

## JavaプログラミングII

### 4回目 クラスの機能(2) コンストラクタ、クラス変数、クラスメソッド 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) コンストラクタはメソッドと同様に戻り値をもつ
- (2) コンストラクタはオブジェクトが生成されると最初に実行される
- (3) コンストラクタはメソッドと同様にオーバーロードができる
- (4) コンストラクタは常に public メンバとしなければならない
- (5) クラス変数はクラス単位で準備される変数である
- (6) クラスメソッド内でインスタンス変数を参照してもよい
- (7) クラスメソッドを実行する場合はクラス名を指定して行う
- (8) クラス変数とクラスメソッドの宣言は static 修飾子で行う

#### ■難易度★☆☆

**課題 1** コンストラクタの機能を用いて、次に示すクラス Triangle のフィールドを与えられた値で初期設定できるようにしなさい。下にコードの実行結果を示します。

ソースファイル名: Assignment4\_1.java (main()メソッドがあるクラス名と同じにします)

```
class Triangle
{
    private double height;    // 高さ
    private double base;     // 底辺

    

ここにコンストラクタを追加します



    public void setHeight(double h){ // 高さの設定
        height = h;
    }
    public void setBase(double b){   // 底辺の設定
        base = b;
    }
    public double getArea(){        // 面積の計算
        return(height*base/2.0);
    }
}
class Assignment4_1
{
    public static void main(String[] args){
        // このように初期設定がしたい
        Triangle tr;
        tr = new Triangle(10.5, 2.5);
        System.out.println("三角形の面積は"+tr.getArea()+"です");
    }
}
```

〔実行結果〕

三角形の面積は 13.125 です

■難易度☆☆☆

**課題 2** コンストラクタの機能を用いて、次に示すクラス ConstructorTest のフィールド：

```
private int i;  
private double d;  
private String s;
```

を与えられた値で初期設定できるようにしなさい。下にコードの実行結果を示します。

ソースファイル名: Assignment4\_2.java (main()メソッドがあるクラス名と同じにします)

```
class ConstructorTest  
{  
    private int i;  
    private double d;  
    private String s;  
  


ここにコンストラクタを追加します

  
    public void setValue(int n){  
        i=n;  
    }  
    public void setValue(double n){  
        d=n;  
    }  
    public void setValue(String n){  
        s=n;  
    }  
    public void dispValue(){  
        System.out.println("int="+i+", double="+d+", String="+s);  
    }  
}  
class Assignment4_2  
{  
    public static void main(String[] args){  
        // このように初期設定がしたい!!  
        ConstructorTest ct;  
        ct=new ConstructorTest(10,12.5,"Constructor");  
        ct.dispValue();  
    }  
}
```

〔実行結果〕

---

int=10, double=12.5, String=Constructor

■難易度★★☆

**課題 3** クラス変数とクラスメソッドを用いることにより簡単なライブラリを作成できます。次に示すクラス変数とクラスメソッドをもつクラス MyMath を宣言しなさい。

〔クラス MyMath のメンバー〕

---

クラス変数：

```
private フィールド double PI=3.141592;
```

クラスメソッド：

```
public メソッド // ラジアン radian を度へ変換 (参考 degree = radian * 180.0 / PI)
double transRad2Deg(double radian);
// 度 degree をラジアンへ変換 (参考 radian = degree * PI / 180.0)
double transDeg2Rad(double degree);
// 半径 radius の円の面積を計算 (参考 area = PI * radius * radius)
double getAreaOfCircle(double radius);
```

■難易度★★☆

**課題 4** クラス MyMath の動作をチェックするために、メインメソッドからクラス MyMath のクラスメソッドを実行するコードを記述しなさい。

■難易度★★★

**課題 5** 次は書籍を管理するクラス Book のメンバーの一覧です。インスタンス変数/メソッドにより個々の書籍の情報を保持し、クラス変数/メソッドにより書籍の総数を保持するようにします。クラス Book を宣言しなさい。

〔クラス Book のメンバー〕

---

クラス変数/クラスメソッド：

```
private フィールド int num_of_books; // 書籍の総数
public メソッド void stat_of_books(); // 書籍の総数を出力
```

インスタンス変数/インスタンスメソッド：

```
private フィールド
String name; // 書籍名
int price; // 価格
public メソッド
void infoBook(); // 書籍名と価格を出力
```

コンストラクタ：

```
public
Book(); // コンストラクタ 1
// 書籍名"No title"、価格は-1で初期化します
// 書籍の総数を1増やします
Book(Sting, int); // コンストラクタ 2
// 与えられた書籍名と価格で初期化します
// 書籍の総数を1増やします
```

■難易度★★☆

**課題6** クラス Book の動作をチェックするために、メインメソッドからクラス Book のオブジェクトを複数個生成し、各書籍の情報と総数を出力するコードを記述しなさい。

〔実行例〕

---

[[総冊数 0 冊]]

書籍名:No title

価 格:不明

書籍名:Java Programming

価 格:3900 円

書籍名:簡単 C++

価 格:1980 円

[[総冊数 3 冊]]

■難易度★★☆

**課題7** コンストラクタの機能を用いて、次に示すクラス Person のフィールド：

```
private String name;
```

を与えられた値で初期設定できるようにしなさい。下に実行結果を示します。

ソースファイル名:Assignment4\_7.java (main()メソッドがあるクラス名と同じにします)

---

```
class Person
```

```
{
```

```
    private String name;
```

ここにコンストラクタを  
オーバーロードして追加します

```
    public void setName(String s){
```

```
        name=s;
```

```
    }
```

```
    public String getName(){
```

```
        return name;
```

```
    }
```

```
}
```

```
class Assignment4_7
```

```
{
```

```
    public static void main(String[] args){
```

```
        Person p1 = new Person();
```

```
        Person p2 = new Person("太郎");
```

```
        System.out.println("p1="+p1.getName());
```

```
        System.out.println("p2="+p2.getName());
```

```
    }
```

```
}
```

〔実行結果〕

---

p1=名無し

p2=太郎

■難易度★★☆

**課題 8** 次に示すディスプレイクラスを宣言しなさい。また、下に示すようにクラス Display を使用するコードを実行して動作を確認しなさい。

〔クラス Display のメンバー〕

---

```
private フィールド：
    String name;           // 商品名                ※初期値 "未設定"
    int width;             // 解像度：横サイズ（ピクセル） ※初期値 0
    int height;           // 解像度：縦サイズ（ピクセル） ※初期値 0
    boolean speaker;      // スピーカ付き?                ※初期値 false
public コンストラクタ：
    // フィールドに初期値を代入します
    Display();
    // 商品名を設定し、それ以外は初期値を代入します
    Display(String s);
    // 商品名と解像度を設定し、それ以外は初期値を代入します
    Display(String s, int w, int h);
    // 商品名と解像度、スピーカの有無を設定します
    Display(String s, int w, int h, boolean sp);
public メソッド：
    // ディスプレイの情報を戻します
    String getDisplay();
```

ソースファイル名: Assignment4\_8.java (main()メソッドがあるクラス名と同じにします)

---

```
// (ここにクラス Display の宣言を記述します)
class Display{
    ...
}

class Assignment4_8{
    public static void main(String[] args){
        Display[] ds = new Display[3];
        ds[0] = new Display("Dell P2815Q Ultra HD(4K)", 3840, 2160, true);
        ds[1] = new Display("Iiyama ProLite E2471HSD", 1920, 1080);
        ds[2] = new Display("LG 29UB65-P");

        for(int i=0;i<ds.length;i++)
            System.out.println(ds[i].getDisplay());
    }
}
```

〔実行例〕

---

```
Dell P2815Q Ultra HD(4K)(3840x2160)※スピーカ付き
Iiyama ProLite E2471HSD(1920x1080)※スピーカ無し
LG 29UB65-P(0x0)※スピーカ無し
```