6610402230 สิริสุข ทนธรรม

แบบฝึกหัด เรื่อง Design Pattern I

วิชา 01418471 Software Design and Development

ข้อ 1

พิจารณาคลาส GameManager ด้านล่าง ซึ่งต้องการให้เป็น Singleton แต่เขียนไม่ถูกต้อง จงแก้โค้ดให้ถูกต้อง

```
public class GameManager {

	public GameManager manager;
	public static List<Character> characters;

	public GameManager() {
		this.characters = new ArrayList<Character>();
	}

	public GameManager getManager() {
		return manager;
	}
	public void addCharacter(Character c) { .... }
	...
}
```

```
// 6610402230 Sirisuk Tharntham
// fix class to be singleton
import java.util.ArrayList;
import java.util.List;

public class GameManager {  3 usages
    public static GameManager manager; //store only one object in static  3 usages
    public List<Character> characters;  2 usages
    private GameManager() { this.characters = new ArrayList<Character>(); }
    public static GameManager getManager(){ //access or create new object if null  no usages
        if(manager == null) manager = new GameManager();
        return manager;

    }

    public void addCharacter(Character c) { characters.add(c); }

}
```

6610402230 ศิริสุข ทานธีรธรรม

ข้อ 2

พิจารณาโปรแกรมจองทัวร์ท่องเที่ยว โดยทัวร์จะมีแบบเดี่ยวกับแบบแพกเกจ ซึ่งในแพกเกจจะประกอบด้วยทัวร์หลายทัวร์ สำหรับทัวร์ทั้งสองแบบ เราต้องการข้อมูลชื่อทัวร์ ข้อมูลราคา และข้อมูลจำนวนที่นั่ง

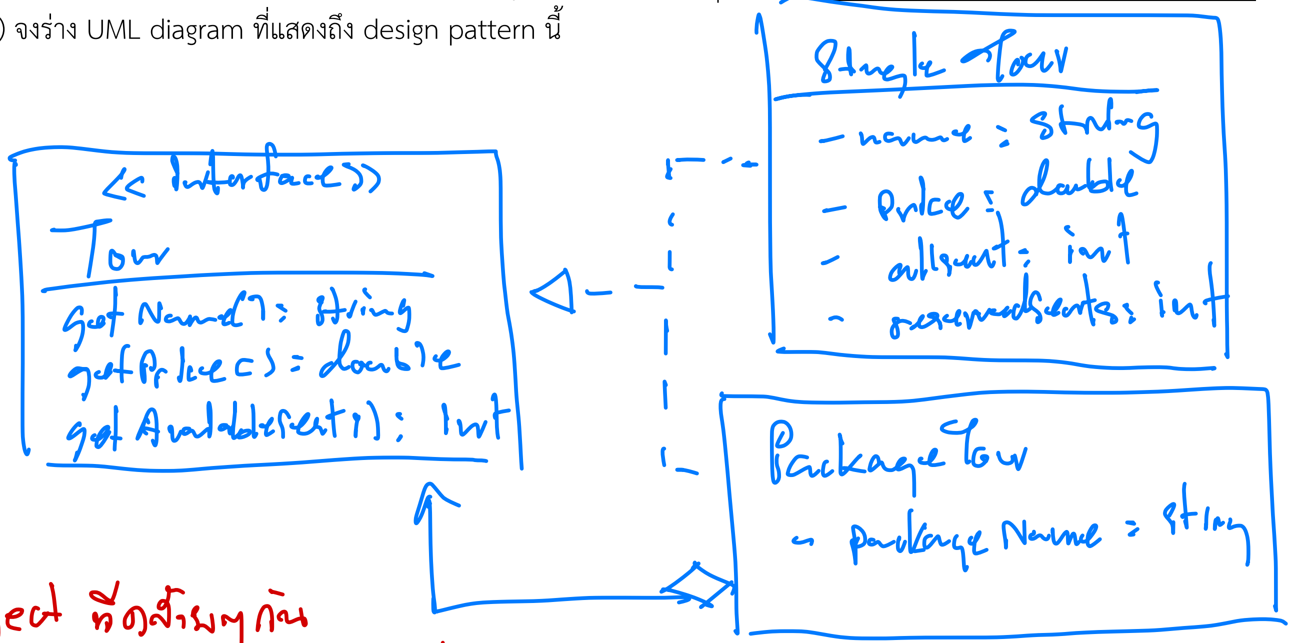
- สำหรับทัวร์แบบเดี่ยว ข้อมูลชื่อ ราคา และจำนวนที่นั่ง จะเป็นของทัวร์นั้น ๆ
- สำหรับทัวร์แบบแพกเกจ ข้อมูลชื่อจะเป็นชื่อแพกเกจและข้อมูลชื่อทัวร์ทุกทัวร์ในแพกเกจด้วย ราคาของแพกเกจจะเป็นราคารวมของทัวร์ทั้งหมดในนั้น แต่จะได้ลด 10% ส่วนข้อมูลจำนวนที่นั่งจะเท่ากับจำนวนที่นั่งที่จะจองได้ทั้งแพกเกจ เช่น ถ้าแพกเกจมีทัวร์ตลาดน้ำกับทัวร์วัดพระแก้ว โดยทัวร์ตลาดน้ำมี 10 ที่นั่งและทัวร์วัดพระแก้วมี 12 ที่นั่ง แพกเกจนี้จะมีจำนวนที่นั่งพอแค่ 10 ที่นั่ง

ตัวอย่างโค้ด

```
public interface Tour {
      String getName();
      double getPrice();
      int getAvailableSeats();
}
public class SingleTour implements Tour {
      private String name;
      private double price;
      private int allSeats;
      private int reservedSeats;

      // constructor and get/set methods ...
}
public class PackageTour implements Tour {
      private String packageName;
      private List<Tour> tours = new ArrayList<Tour>();

      @Override
      public double getPrice() {
            double total = 0;
            for (Tour t : tours)
                  total += t.getPrice();
            return total*0.9;
      }
      // other methods implemented similar to getPrice() ...
}
```

(a) จงตอบว่า การออกแบบในลักษณะนี้ ใช้ design pattern ใดมาประยุกต์ **ตอบ** composite pattern

(b) จงร่าง UML diagram ที่แสดงถึง design pattern นี้



object ที่ต่อสืบทอดกัน

+ มี sub-module (ส่วนย่อย)

66104023๑ ศิริกุล ทนธรรม

ข้อ 3

คุณต้องการสร้างโปรแกรมที่อ่านข้อมูลจำนวนมากผ่านคลาส InputStream ของ java.io โดยต้องการแปลงข้อมูลตัวอักขระภาษาอังกฤษใด ๆ ให้เป็นตัวเล็ก (lowercase) ทั้งหมด คุณจึงสร้างคลาส LowercaseInputStream ที่รับออปเจ็ค InputStream ตัวอื่นเป็นพารามิเตอร์ใน constructor โดยคลาส InputStream จะมีเมธอด read() ในการอ่านข้อมูล และคลาส LowercaseInputStream ได้ override เมธอด read() ดังโค้ดต่อไปนี้

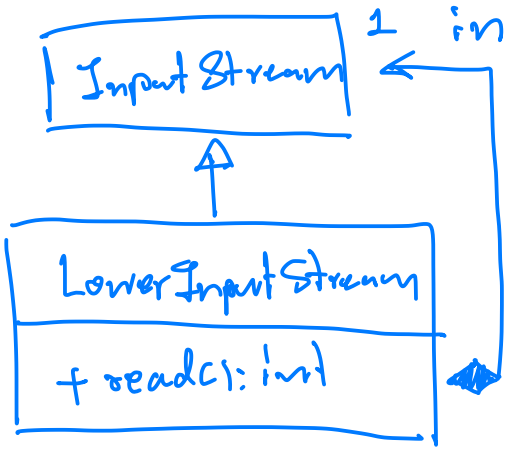
```
public class LowercaseInputStream extends InputStream {
    private InputStream in;

    public LowercaseInputStream(InputStream in) {
        this.in = in;
    }

    @Override
    public int read() throws IOException {
        int data = in.read();  // อ่านข้อมูลจาก in ที่รับผ่าน constructor
        if (Character.isAlphabetic(data)) { // หากเป็นอักขระภาษาอังกฤษ
            return Character.toLowerCase(data); // แปลงเป็นตัวเล็ก
        }
        return data;
    }
}
```

เมื่อใช้งานคลาส `LowercaseInputStream` เราสามารถสร้างออปเจ็ค `LowercaseInputStream` ได้ดังนี้ โดยให้คลาส `FileInputStream` เป็น subclass ของคลาส `InputStream`

```
FileInputStream fileIn = new FileInputStream("filename.txt");
LowercaseInputStream lowerIn = new LowercaseInputStream(fileIn);
char lower = (char) lowerIn.read();
```

(a) จงตอบว่า การออกแบบในลักษณะนี้ ใช้ design pattern ใดมาประยุกต์ **ตอบ** decorator pattern

(b) จงร่าง UML diagram ที่แสดงถึง design pattern นี้ โดยใช้คลาส `InputStream`, `LowercaseInputStream`, `FileInputStream`



+ มี class เพิ่ม
+ same interface
+ more behavior !

To read to lowercase

เป็นการทำ decorate ของเดิม ให้มี behavior ใหม่

มีโจทย์ในหน้าถัดไป.....

66104023๐ ศิริกุล ทนธรรม

ข้อ 4

พิจารณาโปรเจควิชา Software Construction ซึ่งต้องวาดรูปสัตว์หลายชนิด นิสิตคนหนึ่งได้ออกแบบโปรแกรมและเขียนเมธอด draw() ในคลาส Animal ดังโค้ดด้านล่าง เนื่องจากสัตว์ทุกชนิดต้องมีการวาดหัว ตัว และขา เหมือนกัน เช่น ทั้งสิงโตและยีราฟจะมีหัว ตัว และขา แต่รายละเอียดการวาดหัวหรือช่วงตัวจะแตกต่างกัน เช่น หัวสิงโตจะมีแผงคอ หัวยีราฟจะยาว เป็นต้น (นิสิตคนนี้วาดขาสัตว์ส่วนใหญ่เหมือนกัน จึง implement เมธอด drawLegs ไว้ในคลาส Animal ด้วย)

```
public abstract class Animal {
    ...
  public Animal(Color bodyColor) {
    ...
  }

  public void draw() {
    drawHead();
    drawBody();
    drawLegs();
  }

  public abstract void drawHead();
  public abstract void drawBody();
  public void drawLegs() {
    ...
  }
}
```

same all class

different

```
public class Lion extends Animal {

  public Lion(Color bodyColor) {
    super(bodyColor);
  }

  @Override
  public void drawHead() {
      // วาดแผงคอ
  }

  @Override
  public void drawBody() {
        // วาดตัวสีเดียวกัน
  }

}
```

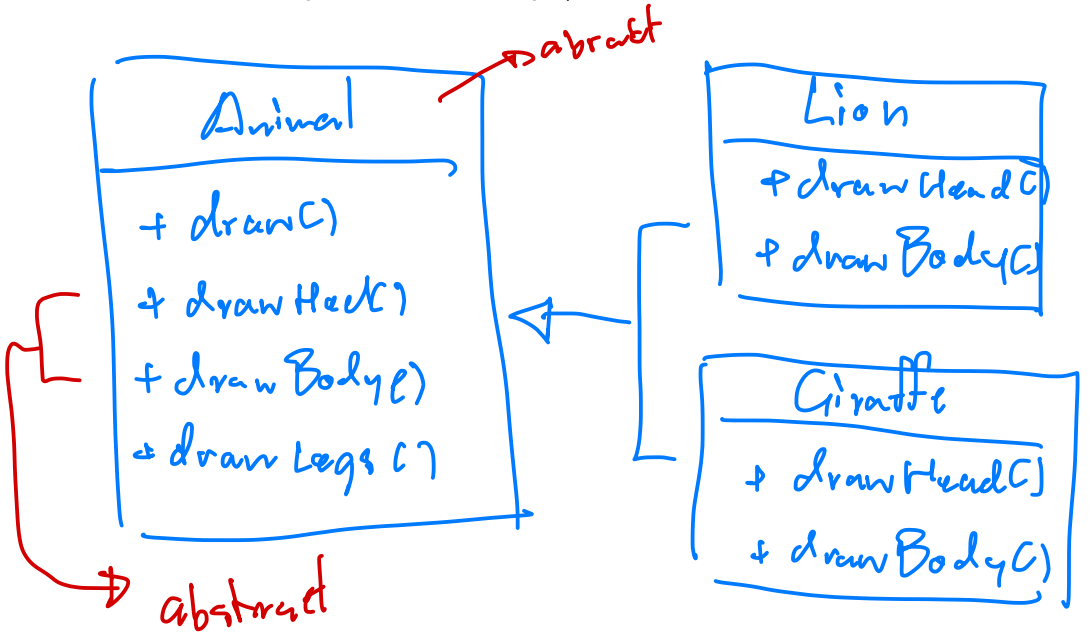
new behavior

```
public class Giraffe extends Animal {

  public Giraffe(Color bodyColor) {
    super(bodyColor);
  }

  @Override
  public void drawHead() {
      // วาดคอยาว ๆ
  }

  @Override
  public void drawBody() {
      // วาดตัวลาย ๆ
  }

}
```

new behavior

(a) จงตอบว่า การออกแบบในลักษณะนี้ ใช้ design pattern ใดมาประยุกต์ **ตอบ** Template method pattern

(b) จงร่าง UML diagram ที่แสดงถึง design pattern นี้ โดยให้มีคลาส Animal, Lion, Giraffe



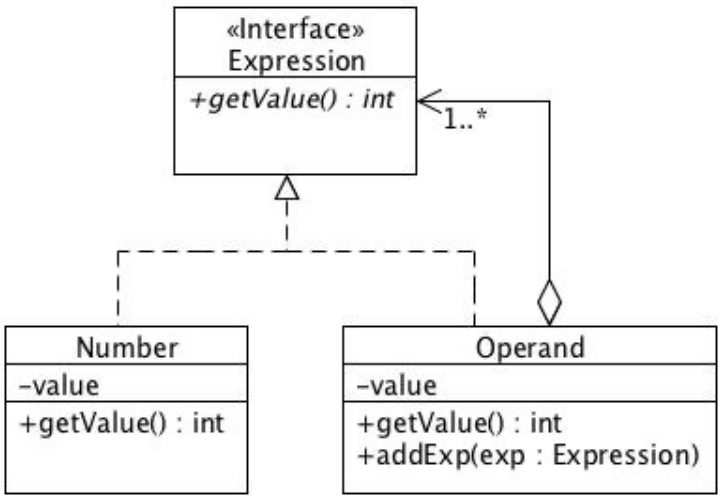
* มี same behavior ทุก subclass

ถ้า มี behavior ที่เปลี่ยนตาม subclass (Implement abstract method)

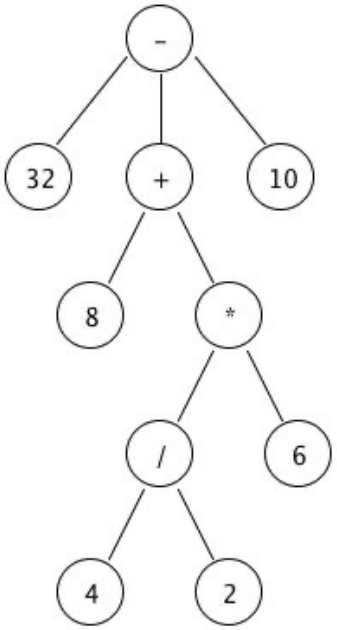
6610402230 ศิริสุข ทนโรม

ข้อ 5

จงเขียนโปรแกรมด้วย Composite Pattern ในการคำนวณค่านิพจน์คณิตศาสตร์ (Arithmetic Expression) โดยใช้โครงสร้างคลาสดังต่อไปนี้



โดยลองทดสอบกับนิพจน์ (32 - (8 + ((4 / 2) * 6)) - 10) โดยเมื่อสร้าง object จากคลาสด้านบน เราจะได้เป็นโครงสร้างต่อไปนี้ และเมื่อเราเรียกเมธอด getValue() จากโหนดสูงสุดเราจะได้ผลลัพธ์เป็น 2 (อาจลองใช้ enum มาช่วยในการ implement ได้ หรือใช้ strategy pattern มาช่วย)



* In the Compressed Code folder!