

JavaプログラミングⅡ

7回目 オーバーライド 課題

確認○×問題

次の各文は正しいか誤っているか答えなさい。

- (1) スーパークラスのメソッドと同じ名前、戻り値、引数の個数と型をもつメソッドをサブクラスで宣言することをメソッドのオーバーライドという
- (2) メソッドのオーバーライドとは、スーパークラスのメソッドに代わってサブクラスのメソッドが実行される機能のことである
- (3) スーパークラス型の変数にサブクラスのオブジェクトは代入できない
- (4) メソッドのオーバーライドはポリモーフィズムを実現する方法の1つである
- (5) サブクラスの中で `super.`修飾子を用いるとスーパークラスのフィールドやメソッドにアクセスできる
- (6) メソッドに `final` 修飾子をつけるとそのメソッドはサブクラスでオーバーライドすることができない
- (7) Java では2つ以上のスーパークラスを同時に継承してサブクラスを宣言できる
- (8) Java ではすべてのクラスは `Object` クラスのメンバーを継承する

難易度★★★

課題1 次にクラスAとB、これを使用するメインメソッドを示す。このコードの実行結果が下に示すようになるようにクラスAの `show()`メソッドをクラスBでオーバーライドするコードを空欄に宣言し、クラスBの宣言を完成させなさい。

ソースファイル名：Assignment7_1.java (main()メソッドがあるクラス名と一致させる)

```
class A
{
    private int x=10;
    public void show(){
        System.out.println(x);
    }
}
class B extends A
{
    private int y=20;
            クラスAの show()メソッドをオーバーライドするコード
    
}
class Assignment7_1
{
    public static void main(String[] args){
        A a=new B();
        a.show();
    }
}
```

〔実行結果〕

```
>java Assignment7_1
10
20
```

難易度★★★

課題2 次は、チケットクラス（スーパークラス）とこれを継承したコンサートチケットクラス（サブクラス）、これを利用するメインメソッドである。このコードの実行結果が下に示すようになるようにチケットクラスの show()メソッドをコンサートチケットクラスでオーバーライドするコードを空欄に宣言し、コンサートチケットクラスの宣言を完成させなさい。

ソースファイル名：Assignment7_2.java （main()メソッドがあるクラス名と一致させる）

// チケットクラス（スーパークラス）

```
class Ticket
{
    private String title;
    private int price;

    public Ticket(String t, int p){
        title=t;
        price=p;
    }
    public void show(){
        System.out.println("チケット名："+title);
        System.out.println("価格："+price+"円");
    }
}
```

// コン서트チケットクラス（サブクラス）

```
class Concert extends Ticket
```

```
{
    private String seat;

    public Concert(String t, int p, String s){
        super(t,p);
        seat=s;
    }

    チケットクラスの show() メソッドをオーバーライドするコード

}
```

```
class Assignment7_2
```

```
{
    public static void main(String[] args){
        Ticket tk = new Concert("矢沢永吉 2009",10000,"S 席");
        tk.show();
    }
}
```

〔実行結果〕

```
>java Assignment7_2
チケット名：矢沢永吉 2009
価格：10000 円
席番号：S 席
```

難易度★★★

課題3 次は建物クラス（スーパークラス）とこれを継承したマンションクラス（サブクラス）である。スーパークラスの表示メソッド `showBuilding()` が呼び出されると代わりにサブクラスの表示メソッドが実行されるようにしたい。スーパークラスの `showBuilding()` メソッドをオーバーライドするようにサブクラスの表示メソッドの宣言（破線矩形内）を変更しなさい。

〔建物を管理するクラス群〕

```
// 建物クラス（スーパークラス）
class Building
{
    private int floors; // 何階建
    private int height; // 高さ

    // 表示メソッド
    public void showBuilding(){
        System.out.println(floors+"階建");
        System.out.println("高さ："+height+"m");
    }
}

// マンションクラス（サブクラス）
class Mansion extends Building
{
    private int families; // 世帯数

    // 表示メソッド
    public void showMansion(){
        showBuilding();
        System.out.println("世帯数："+families);
    }
}
```

難易度★★★

課題4 すべてのクラスは `Object` クラスを継承する。`Object` クラスは `public String toString()`; メソッドをメンバーにもつ。`System.out.println()`;メソッドの宣言は機能的に次のようになっている。

```
public void println(Object x){
    x.toString() の戻り値を画面に出力する;
}
```

各自オリジナルのクラスを宣言しなさい。次に、`Object` クラスから継承している `toString()` メソッドをオーバーライドして、各自のクラスの情報に `System.out.println()`;メソッドで画面に表示できるようにしなさい。

難易度★★★

課題5 次は、MP3 プレーヤや携帯など電子ガジェットを管理するクラス構成である。このコードに、以下に示す3つの要件を満たすようにメンバーを追加/変更しなさい。

〔電子ガジェットを管理するクラス群〕

```
// スーパークラス Gadget
class Gadget{
    private String name; // 製品名
    private int price; // 価格
    public Gadget(String n, int p){
        name=n;
        price=p;
    }
    public void show() {
        System.out.println("製品名 : "+name+"、価格 : "+price+"円");
    }
}
// サブクラス Mp3Player
class Mp3Player extends Gadget{
    private int memory; // 容量 MB
    public Mp3Player(String n, int p, int m){
        super(n, p); // スーパークラスのコンストラクタを呼ぶ
        memory = m;
    }
    public void print(){
        System.out.println("[MP3Player]");
        show(); // 継承されたメソッドを使用
        System.out.println("メモリ : "+memory+"MB");
    }
}
// サブクラス CellPhone
class CellPhone extends Gadget{
    private String number; // 番号
    public CellPhone(String n, int p, String num){
        super(n, p); // スーパークラスのコンストラクタを呼ぶ
        number = num;
    }
    public void print(){
        System.out.println("[CellPhone]");
        show(); // 継承されたメソッドを使用
        System.out.println("番号 : "+number);
    }
}
```

〔要件一覧〕

要件1 (メンバー変更) スーパークラスのデータ表示メソッド show()をオーバーライドするようにサブクラスのデータ表示メソッド print()を変更しなさい。

要件2 (メンバー追加) サブクラス Mp3Player のメンバーとして"Mp3Player"を文字列としてもつ String 型の private メンバー定数を宣言しなさい。クラス CellPhone も同様に。

要件3 (メンバー追加) スーパークラス Gadget、サブクラス Mp3Player と CellPhone のそれぞれにおいて、要件2の定数値、製品名、価格、容量または番号を戻り値とするようにメソッド String toString()を宣言 (オーバーライド) しなさい。

難易度★★★

課題6 メインメソッド内でスーパークラス **Gadget** の配列（配列要素は数個でよい）を作成し、各配列要素にクラス **Mp3Playr** のオブジェクトまたはクラス **CellPhone** のオブジェクトを生成して代入しなさい。オーバーライドされたデータ表示メソッド（スーパークラス **Gadget** のメソッド **show()**）を呼び出して各オブジェクトのデータがオブジェクトに応じて適切に出力されることを確認しなさい。さらに、各オブジェクトに対して **System.out.println(オブジェクト);** を実行し、オーバーライドした **toString()**メソッドによるクラス情報の出力を確認しなさい。