

# ใบงานการทดลองที่ 11

## เรื่อง การใช้งาน Abstract และ Interface

### 1. จุดประสงค์ทั่วไป

- 1.1. รู้และเข้าใจการกำหนดวัตถุ การใช้วัตถุ การซ่อนวัตถุ และการสืบทอดประเภทของวัตถุ
- 1.2. รู้และเข้าใจโครงสร้างของโปรแกรมเชิงวัตถุ

### 2. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

### 3. ทฤษฎีการทดลอง

- 3.1. Abstract Class คืออะไร? มีลักษณะการทำงานอย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

คือ Class ที่เอาไว้ซ่อนรายละเอียดชัดเจน คือจะมี Abstract Method ซึ่ง Abstract Method จะไม่มีรายละเอียดของ Method อยู่ข้างใน  
ถ้าอยากจะใช้งานต้องสืบทอด ไปอีกทีหนึ่ง

- 3.2. Interfaces คืออะไร? มี ลักษณะการทำงานอย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

คือ Abstract Class ที่สมบรูณ์แบบ สมบรูณ์แบบในที่นี้คือใน Method ของ Interfaces จะไม่มีรายละเอียดอยู่เลย  
ถ้าอยากจะใช้งานต้อง implements ไป ถึงจะใช้งานได้ Interfaces ไม่สามารถที่จะสร้าง instance ตรงๆ ได้จะต้องสืบทอดไป  
แล้วไปสร้าง instance ใน class ลูกอีกทีหนึ่งถึงจะทำได้

- 3.3. คำสั่ง extends และ implements มีการใช้งานที่แตกต่างกันอย่างไร?

Extends ใช้กับ Class และ Abstract Class Implements ใช้กับ Interfaces

- 3.4. ภายใน Abstract Class มี Constructor หรือไม่? เพราะเหตุใด?

ไม่ได้เพราะมีการประกาศ Properties ใน class

- 3.5. ภายใน Interface มี Constructor หรือไม่? เพราะเหตุใด?

ไม่มีเพราะใน Interface ไม่มี Properties

### 4. ลำดับขั้นการปฏิบัติการ

- 4.1. ให้ผู้เรียนสร้าง Abstract Class ของรถถัง(ClassicTank) โดยจะต้องมีรายละเอียดดังต่อไปนี้
- 4.1.1. Properties : HP เพื่อกำหนดค่าพลังให้กับรถถัง

- 4.1.2. Properties : Str เพื่อกำหนดค่าความแรงในการยิงของรถถัง
- 4.1.3. Properties : Vit เพื่อกำหนดค่าพลังป้องกันของรถถัง
- 4.1.4. Properties : BaseDamage เพื่อกำหนดค่าพลังการโจมตีพื้นฐาน
- 4.1.5. Method : SetHP( ) ; เพื่อทำการกำหนดค่าพลังเริ่มต้น
- 4.1.6. Method : GetHP( ) ; เพื่อตรวจสอบค่าพลัง ณ เวลาปัจจุบัน
- 4.1.7. Method : Attack( Tank Enemy ) ; เพื่อทำการยิงปืนใหญ่โจมตีศัตรู โดยการโจมตี จะเป็นการลดค่าพลังของรถถังฝั่งตรงกันข้าม (Enemy คือรถถังของศัตรู, Points คือค่าพลังโจมตีของเรา)
- 4.2. ให้ผู้เรียนสร้างคลาส NormalTank เพื่อสืบทอด ClassicTank เพื่อเขียนรายละเอียดของ Method ทั้งหมดอันได้แก่ SetHP() , GetHP() , Attack( Tank Enemy )
- 4.3. ในคลาสหลัก ให้สร้าง Instance จาก NormalTank อยู่จำนวน 2 คัน เพื่อทำการต่อสู้กัน โดยควรต้องมีบทบาทดังนี้
- 4.3.1. สร้างรถถัง A และ B ให้มีค่าพลังเบื้องต้นดังต่อไปนี้
- | ค่าสถานะ   | รถถัง A | รถถัง B |
|------------|---------|---------|
| HP         | 200     | 250     |
| Str        | 12      | 8       |
| Vit        | 9       | 10      |
| BaseDamage | 11      | 10      |
- 4.3.2. รถถังทั้ง A และ B ผลิตกันโจมตีซึ่งกันและกัน เพื่อมุ่งหวังให้ค่าพลังของฝั่งตรงกันข้ามลดลงจนค่า HP = 0
- 4.3.3. รายละเอียดของพลังการโจมตีสามารถคำนวณได้ตามสมการดังต่อไปนี้  

$$\text{DamagePoint} = \text{MyTank\_BaseDamage} * \text{Floor}(\text{MyTank\_Str} / \text{Enemy\_Vit}) * \text{Random}(0.7, 0.9)$$
- 4.3.4. แสดงผลการทำงานผ่าน Console เพื่อให้เห็นรายละเอียดค่าพลังปัจจุบันของรถถังแต่ละคัน พลังการโจมตี ณ ขณะนั้น จนกว่าจะมีรถถังคันใดคันหนึ่งมีค่า HP = 0

#### โค้ดโปรแกรมภายใน Abstract Class

```
package lab11;

abstract class ClassicTank {

    public int HP;
    public int Str;
    public int Vit;
    public int BaseDamage;

    public abstract void SetHP(int x);
    public abstract void GetHP();
    public abstract void Attack(int x);
}
```

#### โค้ดโปรแกรมภายใน NormalTank

```

1 package lab11;
2
3 public class NmTank1 extends ClassicTank{
4
5     @Override
6     public void SetHP(int x) {
7         HP = x;
8     }
9
10    @Override
11    public void GetHP() {
12        System.out.println("Tank 1 Have HP " + HP);
13    }
14
15    @Override
16    public void Attack(int x) {
17        System.out.println("Tank 1 Take DMG " + x);
18        HP = HP - x;
19    }
20

```

```

1 package lab11;
2
3 public class NmTank2 extends ClassicTank{
4
5     @Override
6     public void SetHP(int x) {
7         HP = x;
8     }
9
10    @Override
11    public void GetHP() {
12        System.out.println("Tank 2 Have HP " + HP);
13    }
14
15    @Override
16    public void Attack(int x) {
17        System.out.println("Tank 2 Take DMG " + x);
18        HP = HP - x;
19    }
20

```

#### โค้ดโปรแกรมภายในฟังก์ชันการทำงานหลัก

```

package lab11;

import java.lang.Math;

public class main {
    public static float random(double d, double e) {
        return (float) ( (float)(Math.random() * (e - d)) +d);
    }
    public static void main(String[] args) {

        int turn = 1;
        int DamagePoint = 0;

        NmTank1 Nt1 = new NmTank1();
        NmTank2 Nt2 = new NmTank2();

        //Tank A
        Nt1.SetHP(200);
        Nt1.Str = 12;
        Nt1.Vit = 9;
        Nt1.BaseDamage = 11;

        //Tank B
        Nt2.SetHP(250);
        Nt2.Str = 8;
        Nt2.Vit = 10;
        Nt2.BaseDamage = 10;

        System.out.println("////////////////////////");
        Nt1.GetHP();
        Nt2.GetHP();
    }
}

```

```

System.out.println("/////////////////////////////////");
System.out.println("Start");

do {
    System.out.println("/////////////////////////////////");
    System.out.println("Round = " + turn);
    if(turn%2 == 0) {
        //Tank1 ATTACK
        DamagePoint = (int) (Nt1.BaseDamage * Math.floorDiv( Nt1.Str , Nt2.Vit ) * random(0.7, 0.9) );
        Nt2.Attack(DamagePoint);
        Nt2.GetHP();
    } else {
        //Tank2 ATTACK
        //floor(8/9) = 0 Tank2 DamagePoint == 0 ?
        DamagePoint = (int) (Nt2.BaseDamage *
            Math.floorDiv( Nt2.Str , Nt1.Vit ) * random(0.7, 0.9) );
        Nt1.Attack(DamagePoint);
        Nt1.GetHP();
    }
    turn++;
    if( Nt1.HP <= 0 || Nt2.HP <= 0 ) {
        break;
    }
    System.out.println("/////////////////////////////////");
} while(turn != 0 );
System.out.println("/////////////////////////////////");
if(Nt1.HP <= 0 ) {
    System.out.println("Tank 2 WIN!!!!!!");
} else if(Nt2.HP <= 0 ) {
    System.out.println("Tank 1 WIN!!!!!!");
}
System.out.println("/////////////////////////////////");
}

```

#### ผลลัพธ์การทำงานของโปรแกรม

```

/////////////////////////////////
Tank 1 Have HP 50
Tank 2 Have HP 80
Start
/////////////////////////////////
Round = 1
Tank 1 Take DNG 0
Tank 1 Have HP 50
/////////////////////////////////
Round = 2
Tank 2 Take DNG 7
Tank 2 Have HP 73
/////////////////////////////////
Round = 3
Tank 1 Take DNG 0
Tank 1 Have HP 50
/////////////////////////////////
Round = 4
Tank 2 Take DNG 8
Tank 2 Have HP 65
/////////////////////////////////
Round = 5
Tank 1 Take DNG 0
Tank 1 Have HP 50
/////////////////////////////////
Round = 6
Tank 2 Take DNG 7
Tank 2 Have HP 58
/////////////////////////////////
Round = 16
Tank 2 Take DNG 7
Tank 2 Have HP 21
/////////////////////////////////
Round = 17
Tank 1 Take DNG 0
Tank 1 Have HP 50
/////////////////////////////////
Round = 18
Tank 2 Take DNG 7
Tank 2 Have HP 14
/////////////////////////////////
Round = 19
Tank 1 Take DNG 0
Tank 1 Have HP 50
/////////////////////////////////
Round = 20
Tank 2 Take DNG 8
Tank 2 Have HP 6
/////////////////////////////////
Round = 21
Tank 1 Take DNG 0
Tank 1 Have HP 50
/////////////////////////////////
Round = 22
Tank 2 Take DNG 7
Tank 2 Have HP -1
/////////////////////////////////
Tank 1 WIN!!!!!!
/////////////////////////////////

```

#### 4.4. เปลี่ยน Abstract Class ให้กลายเป็น Interfaces และเปรียบเทียบผลลัพธ์การทำงานของโปรแกรม

หลังจากเปลี่ยน Abstract Class เป็น Interface แล้ว เกิดอะไรขึ้นอย่าง? อธิบายพร้อมยกตัวอย่างประกอบให้ชัดเจน

```

abstract class ClassicTank {

    public int HP;
    public int Str;
    public int Vit;
    public int BaseDamage;

    public abstract void SetHP(int x);
    public abstract void GetHP();
    public abstract void Attack(int x);

}

public class NmTank1 extends ClassicTank{

```



5. สรุปผลการปฏิบัติการ

การใช้ Abstract Class กับ Interface มีการใช้งานที่คล้ายๆกันในส่วนที่ต่างกันตรงที่ Properties โดยใน Interface จะไม่สามารถประกาศ Properties ได้แต่ใน Abstract Class ทำได้จากการทดลองที่ให้ทำการสร้างรถถึง 2 คันมาสลับกันอิงแบบใช้ Abstract กับ Interface หากแก้ไขตรงตามเงื่อนไขแล้วพบว่า ผลลัพธ์ของ 2 แบบเหมือนกันและผลลัพธ์ของ DamagePoint ที่คำนวณได้จากสูตร  $Nt2.BaseDamage * Math.floorDiv(Nt2.Str, Nt1.Vit) * random(0.7, 0.9)$  พบว่าได้ 0 ตลอดเพราะ  $Math.floorDiv(Nt2.Str, Nt1.Vit)$  หากแทนค่าจะพบว่า  $Math.floorDiv(8, 9)$  จะได้ 0 แล้วคูณในสมการก็จะได้ 0 ( $10 * 0 * random(0.7, 0.9)$ )

6. คำถามท้ายการทดลอง

6.1. เมื่อใดจึงควรเลือกใช้งาน Abstract Class

เมื่อต้องเขียนโปรแกรมที่มี Properties ซ้ำกันเยอะๆ หรือมี Properties ที่เหมือนกันเยอะ เช่น HP STRDEF AGI เป็นต้น

6.2. เมื่อใดจึงควรเลือกใช้งาน Interface

เมื่อต้องเขียนโปรแกรมที่มี Properties ไม่ซ้ำกัน หรือมี Properties เฉพาะเยอะ