

พรรคมอนสเตอร์

ข้อนี้จะให้เขียนสร้างพรรคมอนสเตอร์ (Monster Party) เพื่อหาหัวหน้าพรรคจากมอนสเตอร์ที่มีค่า hit point สูงสุด
ให้เขียนสร้างไฟล์ตามลำดับต่อไปนี้

1. เหล่ามอนสเตอร์ที่เป็นสมาชิกพรรค จะมีชื่อมอนสเตอร์และค่า hit point ให้เขียนสร้างคลาส Monster

(//elab-source:Monster.java) เพื่อเป็นต้นแบบของ object มอนสเตอร์ ซึ่งมี constructor ที่
กำหนดค่าชื่อ (String) และค่า hit point (int) ตามลำดับ

คลาส Monster จะมี method String toString() ที่คืนค่าในรูปแบบของ ชื่อ:ค่า hit point เช่น ถ้ามอนสเตอร์
ชื่อ Poring มีค่า hit point เท่ากับ 10 แล้ว method toString() จะคืนค่า Poring:10

คลาส Monster จะมี method int compareTo(Object o) ที่ implements จาก interface Comparable
เพื่อใช้เปรียบเทียบค่า hit point ของเหล่ามอนสเตอร์ด้วยกันเอง ให้เขียนเปรียบเทียบค่า hit point เพื่อให้สามารถเรียงลำดับ
มอนสเตอร์ที่มีค่า hit point มาก ไปหามอนสเตอร์ที่มีค่า hit point น้อย

2. ให้เขียนสร้าง interface MonsterParty ที่มีโค้ดดังนี้

```
//elab-source:MonsterParty.java
import java.util.ArrayList;
public interface MonsterParty {
    ArrayList<Monster> createMonsters();
}
```

interface นี้ใช้เพื่อสร้างสมาชิกพรรคของมอนสเตอร์ ผ่าน method createMonsters()

// มีต่อหน้าต่อไป

3. สร้างคลาส MyParty เพื่อสร้างพรรคของมอนสเตอร์ โดยจะต้อง implements interface MonsterParty
 นิสิตอาจให้คลาสนี้มี method ที่ override จาก interface MonsterParty เพียง method เดียวก็ได้ เช่น

```
@Override
public ArrayList<Monster> createMonsters() {
    ArrayList<Monster> monsters = new ArrayList<>();
    monsters.add(new Monster("Thief Bug", 500));
    monsters.add(new Monster("Ant Egg", 5000));
    monsters.add(new Monster("Poring", 10));
    monsters.add(new Monster("Rare Mandragora", 38));
    return monsters;
}
```

4. สร้างคลาส MainProgram ที่จะแสดงสมาชิกพรรค และสมาชิกพรรคที่เรียงลำดับตาม hit point แล้ว ดังนี้

```
//elab-source:MainProgram.java
//elab-mainclass:MainProgram
import java.util.ArrayList;
import java.util.Collections;

public class MainProgram {
    public static void main(String[] args) {
        MonsterParty party = new MyParty();
        ArrayList<Monster> monsters = party.createMonsters();
        System.out.println(monsters);
        Collections.sort(monsters);
        System.out.println(monsters);
    }
}
```

หากทำถูกต้องนิสิตควรจะได้ผลลัพธ์ 2 บรรทัด บรรทัดแรกคือข้อมูลในลิสต์ของมอนสเตอร์ที่ยังไม่เรียงลำดับ
 บรรทัดที่สองคือข้อมูลในลิสต์ของมอนสเตอร์ที่เรียงลำดับตามค่า hit point จากมากไปหาน้อย

ให้นิสิตส่งโค้ดทั้งหมดเมื่อสำเร็จ ยกเว้นคลาส MyParty