ＪａｖａプログラミングⅡ

６回目　継　　承　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)新しいクラスを宣言するとき既存のクラスを利用することはできない
(2)新しいクラスが既存のクラスのメンバーを受け継ぐことを継承という
(3)クラスの拡張における既存のクラスをサブクラスという
(4)サブクラスからスーパークラスの private メンバーをアクセスすることはできない
(5)クラスの拡張はキーワード extends により指定する
(6)super();はコンストラクタのどこに書いても良い
(7)引数なしの super();によりスーパークラスの引数なしのコンストラクタが実行される
(8)スーパークラスの protected メンバーはサブクラスからアクセスが可能である

■難易度★☆☆

課題 1 **次は文房具全般を表す文房具クラスです。この文房具クラス（スーパークラス）を拡張して、具体的な文房具を表すクラス（サブクラス）を宣言しなさい。例えば、ボールペンを表すクラスは、メンバーに色や線の太さを加えて以下のように宣言すればよいでしょう。追加するメンバーは各自にお任せします。**

〔文房具クラス（スーパークラス）とボールペンクラス（サブクラスの例)〕

```
// 文房具クラス（スーパークラス）
class Stationery
{
   public String whattodo; // 機能

   public void showStationery(){
      System.out.println("機能:"+whattodo);
   }
}

// ボールペンクラス（サブクラス）
class BallpointPen extends Stationery
{
   public String color;    // 色
   public double size;     // 太さ(mm)

   public void showBallpointPen(){
      System.out.println("【ボールペン】");
      showStationery();
      System.out.println("色："+color);
      System.out.println("太さ(mm):"+size);
   }
}
```

この他、具体的な文房具を表すクラスをここに宣言してください
たとえば、ノート（Notebook）や修正液（Whiteout）などがありますね

次に、メインメソッドから各自のサブクラスのオブジェクトを作成して動作を確認しなさい。
ソースファイル名：Assignment6_1.java（main()メソッドがあるクラス名と同じにします）

```
// ここへ文房具クラスとボールペンクラス、各自のクラスの宣言を書きましょう

class Assignment6_1
{
  public static void main(String[] args){
    // 作成したサブクラスの動作を確認（ボールペンクラスの場合）
    BallpointPen mypen=new BallpointPen();
    mypen.whattodo="字を書く";
    mypen.color="青";
    mypen.size=0.7;
    mypen.showBallpointPen();

    各自のサブクラスの動作を確認するコードをここに記述してください

  }
}
```

〔実行例〕

```
【ボールペン】
機能:字を書く
色：青
太さ(mm):0.7
  :
  :
```

（以降、各自のサブクラスの画面出力が続きます）

■難易度★☆☆

課題 2 次は建物全般を表す建物クラスです。この建物クラス（スーパークラス）を拡張して、具体的な建物を表すクラス（サブクラス）を宣言しなさい。例えば、マンションを表すクラスは、メンバーに世帯数を加えて以下のように宣言すればよいでしょう。追加するメンバーは各自にお任せします。

〔建物クラス（スーパークラス）とマンションクラス（サブクラスの例）〕

```
// 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

   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

   public void showMansion(){
      showBuilding();
      System.out.println("世帯数："+families);
   }
}
```

この他、具体的な建物を表すクラスをここに宣言してください
たとえば、デパート（Department）や大学（University）などがありますね

次に、メインメソッドから各自のサブクラスのオブジェクトを作成して動作を確認しなさい。

ソースファイル名：Assignment6_2.java（main()メソッドがあるクラス名と同じにします）

```
// ここへ建物クラスとマンションクラス、各自のクラスの宣言を書きましょう

class Assignment6_2
{
   public static void main(String[] args){
      // 作成したサブクラスの動作を確認（マンションクラスの場合）
      Mansion mymansion=new Mansion();
      mymansion.floors=10;
      mymansion.height=20;
      mymansion.families=50;
      mymansion.showMansion();
```

各自のサブクラスの動作を確認するコードをここに記述してください

```
   }
}
```

〔実行例〕

```
10 階建
高さ：20m
世帯数：50
   :
   :
（以降、各自のサブクラスの画面出力が続きます）
```

■難易度★★☆

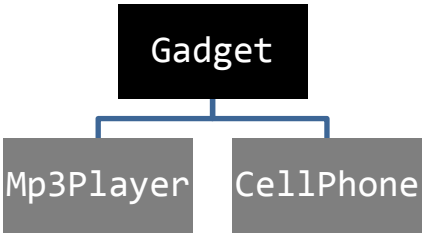
課題 3 次は物体全般を表す物体クラスです。この物体クラス（スーパークラス）を拡張し、具体的な物体を表すクラス（サブクラス）を宣言しなさい。例えば、机、車、テレビを表すクラスなどがあります。サブクラスではフィールドを初期設定するためのコンストラクタも適切に宣言しなさい。さらに、メインメソッドからサブクラスのオブジェクトを作成して動作を確認しなさい。

〔物体クラス（スーパークラス）の宣言〕

```
// 物体クラス
class Objects
{
   public String name;    // オブジェクトの名前
   public double[] size;  // サイズ縦、横、奥行(mm)
   public double weight;  // 重さ(g)

   public Objects(){
      // フィールドの初期設定
      name="No name";
      size=new double[3];
      for(int i=0;i<size.length;i++)
         size[i]=0.0;
      weight=0.0;
   }

   public Objects(String n, double[] s, double w){
      // 与えられたデータをフィールドに設定
      name=n;
      size=s;
      weight=w;
   }

   public void show(){
      // 各フィールドを表示
      System.out.println("Name:"+name);
      System.out.println("size:H"+size[0]+",W"+size[1]+",D"+size[2]+"(mm)");
      System.out.println("Weight:"+weight+"(g)");
   }
}
```

■難易度★★★

課題 4 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を作ります。そこで次のようなクラス階層を考えます。これらのクラスを宣言しなさい。



〔スーパークラス Gadget のメンバー〕

private フィールド：	製品名（String）、価格（int）
public メソッド：	void show(); データ（製品名と価格）の表示
public コンストラクタ：	Gadget(String, int); 製品名と価格の設定

〔サブクラス Mp3Player のメンバー〕

private フィールド：	容量 MB（int）
public メソッド：	void print(); データ表示（製品名と価格、容量）の表示 ※製品名と価格は継承された show()メソッドを用いて表示
public コンストラクタ：	Mp3Player(String, int, int); 製品名と価格、容量の設定 ※製品名と価格は super()を用いて設定

〔サブクラス CellPhone のメンバー〕

private フィールド：	電話番号（String）
public メソッド：	void print(); データ（製品名と価格、番号）の表示 ※製品名と価格は継承された show()メソッドを用いて表示
public コンストラクタ：	CellPhone(String, int, String); 製品名と価格、番号の設定 ※製品名と価格は super()を用いて設定

次に、メインメソッド内でクラス Mp3Player と CellPhone のオブジェクトを生成しなさい。また各オブジェクトのデータを出力しなさい。

ソースファイル名：Assignment6_4.java（main()メソッドがあるクラス名と同じにします）

```
// ここへ Gadget クラスと Mp3Player クラス、CellPhone クラスの宣言を書きましょう

class Assignment6_4
{
    public static void main(String[] args){
        Mp3Player myplayer = new Mp3Player("Creative Zen nano",9980,1024);
        CellPhone myphone = new CellPhone("Nokia 6280",37000,"09012345678");
        myplayer.print();
        myphone.print();
    }
}
```

■難易度★☆☆

課題 5 次は仕事する人全般を表わす Worker クラスです。この Worker クラス（スーパークラス）を拡張して、具体的な仕事する人を表わすクラス（サブクラス）を宣言しなさい。例えば、医者を表すクラスは、メンバーに患者数を加えて以下の例のように宣言すればよいでしょう。追加するメンバーは各自にお任せします。

〔Worker クラス（スーパークラス）と医者クラス（サブクラスの例）〕

```
// Worker クラス（スーパークラス）
class Worker
{
   public String name;     // 名前
   public int age;         // 年齢

   public void showWorker(){
      System.out.println("名前:"+name);
      System.out.println("年齢:"+age);
   }
}

// 医者クラス（サブクラス）
class Doctor extends Worker
{
   public int patients; // 患者数

   public void showDoctor( ){
      System.out.println("■医者■");
      showWorker();
      System.out.println("患者数:"+patients);
   }
}
```

この他、具体的な仕事を行う人を表すクラスをここに宣言してください
たとえば、ドライバ（Driver）や先生（Teacher）などがあります

次に、メインメソッドから各自のサブクラスのオブジェクトを作成して動作を確認しなさい。
ソースファイル名：Assignment6_5.java（main()メソッドがあるクラス名と同じにします）

```
// ここへ Worker クラスと医者クラス、各自のクラスの宣言を書きましょう

// 医者の場合のコード例
class Assignment6_5
{
   public static void main(String[] args){
      Doctor doc=new Doctor();
      doc.name="福工大 太郎";
      doc.age=25;
      doc.patients=55;
      doc.showDoctor();
```

各自のサブクラスの動作を確認するコードをここに記述してください

```
   }
}
```

〔実行例〕

```
■医者■
名前:福工大 太郎
年齢:25
患者数:55
   :
   :
（以降、各自のサブクラスの画面出力が続きます）
```