

JavaプログラミングⅡ

6回目 継承 課題

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) 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を受け継ぐことを継承という
- (3) 継承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メンバをアクセスすることは一般にはできない
- (5) 継承はキーワード `extends` により指定する
- (6) コンストラクタはクラスのメンバではないため、継承されない
- (7) スーパークラスのコンストラクタは `super()` により呼び出しが可能である
- (8) スーパークラスの `protected` メンバ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）を継承して、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。追加するメンバは各自にお任せします。次に、例のよう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一致させる)

```
// 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

```
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

```
}
}
```

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)を継承して、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。追加するメンバは各自にお任せします。次に、例のよう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一致させる)

```
// 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

```
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

```
    }
}
```

難易度★★☆

課題3 次は物体全般を表す物体クラスです。この物体クラスを継承し、具体的な物体を表す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5 次は仕事するヒトを表わすクラス Worker である。この Worker クラスを継承し、具体的な仕事を行うヒトを表わすサブクラスを宣言しなさい。例えば、医者、ドライバ、先生などがあります。医者であれば、患者数などの変数と、表示するメソッド void showall(); を以下の例のように宣言すればよいでしょう。次に、例のよう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。

```
// 仕事をする人を表わすクラス（スーパークラス）
```

```
class Worker
{
    public String name; // 名前
    public int age;     // 年齢

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

```
// 医者を表わすクラス（サブクラス）
```

```
class Doctor extends Worker
{
    public int patients; // 患者数

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

この他、具体的な仕事を行うヒト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

```
// 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.showall();

        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
    }
}
```