

オブジェクト指向言語 Javaプログラミング(2010年度) 補足資料(05/10)

桑原 寛明
情報理工学部 情報システム学科

クラスの拡張

- Javaでは既存のクラスのインスタンスを生成するだけでなく、既存のクラスを拡張して新しいクラスを作成することができる
- 拡張の方法として**合成**と**継承**

合成の例

```
class Book {
    String title;
    int price;

    Book(String title, int price) {
        this.title = title;
        this.price = price;
    }

    public String getTitle() {
        return title;
    }

    public int getPrice() {
        return price;
    }

    public void setPrice(int price) {
        this.price = price;
    }
}
```

```
class OnlineBook {
    Book book;
    String website;
    OnlineBook(String title, int price,
                String website) {
        book = new Book(title, price);
        this.website = website;
    }
    public String getWebsite() {
        return website;
    }
    public String getTitle() {
        return "Online: " +
            book.getTitle();
    }
    public int getPrice() {
        return book.getPrice();
    }
    public void setPrice(int price) {
        book.setPrice(price);
    }
}
```

継承の例

```
class Book {
    String title;
    int price;

    Book(String title, int price) {
        this.title = title;
        this.price = price;
    }

    public String getTitle() {
        return title;
    }

    public int getPrice() {
        return price;
    }

    public void setPrice(int price) {
        this.price = price;
    }
}
```

```
class OnlineBook extends Book {
    String website;

    OnlineBook(String title, int price,
                String website) {
        super(title, price);
        this.website = website;
    }

    public String getWebsite() {
        return website;
    }

    public String getTitle() {
        return "Online: " + title;
    }
}
```

ここでクイズ

以下のコードの実行結果はどうなるだろうか。
OnlineBookクラスを合成で定義した場合と継承で定義した場合の双方を考えよ。

```
public static void main(String[] args) {  
    Book book = new OnlineBook("Title",2000,"URL");  
    System.out.println(book.getTitle());  
}
```

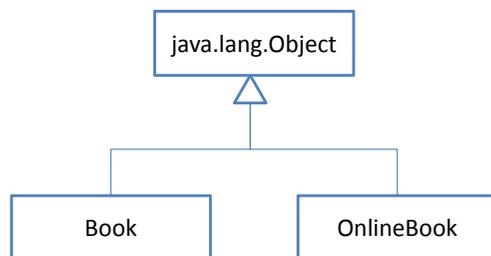
1. コンパイルエラー
2. 「Title」と表示される
3. 「Online: Title」と表示される

答え

- 合成の場合
 1. コンパイルエラー
- 継承の場合
 3. 「Online: Title」と表示される

解説(合成の場合)

- BookクラスとOnlineBookクラスは無関係
 - たまたま類似したメソッドを持っているだけ



解説(継承の場合)

- OnlineBookクラスがBookクラスを継承
 - OnlineBook **is a** Book (OnlineBookはBookである)
 - Book型変数にOnlineBookオブジェクトを代入可能
 - ただしBook型として扱われる
 - Bookのインタフェースのみ利用可能
- 呼び出されるメソッドは実行時のオブジェクトによって決まる
 - 変数bookにはOnlineBookオブジェクトが入っている
 - OnlineBookのgetTitleメソッドが実行される

動的束縛・多態性

```
public static void main(String[] args) {
    Book book = new Book("Title1", 1000);
    System.out.println(book.getTitle());

    book = new OnlineBook("Title2", 2000, "URL");
    System.out.println(book.getTitle());
}
```

- book.getTitle()の結果はbookが指すインスタンスのクラスによって決まる
 - 実行時に呼び出されるメソッドが決定される⇒動的束縛
 - 同じメソッド呼出しgetTitle()で異なる動作を実現する⇒多態性(多相性)

練習問題 9

- Vehicleクラスを拡張して、座席数と座っている人数を扱えるPassengerVehicleクラスを定義せよ
- 座席数を引数に取るコンストラクタを用意せよ
- 合成と継承の双方で作れ

```
class Vehicle {
    int speed;
    int direction;
    String owner;

    Vehicle(String owner) {
        this.speed = 0;
        this.direction = 0;
        this.owner = owner;
    }

    int getSpeed() {
        return speed;
    }

    void setSpeed(int speed) {
        this.speed = speed;
    }

    /* 以下省略 */
}
```

練習問題 9 (プログラム例・継承の場合)

```
class PassengerVehicle extends Vehicle {
    int passengerMax;
    int passengerNum;

    PassengerVehicle(String owner, int passengerMax) {
        super(owner);
        this.passengerMax = passengerMax;
    }

    int getMaxPassengers() {
        return passengerMax;
    }

    void addPassenger() {
        if (passengerNum < passengerMax) {
            passengerNum++;
        }
    }

    /* 以下省略 */
}
```