

JavaプログラミングII

3回目 クラスの機能（1）アクセス制限、オーバーロード 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) クラスの `private` メンバは、そのクラスからのみアクセス可能なメンバである
- (2) 一般に、クラスのフィールドはどこからでもアクセスできるように `public` メンバで宣言すべきである
- (3) クラスは `private` メンバと `public` メンバを同時に持つことはできない
- (4) `public` メンバはどこからでもアクセスできるメンバであるため、宣言する際には注意が必要である
- (5) クラスのカプセル化は、クラスの内部のデータを外部から隠ぺいして、データの保護や保守に大きく貢献する。
- (6) クラスのカプセル化は、ポリモーフィズムと呼ばれる。
- (7) オーバーロードとは、戻り値の型が異なる同じ名前と仮引数のメソッドを複数宣言することである。
- (8) オーバーロードされたメソッドを呼び出すとき、与えられた実引数の違いにより実際に呼び出されるメソッドが決定される。

■難易度☆☆☆

課題1 クラス `Score` は得点を管理するクラスです。変数 `point` には `0` 以上 `100` 以下の値のみ代入されるようにします。空欄を埋めてクラスをカプセル化してコードを完成させなさい。ソースファイル名: `Assignment3_1.java` (`main()`メソッドがあるクラス名と同じにします)

```
class Score
{
     int point; // 得点

    // 得点設定メソッド
     void setPoint(int p){ // 値を検査して 0~100 であれば代入します
        
    }
    // 現設定値取得メソッド
     int getPoint(){ // 現在の値を戻り値として返します
        
    }
}
class Assignment3_1
{
    public static void main(String[] args){
        Score sc=new Score();
        sc.setPoint(90); // 90 点を設定します
        System.out.println(sc.getPoint()); // 得点を表示します
        sc.setPoint(-10); // -10 点を設定してみます
        System.out.println(sc.getPoint()); //得点を表示します
    }
}
```

■難易度★☆☆

課題 2 クラス EvenInt は偶数を管理するクラスです。変数 num には偶数のみ代入されるようにします。空欄を埋めてクラスをカプセル化してコードを完成させなさい。

ソースファイル名: Assignment3_2.java (main()メソッドがあるクラス名と同じにします)

```
class EvenInt
{
     int num; // 偶数

    // 偶数設定メソッド
     void setInt(int i){ // 値を検査して偶数であれば代入します
        
    }
    // 現設定値取得メソッド
     int getInt(){ // 現在の値を戻り値として返します
        
    }
}
class Assignment3_2
{
    public static void main(String[] args){
        EvenInt ei=new EvenInt();
        ei.setInt(12);    // 偶数を設定します
        System.out.println(ei.getInt());    // 結果を表示します
        ei.setInt(15);    // 奇数を設定してみます
        System.out.println(ei.getInt());    // 結果を表示します
    }
}
```

■難易度★★☆

課題 3 メソッドのオーバーロードの機能を用いて、次に示すクラス OverLoadTest 内のメソッドを一つにまとめなさい。

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

    public void setInt(int n){
        i=n;
    }
    public void setDouble(double n){
        d=n;
    }
    public void setString(String n){
        s=n;
    }

    public void dispValue(){
        System.out.println("int="+i+", double="+d+", String="+s);
    }
}
```

```
class Assignment3_3
{
    public static void main(String[] args){
        OverLoadTest olt=new OverLoadTest();

        //olt.setInt(10);
        //olt.setDouble(12.5);
        //olt.setString("Hello");
        //olt.dispValue();

        // 次のようにアクセスしたい！！
        olt.setValue(10);
        olt.setValue(12.5);
        olt.setValue("OverLoad");
        olt.dispValue();
    }
}
```

■難易度★★☆

課題 4 個人データを管理するクラス Person を次のように宣言しなさい。このとき、メソッド内では入力値が正しいかどうかのチェックを行った後、private フィールドに値を設定するようにすること。

〔クラス Person のメンバー〕

private フィールド：

```
String name; // 名前
int age;     // 年齢 (0 以上)
char gender; // 性別 (F or M)
```

public メソッド：

```
void setName(String); // 名前を設定
String getName();     // 名前を取得
void setAge(int);     // 年齢を設定
int getAge();         // 年齢を取得
void setGender(char); // 性別を設定
char getGender();     // 性別を取得
```

■難易度★★☆

課題 5 クラス Person の動作をチェックします。メイン関数からクラス Person のオブジェクトを生成して、オブジェクトに値を設定するコードを記述しなさい。また、設定した値を取得して出力しなさい。

■難易度★★☆

課題 6 メソッドのオーバーロードの機能を用いて、クラス Person の下の 3 つのメソッドを 1 つのメソッド setData() で実現しなさい。

```
void setName(Sting)
void setAge(int)
void setGender(char)
```

■難易度★★☆

課題 7 次に示す USB メモリクラスを宣言しなさい。また、下に示すようにクラス USB 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 USB のメンバー〕

private フィールド：

```
int capacity; // 容量 (GB)
```

public メソッド：

```
void setCap(int); // 容量の設定 (入力値が 0 以上の場合のみ設定します)
```

```
int getCap(); // 容量の取得
```

ソースファイル名: Assignment3_7.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 USB の宣言を記述します)
```

```
class USB
```

```
{
```

```
    ...
```

```
}
```

```
class Assignment3_7
```

```
{
```

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

```
        USB ToteBag = new USB();
```

```
        // 容量データの設定と表示
```

```
        ToteBag.setCap(16);
```

```
        System.out.println("USB メモリ("+ToteBag.getCap()+"GB");
```

```
        // 容量データの設定と表示 (異常な値を設定)
```

```
        ToteBag.setCap(-32);
```

```
        System.out.println("USB メモリ("+ToteBag.getCap()+"GB");
```

```
    }
```

```
}
```

〔実行例〕

容量を 16GB に設定しました。

USB メモリ(16GB)

不正な容量値(-32GB)です。

USB メモリ(16GB)

■難易度★★☆

課題 8 次に示す充電電池クラスを宣言しなさい。また、下に示すようにクラス Battery 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 Battery のメンバー〕

```
private フィールド：
    int level;           // 充電レベル% (0~100)
public メソッド：
    void setLevel(int); // 充電レベルの設定
                        // ※入力が 0 以上 100 以下の場合のみ設定します
    int getLevel();     // 充電レベルの取得
```

ソースファイル名: Assignment3_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 Battery の宣言を記述します)
class Battery
{
    ...
}

class Assignment3_8
{
    public static void main(String[] args){
        Battery evolta = new Battery();

        // 充電レベルの設定と表示
        evolta.setLevel(80);
        System.out.println("現在の充電レベルは"+evolta.getLevel()+"%です。");

        // 過充電レベルの設定 (異常な値を設定)
        evolta.setLevel(102);
        System.out.println("現在の充電レベルは"+evolta.getLevel()+"%です。");
    }
}
```

〔実行例〕

```
[充電レベルを 80%に設定しました]
現在の充電レベルは 80%です。
[異常な充電レベル(102%)です]
[充電レベルを設定できませんでした]
現在の充電レベルは 80%です。
```