

Java 練習問題. 06 クラスとメソッド

設問1. クラスについての理解

典型的なクラスは次のような構成をしています。

```
[修飾子] class <クラス名> {  
  
    // クラス変数 (フィールド)  
    [修飾子] <型> <変数名> = 値;  
  
    // コンストラクタ  
    [アクセス修飾子] <クラス名>([引数]) {  
  
    }  
  
    // メソッド  
    [修飾子] <戻り値型> <メソッド名>([メソッド引数]) {  
  
    }  
}
```

- (1) クラスを構成する要素は何か。

解答：フィールドとメソッド

※コンストラクタはメソッドの一種です

new 演算子でインスタンス生成した際に自動で呼び出されます

- (2) 上記の要素のうち、処理（プログラム文）を記述するのはどの要素の中でなければならないか。

解答：メソッドとコンストラクタ

※コンストラクタはインスタンス生成時に行う処理を、メソッドはインスタンスの振る舞い/動作の処理を記述します

設問2. メソッドの作成についての理解

- (1) メソッドの<戻り値型>のところが void となっているとき、そのメソッドはどのようなメソッドか。

また、そのメソッドの最後に記述する必要がないキーワードは何か。

解答：メソッドの戻り値が無い、return

※戻り値が無い場合、「return」は省略可能

- (2) メソッドの最後に『return 戻り値;』の記述がある場合、この戻り値はどのような型でなければならないか。

解答：メソッド定義の戻り値型と同じ型

- (3) 以下のメソッドの間違いを説明しなさい。

```
public void sampleMethod(int a) {  
    String str = "数字は" + a ;  
    return str;  
}
```

解答：void メソッドのため return で戻り値を返すことができない

もしくは

return で戻り値を返す場合は、戻り値型(この場合は String)を定義する必要がある

- (4) 以下のメソッドにおいて、return の後に記述する変数名は以下の中でどれが適切か。

```
public SampleBean sampleMethod(int a) {  
    SampleBean sb = new SampleBean();  
    String str = "Hello";  
    return ____;  
}
```

解答：sb

※メソッドの定義「public SampleBean sampleMethod(int a)」より、このメソッドの戻り値型は『SampleBean』。

メソッド内において SampleBean 型の変数は sb であるため。

設問3. メソッドの利用（呼び出し）と処理の流れについての理解

基本的なメソッドの利用（呼び出し）方法は、次のようになります。メソッドを呼び出すと処理は呼び出したメソッド内に移ります。そして呼ばれたメソッド内の処理が順に実行され最後まで達すると、呼び出したメソッドに戻ります。

```
// 自クラスの戻り値のないメソッドを呼び出す場合
```

```
メソッド名(値 1, 値 2, ....);
```

```
もしくは
```

```
this.メソッド名(値 1, 値 2, ....);
```

```
// 自クラスの戻り値のあるメソッドを呼び出す場合
```

```
<戻り値型> 変数 = メソッド名(値 1, 値 2, ....);
```

```
もしくは
```

```
<戻り値型> 変数 = this.メソッド名(値 1, 値 2, ....);
```

```
// 他クラスの戻り値のないメソッドを呼び出す場合
```

```
<クラス型>変数.メソッド名(値 1, 値 2, ....);
```

```
// 他クラスの戻り値のあるメソッドを呼び出す場合
```

```
<戻り値型> 変数 = <クラス型>変数.メソッド名(値 1, 値 2, ....);
```

※引数を受け取らないメソッドの場合、メソッド名の後ろの括弧の中(引数)は空欄

- (1) 下記のメソッドの処理Aと処理B間に、クラス内の betushori メソッドを呼び出す文を記入しなさい。

```
class TestClass {  
    public void callMethod() {  
        処理A  
        betushori(); もしくは this.betushori();  
        処理B  
    }  
  
    private void betushori() {  
        処理C  
        処理D  
    }  
}
```

- (2) 上記のプログラムにおいて、処理A～Dが処理される順番に並べなさい。

1. A 2. C 3. D 4. B

(3) 下のプログラムの結果として、何が表示されるか答えなさい。

```
public void shori() {  
    int c = 0;  
    String str = "おはよう";  
    str = change(c);  
    System.out.println(str);  
}  
  
private String change(int a) {  
    String s = " ";  
    if (a == 1) {  
        s = "こんにちは";  
    }  
    return s;  
}
```

解答：空白

(4) 引数を受け取るメソッドを使用するときに、引数部分を記入する際に気をつけなければならない点を挙げなさい。

引数の型、引数の数、引数の順序

(5) 以下のプログラムを実行したとき、何が表示されるか答えなさい。

```
public class Main {  
  
    public static void main(String[] args) {  
  
        int[] array = { 28, 10, 93, 8, 45 };  
        int x      = 5;  
        int num     = 0;  
  
        num = array[3];  
  
        Jouken j = new Jouken();  
        num = j.getNumber(num);  
  
        Loop l = new Loop();  
        num = l.kurikaesi(num, x);  
        num = num + 4;  
        l.show(num);  
    }  
}
```

```
public class Jouken {  
  
    public int getNumber(int num) {  
  
        String str = "";  
        int x      = 0;  
        x          = 2;  
  
        switch(x) {  
        case 1:  
            num = 30;  
            break;  
        case 2:  
            num = 40;  
            break;  
        case 3:  
            num = 50;  
            break;  
        }  
  
        return x;  
    }  
}
```

```
public class Loop {  
    public void show(int num) {  
        System.out.println(num);  
    }  
    public int kurikaesi(int num, int i) {  
        int[] array = new int[i];  
        for (int c = 0; c < i; c++) {  
            array[c] = c * 3;  
        }  
        num = array[2];  
        return num;  
    }  
}
```

表示結果： 10