

# Programming II



第12回 インタフェース(第11章後半)

塩澤秀和

# 12.1 多重継承の禁止

---

- 多重継承とは (p.377)
  - 1つのクラスが、複数の親クラスを(直接)継承すること
  - 複合的な機能を持つクラスの実現に便利なのだが...
  
- 多重継承の問題点 (p.445)
  - 異なる親クラスが先祖に同じクラスを持っていると(ひし形継承), 文法解釈や内部構造で競合が生じてしまう
  - プログラムがどう実行されるか分かりづらくなりやすい
  
- Javaの解決策 (p.447)
  - クラスの多重継承は禁止し, 「インタフェース」を提供
  - インタフェース = メソッド名だけを持つ特殊な抽象クラス

## 12.2 インタフェース

---

### □ インタフェース (p.438)

- 働きとしては、抽象メソッドだけの純粹抽象クラスと同じ

```
public interface インタフェース名 {  
    public abstract void method1();  
    public abstract int method2(int a);  
}
```

- アクセス指定を省略しても、「public abstract」になる

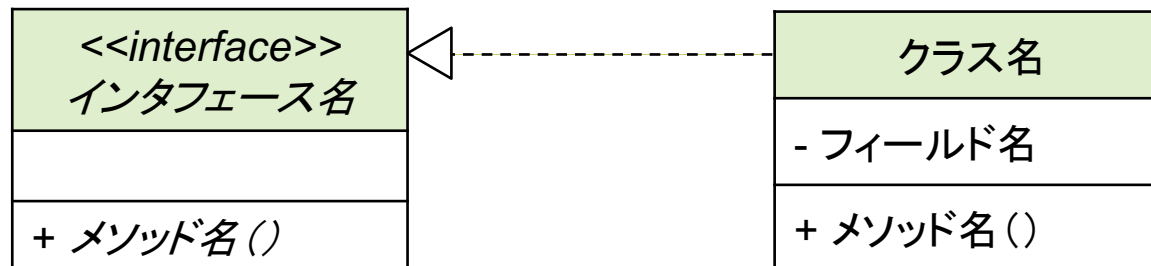
### □ インタフェースの“実装” (p.441)

- 継承は、extendsの代わりにimplementsを使う

```
class クラス名 implements インタフェース名 {  
    public void method1() { /* メソッドの処理内容 */ }  
    ...  
}
```

## 12.3 インタフェース

- インタフェースによる(擬似)多重継承(p.447)
  - クラスは複数の親インタフェースを持つことができる  
class クラス名 **implements** インタフェース1, インタフェース2...
  - クラスの継承と併用する場合は後に書く  
class クラス名 extends 親クラス **implements** インタフェース...
  - インタフェースからインタフェースを継承できる(extends)  
interface インタフェース名 **extends** インタフェース1...
  
- クラス図でのインタフェース



## 12.4 クラス図を描いてみよ

---

```
/* Fish.java */
public class Fish {
    public void swim() {
        System.out.println("すいすい~");
    }
}
```

```
/* FlyingFish.java */
public class FlyingFish extends Fish
    implements Flyable {
    public void swim() {
        System.out.println("スイスイ~");
    }
    public void fly() {
        System.out.println("パタパタ~");
    }
}
```

```
/* Flyable.java */
public interface Flyable {
    void fly();
}
```

```
/* Main.java */
public class Main {
    public static void main(String[] a) {
        FlyingFish f1 = new FlyingFish();
        f1.swim();
        f1.fly();
        Fish f2 = new FlyingFish();
        f2.swim();
        FlyingFish f3 = (FlyingFish) f2;
        f3.fly();
    }
}
```

## 12.6 インタフェースと多態性

---

- インタフェースの利用価値
  - 実装が空のインタフェースを継承する意味は？
  - インタフェースは多態性と組み合わせて効果を発揮する
  - 共通のインタフェースを持つ = 共通のコードで処理できる
  
- インタフェース型変数
  - インタフェースも抽象クラスと同様にデータ型として扱える
  - インタフェース型の変数, 配列, 引数等が定義できる
  - 例) `if (fish instanceof Flyable)`  
`Flyable f = (Flyable) fish;`
  - Javaでは, 別々のクラスに同じ名前のメソッドがあっても, 同一のインタフェースからの継承でなければ, 関係ない

# 12.5 演習

---

## □ 演習課題 2

- ClassYに必要なフィールドとメソッドを追加すること
- セッター, ゲッターの名前に対応するフィールドを作ること

# 第11回 まとめ

---

- 多重継承の禁止
  - 多重継承の問題点
- インタフェース
  - interface
  - implements
- インタフェースによる多重継承
- インタフェースの継承
  
- 補足 (p.451)
  - 新しいJavaでは, インタフェースの機能が拡張されている
  - defaultメソッド, staticメソッド, privateメソッド