

# 介面

TYIC桃高資訊社

# 介面

介面(**interface**)幾乎可以視為抽象類別  
但介面是專門用來定義方法  
而且介面通常定義的是動態抽象方法  
交由實作(**implement**)該介面的類別去定義  
介面也可以定義欄位，且皆為公開靜態不可變欄位  
介面的靜態方法預設為公開靜態方法  
介面的動態方法預設為公開動態抽象方法

# 介面

```
public class Main {
    public static void main(String[] args) {
        FlyingFish flyingFish = new FlyingFish();
        flyingFish.fly();
        flyingFish.swim();
        Aircraft aircraft1 = new Airplane();
        aircraft1.fly();
        Aircraft aircraft2 = new Helicopter();
        aircraft2.fly();
        Aircraft aircraft3 = new Aircraft() {
            @Override
            public void fly() {
                System.out.println("鸚鵡用翅膀飛");
            }
        };
        aircraft3.fly();
    }
}

interface CanSwim {
    void swim();
}

interface CanFly {
    void fly();
}

abstract class Aircraft implements CanFly {
}
```

```
class Fish implements CanSwim {
    @Override
    public void swim() {
        System.out.println("魚兒魚兒水中遊");
    }
}

class Airplane extends Aircraft {
    @Override
    public void fly() {
        System.out.println("飛機用引擎飛");
    }
}

class Helicopter extends Aircraft {
    @Override
    public void fly() {
        System.out.println("直升機用螺旋槳飛");
    }
}

class FlyingFish extends Fish implements CanFly {
    @Override
    public void fly() {
        System.out.println("飛魚用魚翅飛");
    }
}
```