



Interface

saacsos



Interface

- เป็นเหมือน Class ที่รวบรวม methods ที่ยังไม่ระบุการทำงานภายใน (ยังไม่มี implementation) หรือมีเฉพาะ method definition
- สร้าง object จาก Interface ไม่ได้ (`new Interface()` ไม่ได้) จึงไม่เรียก Interface ว่า Class
- Class ที่นำ Interface ไปใช้ ต้องระบุการทำงานภายในของ method ที่มีใน Interface



ตัวอย่าง Interface Washable

```
interface Washable {  
    public void wash();  
}
```



ตัวอย่าง Interface Edible (กินได้)

```
interface Edible {  
    public int giveEnergy();  
}
```



ตัวอย่าง Interface VolumeChangeable

```
interface VolumeChangeable {  
    public void increaseVolume();  
    public void decreaseVolume();  
}
```



ตัวอย่าง Interface NetworkInterface

```
interface NetworkInterface {  
    public void connect(Network network);  
    public void disconnect(Network network);  
}
```



Class ที่นำ Interface ไปใช้

- ต้องระบุการทำงานภายในของ method ที่มีใน Interface
- Keyword `implements`



ตัวอย่าง Dog

```
class Dog implements Washable {  
    @override  
    public void wash() {  
        // wash my four legs  
        // wash my head  
        // wash my body  
        // wash my tail  
    }  
}
```




ตัวอย่าง Chicken

```
class Chicken implements Washable, Edible {  
    @Override  
    public void wash() {  
        // wash my two legs  
        // wash my head  
        // wash my body  
        // wash my tail  
    }  
    @Override  
    public int giveEnergy() {  
        return (int) (4 * weight);  
    }  
}
```



ตัวอย่าง Vegetable

```
class Vegetable implements Washable, Edible {  
    @Override  
    public void wash() {  
        // wash my leaves  
    }  
    @Override  
    public int giveEnergy() {  
        return (int)(0.5 * weight);  
    }  
}
```



ตัวอย่าง Radio

```
class Radio implements VolumeChangeable {  
    private int volume;  
    @Override  
    public void increaseVolume() {  
        volume = volume++ > MAX_VOLUME ? MAX_VOLUME : volume;  
    }  
    @Override  
    public void decreaseVolume() {  
        volume = volume-- < MIN_VOLUME ? MIN_VOLUME : volume;  
    }  
}
```



ตัวอย่าง Computer

```
class Computer implements VolumeChangeable, NetworkInterface {
    private int volume;

    @Override
    public void increaseVolume() {
        volume = ++volume > MAX_VOLUME ? MAX_VOLUME : volume;
    }

    @Override
    public void decreaseVolume() {
        volume = --volume < MIN_VOLUME ? MIN_VOLUME : volume;
    }

    @Override
    public void connect(Network network) { ... }
    ...
}
```



Polymorphism

saacsos



Polymorphism

- object มีได้หลายรูปแบบ ซึ่งสืบทอดมาจาก Superclass หรือ Interface
- การทำงานของ method ขึ้นกับ Implementation ที่ Override แล้วเป็นหลัก (ถ้าไม่มี override method จะใช้เหมือน Superclass)



Static Polymorphism

- Compile Time Polymorphism, Static Binding
- Overloading



Dynamic Polymorphism

- Runtime Polymorphism, Dynamic Method Dispatch
- Upcasting
- Overriding
 - Inheritance
 - Interface



Superclass Weapon

```
class Weapon {  
    private int damage;  
    private int durability;  
    public Weapon(int dmg, int dur) {  
        damage = dmg;  
        durability = dur;  
    }  
    public int getDamage() {  
        return damage;  
    }  
    public void use() {  
        if (--durability < 0) durability = 0;  
    }  
}
```

Weapon

- damage:int
- durability:int

- + Weapon(dmg, dur)
- + getDamage():int
- + use():int



Subclass ของ Weapon (Gun)

```
class Gun extends Weapon {  
    private int numAmmo;  
    private double accuracy;  
    public Gun(int dmg, int dur, int ammo, double acc) {  
        super(dmg, dur);  
        numAmmo = ammo;  
        accuracy = acc;  
    }  
    @Override  
    public void use() {  
        super.use();  
        if (--numAmmo < 0) numAmmo = 0;  
    }  
}
```

ตัวอย่าง - GameCharacter

```
class GameCharacter {  
    private int hp;  
    private String name;  
    private int damage;  
    private Weapon weapon;  
    public GameCharacter(String name, int hp, int damage, Weapon weapon)  
    {  
        this.name = name;  
        this.hp = hp;  
        this.damage = damage;  
        this.weapon = weapon;  
    }  
    public int getDamage() {  
        return damage + weapon.getDamage();  
    }  
}
```

ตัวอย่าง - GameCharacter (ต่อ)

```
class GameCharacter {  
    // ... (Code หน้าที่แล้ว)  
  
    public void isAttackedWith(int damage) {  
        hp -= damage;  
        if (hp < 0) hp = 0;  
    }  
  
    public void attack(GameCharacter enemy) {  
        weapon.use();  
        enemy.isAttackedWith(getDamage());  
    }  
}
```

ตัวอย่าง - GameCharacter: Weapon Polymorphism

```
GameCharacter slime = new GameCharacter(  
    "Slime", 100, 40, new Weapon(0, 0)  
);  
  
Weapon stick = new Weapon(3, 30);  
GameCharacter player = new GameCharacter("Inorin", 100, 20, stick);  
  
  
Gun rpg = new Gun(1200, 1, 1, 60);  
GameCharacter leon = new GameCharacter("Leon", 340, 44, rpg);  
  
  
Weapon a = new Gun();
```



ตัวอย่าง Interface Polymorphism Dog

```
class Dog implements Washable {  
    @Override  
    public void wash() {  
        // wash my four legs  
        // wash my head  
        // wash my body  
        // wash my tail  
    }  
  
    public void eat(Edible food) {  
        hp += food.giveEnergy();  
    }  
}
```



ตัวอย่าง - Interface Polymorphism: Edible

```
Dog dog = new Dog();
```

```
Chicken kfc = new Chicken();  
dog.eat(kfc);
```

```
Vegetable lettuce = new Vegetable();  
dog.eat(lettuce);
```



ตัวอย่าง - Remote Control

```
class RemoteControl {  
    public static void increaseVolume( VolumeChangeable device) {  
        device.increaseVolume();  
    }  
  
    public static void decreaseVolume( VolumeChangeable device) {  
        device.decreaseVolume();  
    }  
}
```




ตัวอย่าง - Polymorphism: VolumeChangeable

```
Computer com = new Computer();
```

```
Radio radio = new Radio();
```

```
RemoteControl.increaseVolume(com);
```

```
RemoteControl.decreaseVolume(radio);
```