

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;
    public A(int a){
        x=a;
    }
    public void show(){
        System.out.println(x);
    }
}
class B extends A{
    private int y;
    public B(int a, int b){
        super(a);
        y=b;
    }
    -----}{
        super.show();
        System.out.println(y);
    }
}
class Assignment7_1{
    public static void main(String[] args){
        A a=new B(10,20);
        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 2017",10000,"S 席");
        tk.show();
    }
}
```

〔実行結果〕

チケット名：矢沢永吉 2017
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

ソースファイル名: Assignment7_3.java (main() メソッドがあるクラス名と同じにします)

// 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

    // コンストラクタ
    public Building(int fl, int he){
        floors=fl;
        height=he;
    }
    // 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

    // コンストラクタ
    public Mansion(int fl, int he, int fa){
        super(fl, he);
        families=fa;
    }
    // 表示メソッド
    public void showMansion(){
        showBuilding();
        System.out.println("世帯数："+families);
    }
}
```

```
class Assignment7_3{
    public static void main(String[] args){
        Building bl=new Mansion(12,52, 68);
        bl.showBuilding();
    }
}
```

〔実行結果〕

12 階建
高さ：52m
世帯数：68

■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。

〔人を管理するクラス Person の例〕

```
// 人を管理するクラス
class Person{
    private String name;
    private int age;

    public Person(String n, int a){
        name=n;
        age=a;
    }

    // toString() をオーバーライドする
    public String toString(){
        String str=new String(name+"("+age+ ")");
        return str;
    }
}

class Assignment7_4{
    public static void main(String[] args)
    {
        Person p=new Person("福工大 太郎", 20);

        // オーバーライドされたメソッドが実行される
        System.out.println(p);
    }
}
```

〔実行結果〕

福工大 太郎(20)

■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 2 つの要件を満たすようにメンバーを変更/追加しなさい。

[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 GB
    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+"GB");
    }
}

// サブクラス Mobile
class Mobile extends Gadget
{
    private String number; // 番号
    public Mobile (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("[Mobile]");
        show();           // 継承されたメソッドを使用
        System.out.println("番号 : "+number);
    }
}
```

●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名 print()を変更しなさい。変更後の実行例を下に示します。

```
...
class Assignment7_5
{
    public static void main(String[] args){
        Gadget g=new Mp3Player("NW-A35", 20800, 16);
        g.show();
    }
}
```

〔実行結果〕

[MP3Player]
製品名：NW-A35、価格：20800 円
メモリ：16GB

●要件 2【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、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。

```
...
class Assignment7_5
{
    public static void main(String[] args){
        Mobile m=new Mobile("iPhone7Plus", 82300, "092-111-4321");
        System.out.println(m);
    }
}
```

〔実行結果〕

iPhone7Plus/¥82300/#092-111-4321