนาดต่างๆ กัน operation ที่จะเปรียบเทียบ คือ method **add**, method **remove**, method **contains**

(อาจใช้ `long t = System.nanoTime();` เพื่ออ่านเวลาของ CPU)

เขียน methods สำหรับคลาส **LinkedList**

เขียน method ต่อไปนี้เพิ่มสำหรับคลาส **LinkedList** ที่เรียนมา (ส่วนใหญ่มาจาก [1])

- Constructor **LinkedList(Object[] c)** ที่สร้าง **LinkedList** ใหม่ที่เก็บค่าชุดเดียวกับ **c**
- Constructor **LinkedList(Collection c)** ที่สร้าง **LinkedList** ใหม่ที่เก็บค่าชุดเดียวกับ **c**
- **void clear()** ที่ลบข้อมูลทั้งหมดใน **LinkedList** ที่เป็น implicit parameter
- **int frequency(Object e)** ที่นับจำนวน object **e** ใน **LinkedList** ที่เป็น implicit parameter
- **Object mode()** ที่ return ฐานนิยม (mode) ของค่าใน **LinkedList**
- **void removeAll(Object e)** ที่ลบ **e** ทุกตัวที่อยู่ใน **LinkedList** ที่เป็น implicit parameter
- **boolean containsDup()** ที่ตรวจสอบว่ามีข้อมูลที่ซ้ำกันใน **LinkedList** ที่เป็น implicit parameter หรือไม่
- **void removeDup()** ที่ลบข้อมูลที่ซ้ำกันใน **LinkedList** ที่เป็น implicit parameter ให้เหลือตัวเดียว
- **boolean containsAll(LinkedList c)** ที่ตรวจสอบว่าข้อมูลทุกตัวใน **c** อยู่ใน **LinkedList** ที่เป็น implicit parameter
- **boolean equals(Collection c)** ที่ตรวจสอบว่าข้อมูลทุกตัวใน **c** อยู่ใน **LinkedList** ที่เป็น implicit parameter และข้อมูลทุกตัวใน **LinkedList** ที่เป็น implicit parameter อยู่ใน **c**
- **void addArray(LinkedList c)** ที่นำ object ทุกตัวใน **c** ใส่เพิ่มใน **LinkedList** ที่เป็น implicit parameter
- **void removeArray(LinkedList c)** ที่ลบ object ทุกตัวใน **c** ออกจาก **LinkedList** ที่เป็น implicit parameter

และสร้าง test class ที่ใช้ **LinkedList** เพื่อทดสอบ method ที่สร้างมา

สร้างคลาส **LinkedListwithDup**

เมื่อมีข้อมูลที่มีค่าซ้ำกันใน Collection เป็นจำนวนมาก จะเปลืองเนื้อที่เก็บข้อมูลซ้ำๆ กัน ให้สร้างคลาส **LinkedListwithDup** ที่เก็บข้อมูลที่ซ้ำกันที่โหนดเดียวแต่จะเก็บจำนวนครั้งที่ข้อมูลนั้นปรากฏใน Collection ให้คลาส **LinkedListwithDup** implement **Collection** ที่กำหนดไว้ด้วย และสร้าง test class ที่ใช้ **LinkedListwithDup** เพื่อทดสอบคลาสที่สร้างมา

สร้างคลาส LinkedSet

กำหนด interface **Set** ต่อไปนี้

```
public interface Set extends Collection {  
    public void add(Object element); // add a new element without duplication  
    public Set union(Set s);         // return the union of implicit parameter and s  
    public Set intersect(Set s);     // return the intersection of implicit parameter and s  
    public Set difference(Set s);    // return the set difference of implicit parameter and s  
}
```

สร้างคลาส **LinkedSet** ที่ implement **Set** และสร้าง test class ที่ใช้ **LinkedSet** เพื่อทดสอบคลาสที่สร้างมา