

MIDTERM 13.30 -16.00 อนุญาตให้เปิดเฉพาะ eclipse (ดูเล็บเก่าได้) , ไฟล์ slide ของวิชานี้ที่ให้ทางเว็บ และ Help file ของ Java ที่ให้ เท่านั้น เอกสารเอาเข้าห้องได้ แต่ห้ามส่งหรือสื่อสารกันด้วยวิธีใดๆทั้งสิ้น

เราต้องการเขียนเกมต่อสู้ โดยจะสร้างตัวละครมาสองประเภทแล้วเอามาทดลองสู้กัน ตัวละครที่เอามาใส่ในเกมต่อสู้นั้น จะต้องอิมพลีเม้นท์อินเตอร์เฟซ Fighter ซึ่งมีฟังก์ชันให้เรียกใช้ดังนี้

```
public Object getOpponent();  // ระบุตัวละครฝ่ายตรงข้ามออกมา
public void setOpponent(Object b);  // ตั้งค่าศัตรูที่จะสู้ด้วยคือ b
public void move();  // เมธอดนี้ทำการตรวจสอบสภาพของตัวละครฝ่ายเราและฝ่ายตรงข้าม และทำการเรียก attack ด้วย
// พารามิเตอร์ต่างๆขึ้นกับสภาพของตัวละครทั้งสองฝ่าย
public void attack(int attackType);  // สั่งโจมตีตัวละครฝ่ายตรงข้ามด้วยชนิดของท่าต่างๆกันไป จำนวนชนิดของ
// ท่านั้นแล้วแต่คลาสของตัวละคร
public void attacked(int attackLevel);  // จะถูกเรียกจากเมธอด attack ของฝ่ายตรงข้าม เมธอดนี้จะ
// ไปทำการลดพลังชีวิตของตัวละครตาม attackLevel ซึ่ง attackLevel มีค่า 1 ถึง 3
public double getLifePercent();  // ระบุค่าพลังชีวิตของตัวละคร คิดเป็นเปอร์เซ็นต์จากค่าตัวเลขพลังชีวิตเต็ม ค่าพลัง
// ชีวิตเต็มของแต่ละคลาสนั้นไม่เท่ากัน ทั้งนี้การคิดค่าเป็นเปอร์เซ็นต์ในส่วนต่างๆของโปรแกรมจะมีส่วนช่วยในการ scale ตัวละครจากคลาสสองคลาสให้เอามาต่อสู้
// กันได้สูสีหน่อย
public double getStrengthPercent();  // ระบุค่าความแข็งแกร่งของตัวละคร คิดเป็นเปอร์เซ็นต์ โดยเทียบกับ
// ผลรวมของค่า strength กับ defence (ค่าสองค่านี้รวมกันได้ 100%) ค่าความแข็งแกร่งนี้ ยิ่งมาก ศัตรูจะยิ่งพลังชีวิตลดมากตอนถูกโจมตี
public double getDefencePercent();  // ระบุค่าพลังป้องกันของตัวละคร คิดเป็นเปอร์เซ็นต์ โดยเทียบกับผลรวมของค่า
// strength กับ defence ค่าพลังป้องกันนี้ ยิ่งมาก พลังชีวิตจะลดลงน้อยเมื่อถูกศัตรูโจมตี
public int getRecoverTime();  // ระบุค่าตัวเลขที่บอกเวลาในการฟื้นตัว หลังจากออกท่าโจมตีศัตรูไป ถ้าค่านี้มากกว่าศูนย์
// ตัวละครจะไม่สามารถโจมตีศัตรูได้
public void setRecoverTime(int time);  // ตั้งค่าตัวเลขเวลาในการฟื้นตัว
```

1. เขียนตัวอินเตอร์เฟซ Fighter นี้ให้เรียบร้อย

ให้โปรแกรมทดลองการต่อสู้ของ Fighter สองชนิด มีชื่อคลาสว่า FightingGame และมี instance variable หรือ field ดังนี้

- private player1: เป็น Fighter
- private player2: เป็น Fighter
- private winner: เป็น Fighter

คอนสตรัคเตอร์:

- ต้องรับ Fighter เข้ามาสองตัว เซ็ต player1 กับ player2 ให้เป็นตามนั้น แล้วตั้งค่า winner ให้เป็น null

เมธอด:

- `public void gameLoop()`: เมธอดนี้วนลูปตลอด ในแต่ละครั้งให้เรียก `move()` ของ `player1` กับ `player2` แล้วจึงหาว่าใครคือผู้ชนะ (อาจมีน็อคทั้งคู่ก็ได้) พอได้ตัวคนชนะ หรือพอน็อคทั้งคู่ ต้องออกจากลูป แสดงผู้ชนะ หรือ น็อคทั้งคู่ ออกหน้าจอ และจบเมธอดทันที
- `public static void main(...)`: เมธอดนี้สร้างตัว `Fighter` มาสองตัว ตัวหนึ่งสร้างจากคลาส `FinalFantasyFighter` อีกตัวหนึ่งสร้างจากคลาส `KungFuFighter` (เดี๋ยวจะมีรายละเอียดการเขียนสองคลาสนี้) พอสร้างขึ้นมาแล้วก็ให้ทั้งสองตัวรับรู้ศัตรูคือตัวไหน จากนั้น `new FightingGame` ขึ้นมาแล้วเรียกใช้ `gameLoop()`

2. เขียนคลาส `FightingGame` ให้เรียบร้อย อาจต้องดูกลับไปกลับมาระหว่างคลาสนี้กับ `Fighter` ทั้งสองคลาสเพื่อให้เขียนได้ถูกต้อง

เรามีคลาส `KungFuMan` (ไฟล์เอาจากหน้าเว็บ) ซึ่งเคยถูกนำไปใช้เขียนเกมอื่น ที่นี้เราต้องการเอาคลาสนี้มาใช้สร้างเป็น `Fighter` เพราะว่ามีหลายองค์ประกอบคล้ายกัน แต่มีเงื่อนไขว่าต้องไม่เปลี่ยนแปลงอะไรในคลาส `KungFuMan` ดังนั้นต้องเปลี่ยนคลาสที่ `extends` จากคลาส `KungFuMan` นี้แทน ให้คลาสที่ `extends` จาก `KungFuMan` นี้ชื่อว่าคลาส `KungFuFighter` และให้มีเมธอดต่างๆดังนี้

คอนสตรัคเตอร์:

- เขียนก็ได้ทั้งคู่ แต่ให้ค่าตัวเลขต่างๆเซ็ทเหมือนกับในคอนสตรัคเตอร์ของซูเปอร์คลาส

เมธอด:

- `getDefencePercent()`, `getLifePercent()`, `getStrengthPercent()`, `getRecoverTime()`: เขียนเพิ่มเข้ามา ให้ถูกต้อง
- `attacked`: ให้ `override` เมธอดนี้ของซูเปอร์คลาสซะ ใช้วิธีคิดเหมือนกันแต่ทำการคำนวณเป็นค่าร้อยละแทน แล้วค่อยเอาพลังชีวิตที่เหลือ(ที่เป็นค่าร้อยละของพลังชีวิตเต็ม) ไปคำนวณหาพลังชีวิตของตัวละครที่เหลือจริง
- `attack`: ให้ `override` เมธอดนี้ของซูเปอร์คลาสซะ โดยให้การทำงานเหมือนเดิม แต่ใช้กับศัตรูที่เป็น `Fighter` ได้
- `move`: ให้ `override` เมธอดนี้ของซูเปอร์คลาสซะ โดยการทำงานจะเป็นดังนี้:
 - ถ้าตัวละครฟื้นจากการ `attack` แล้ว และศัตรูมีพลังเหลือน้อยกว่าหรือเท่ากับ 25% ให้ใช้ท่า `attack`

ชนิดที่ 3

- ถ้าตัวละครฟื้นจากการ **attack** แล้ว และศัตรูมีพลังเหลือตั้งแต่ 25% ขึ้นไปจนถึง 65% ให้ใช้ทำ **attack** ชนิดที่ 2
- ถ้าตัวละครฟื้นจากการ **attack** แล้ว และ ศัตรูมีพลังเหลือมากกว่า 65% ขึ้นไป ให้ใช้ทำ **attack** ชนิดที่ 1
- ถ้าตัวละครยังไม่ฟื้นจากการ **attack** ให้ลดค่าเวลาฟื้นตัวลงหนึ่งหน่วย

3. จงเขียนคลาส **KungFuFighter** ให้เรียบร้อย อย่าลืมว่าต้องสามารถเอาออบเจกต์ของคลาสนี้ไปใช้ได้ในฐานะ **Fighter**

คลาสที่จะเอามาใช้เป็น **Fighter** ต่อไป ซึ่งจะนำไปทดลองต่อสู้กับ **KungFuFighter** มีชื่อว่าคลาส **FinalFantasyFighter** ซึ่งคลาสนี้มีฟิลด์ดังนี้

```
private static final int FF_MAXSTAT = 255;  // ค่านี้เป็นค่าคงที่ที่ใช้บ่อยในเกมประเภท RPG
private Fighter opponent;  // คู่ต่อสู้
private int timeToRecover;  // เวลาในการฟื้นตัวหลังจากการออกทำ attack ซึ่งขึ้นกับท่าที่ได้ออกไป
// ในตอนเริ่มเล่น ค่าเวลานี้จะเท่ากับ 0 ถ้าเวลาไหนค่านี้มีค่ามากกว่า 0 นั้นหมายความว่าตัวละครจะ attack ไม่ได้
private int life;  // พลังชีวิต ซึ่งเริ่มต้นจะมีค่าเท่ากับ 255
private int strength;  // ค่าความแข็งแกร่ง เขา 255 ลบค่านี้จะได้ค่าพลังป้องกันของตัวละคร ในตอนที่สร้างตัวละครขึ้นมา ค่า strength จะมีค่า 130
```

คอนสตรัคเตอร์:

- สร้างให้รับพารามิเตอร์น้อยที่สุดเท่าที่จะทำได้ สามารถมีได้มากกว่าหนึ่งคอนสตรัคเตอร์

เมธอด

- **Get** กับ **set** เขียนให้ครบ
- **public void attacked(int attackLevel):** วิธีคำนวณเหมือนกับในคลาส **KungFuFighter**
- **public void attack(int attackType):**
 - ถ้า **attackType** เท่ากับ 1 ให้เรียก **attacked(1)** กับศัตรู แล้วให้ปรีน "light Lunge" ออกหน้าจอ
 - ถ้า **attackType** เท่ากับ 2 ให้เรียก **attacked(2)** กับศัตรู แล้วให้ปรีน "medium Lunge" ออกหน้าจอ
 - ถ้า **attackType** เท่ากับ 3 ให้เรียก **attacked(3)** กับศัตรู แล้วให้ปรีน "heavy Lunge" ออกหน้าจอ

- นอกนั้นจะถือว่าการ **attack** นั้นไม่ประสบผล ให้พิมพ์ว่า "FF fighter move failed." ออกหน้าจอ
- **public void** move () : ใช้กรณีการทดสอบแบบเดียวกับใน KungFuFighter เลย แต่จะสลับลำดับการเรียก **attack** ซะ โดยถ้าตัวละครฟื้นจากการ **attack** แล้ว และศัตรูมีพลังเหลือน้อยกว่าหรือเท่ากับ 25% ให้ใช้ท่า **attack** ชนิดที่ 1 แทน ส่วนถ้าตัวละครฟื้นจากการ **attack** แล้ว และ ศัตรูมีพลังเหลือน้อยกว่า 65% ขึ้นไป ให้ใช้ท่า **attack** ชนิดที่ 3

4. จงเขียนคลาส **FinalFantasyFighter** ให้เรียบร้อยอย่าลืมว่าต้องสามารถเอาออบเจกต์ของคลาสนี้ไปใช้ได้
ในฐานะ **Fighter**

5. ลองรัน **FightingGame** ดู เอาผลการรันใส่ **text file** แล้วเอา **text file** นี้ ตั้งชื่อว่า **<id>_runResult.txt**
เอาใส่เข้ามาใน **jar file** ที่จะต้องส่งนี้

6. วิธีการส่ง

- ตั้งชื่อโปรเจกต์ **<id>_progMethMidterm**
- สำหรับแต่ละ **.java** ไฟล์ ให้ขึ้นต้นไฟล์ด้วยข้อมูลของตน ดังนี้

/* รหัสนิสิต

* ชื่อ-นามสกุล

*หมายเลขห้อง หมายเลขเครื่อง

*/
- Export jar file ออกมาให้มีชอร์สโค้ดด้วย ตั้งชื่อไฟล์เป็น **<id>_progMethMidterm.jar**
- เอา **jar file** ใส่ไว้ใน **c:\temp** เปิดเครื่องกับ **eclipse** ทิ้งไว้ แล้วออกจากห้อง