

**確認〇×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) スーパークラスのメソッドをサブクラスでオーバーライドすると、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ようになる
- (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` クラスを継承する

**課題1** 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を変更しなさい。

```
// 建物クラス（スーパークラス）
class Building
{
    public int floors; // 何階建
    public 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
{
    public int families; // 世帯数

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

**課題2** 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を `System.out.println()`;メソッドで画面に表示できるようにしなさい。

**課題3** 前回、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群を作った。次に示す3つの要件を満たすようにそれらを変更しなさい。各メンバーのアクセス制限は各自で適切に設定しなさい。

要件1. サブクラスのデータ表示メソッドはスーパークラスのデータ表示メソッドをオーバーライドして作成すること。このとき、スーパークラスのデータ表示は `super.`を用いてスーパークラスのデータ表示メソッドを利用すること。

要件2. クラス `Mp3Player` のメンバーとして”Mp3Player”を文字列としてもつ `String` 型の定数を宣言すること。クラス `CellPhone` についても同様。

要件3. スーパークラス `Gadget` は `Object` クラスを継承している。サブクラス `Mp3Player` と `CellPhone` の各々において、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()`をオーバーライドすること。

**課題4** 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 `Mp3Plary` 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。オーバーライドしたデータ表示メソッドを用いて各オブジェクトのデータを順次出力しなさい。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。