

**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と、これらのメソッドはオーバーライドの関係になる
- (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拡張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 `main()` メソッドを示します。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同じにします)

```
class A
{
    private int x=10;
    public void show(){
        System.out.println(x);
    }
}
class B extends A
{
    private int y=20;
    -----}{
        super.show();
        System.out.println(y);
    }
}
class Assignment7_1
{
    public static void main(String[] args){
        A a=new B();
        a.show();
    }
}
```

〔実行結果〕

---

10  
20

■難易度★☆☆

**課題 2** 次は、チケットクラス（スーパークラス）とこれを拡張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 main() メソッドです。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同じにします)

```
// 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");
    }
}

// コンサートチケットクラス（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メソッドを宣言します

}

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〕

---

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）とこれを拡張したマンションクラス（サブクラス）です。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ます。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）を変更してください。ヒント：super. 修飾子も利用します

〔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 クラスのメンバーを継承します。Object クラスは public String toString(); メソッドをメンバーに持ちます。System.out.println(); メソッドの宣言は機能的に次のようになっています。

〔メソッド println() の宣言〕

---

```
public void println(Object x){
    // x.toString() の戻り値を画面に出力する処理をここで行います
}
```

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 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す。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び
        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び
        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 において"製品名/価格"を戻り値とするようにメソッド toString()をオーバーライドしなさい。さらに、サブクラス Mp3Player と CellPhone において、"要件 2 の定数値/製品名/価格/容量 or 番号"を戻り値とするようにメソッド toString()をオーバーライドしなさい。

■難易度★★☆

**課題 6** main()メソッド内でスーパークラス Gadget の配列（配列要素は 2~3 個で）を作成し、各配列要素にクラス Mp3Player 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、各オブジェクトを引数に与えて System.out.println(); 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出力の動作を確認しなさい。