

# Java 基本練習問題

## ～4. Java検定対策～

(1) 次の画面の画面を参考にプログラムを書きなさい。

```
C:¥yy>java Za
文字を入れて下さい。

C:¥yy>java Za abcd
abcdの文字数は4です

C:¥yy>java Za #####
#####の文字数は12です
```

※実行確認のときに、java ファイル名 ×× とすると、××をプログラム中のargs[ ]内に代入して処理させることができる。上の画面例だと、abcdefgという文字列がargs[0]～args[6]までに1文字ずつ代入され、実行される。

このように、実行時にプログラムに渡す値のことを、引数(ひきすう)という。

### ファイル名: Za.java

```
class Za{
    public static void main(String args[]){
        int i=0;
        if(args.length==0){
            System.out.println("文字を入れて下さい。");
            return;
        }
        while(i<args.length){
            System.out.println(args[i]+"の文字数は"+args[i].length()+"です");
            i++;
        }
    }
}
```

#### 【ワンポイント】while(条件式){～}

条件を満たす間、{から}までの処理を繰り返す。繰り返しから抜けるときは、return;を使う。

#### 【ワンポイント】length

配列変数の個数を調べるときには、配列名.length となる。

if(args.length==0){ ...もし、argsという配列の個数が0ならば、...という意味。

#### 【ワンポイント】キャスト化

プログラムZd.javaでは、r=(int)Math.round(Math.random()\*9);という記述がある。

これは、Math.round()というメソッドがfloat型の数値しか受け入れられないのだが、rはint型で宣言している。そこで、Math.round()メソッドをint型に変換しなければならない。

そのような場合に、使用しながら変数の型を変換するときには()で囲い、その前に記述する。

このような変換をキャスト化という。

ちなみに、r=(int)Math.round(Math.random()\*9);は、「0～9までの乱数をrに代入せよ」という意味。Math.random()メソッドは、import java.util.\*;を宣言しなくとも使える乱数。0～1までのデタラメな数を発生させる。Math.round()メソッドは、()内の数値を四捨五入する。

つまり、この場合は0～1までの乱数を9倍し、それを四捨五入することによって0～9までの乱数を発生させている。四捨五入なので9まで乱数に含まれる。

(2) 次の画面を参考にプログラムを入力し、実行結果を確認しなさい。

```
C:\¥yy>java Zb a 3
1:a
2:aa
3:aaa

C:\¥yy>java Zb # 2
1:#
2:##

C:\¥yy>java Zb
文字を2つ入れて下さい。
```

ファイル名: Zb.java

```
class Zb{
    public static void main(String args[]){
        int a;
        if(args.length!=2){
            System.out.println("文字を2つ入れて下さい。");
        }else{
            a=Integer.parseInt(args[1]);
            display(args[0], a);
        }
    }
    static void display(String str, int a){
        for(int i=1;i<=a;i++){
            System.out.print(i+":");
            for(int j=0;j<i;j++){
                System.out.print(str);
            }
            System.out.println();
        }
    }
}
```

#### 【ワンポイント】static void

関数といってdisplay()をひとつの命令として宣言できる。最初に使われたdisplayという命令の中身は、static void内で処理されている。Integer.parseInt()、は文字列を数値化するメソッド。

#### 【ワンポイント】比較演算子

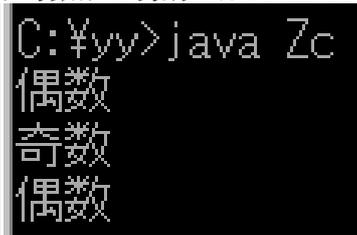
!= は、そうでないなら、という意味。

```
if(args.lenthg != 2) { ...
```

では、args.lengthの値(代入された文字)が2つなかったら、...せよ、という意味になる。

(比較演算子)	(意味)
A<=B	BはA以上である
A>=B	AはB以上である
A>B	AはBよりも大きい
A<B	BはAよりも大きい
A==B	AとBは等しい
A!=B	AとBは等しくない

(3) 配列変数moji[]を設定し、それに30,51,8を代入し、それらが偶数か奇数かを調べるプログラムを作りなさい。ただし、偶数か奇数かの判断には、Numberという関数を作成すること。



**ファイル名: Kc.java**

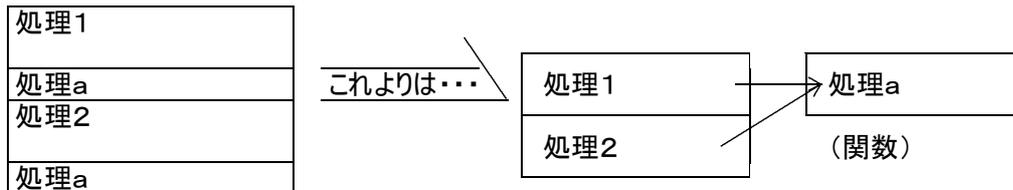
```
class Zc{
    public static void main(String args[]){
        String moji[ ]={"30","51","8"};
        for(int i=0 ; i<moji.length ;i++){
            int num=Integer.parseInt(moji[i]);
            System.out.println(Number(num));
        }
    }
    static String Number(int n){
        int amari=n%2;
        if(amari!=1){return "偶数";}
        else{return "奇数";}
    }
}
```

**【ワンポイント】関数**

プログラム上で、何度も同じ処理をする場合には、実行するたびにその部分を書くよりは、呼び出すほうが効率的である。

上のプログラムでいうと、Number{~}という部分がSystem.out.println(Number(num));という部分で何度も呼び出されている。

このように、同じ処理を一つにまとめ名前をつけ、あたかも一つの命令のようにまとめることを関数化という。



**【ワンポイント】演算子**

JavaではC言語とよく似た演算子が用意されている。上記のプログラムの%は、割った余り、という意味だ。

(演算子)	(演算内容)	
+	足し算	
-	引き算	
*	掛け算	
/	割り算	
%	割り算の余りを求める	(例) 10%3=1
+=	和の代入	(例) a=a+b; と a+=b; は同じ
-=	差の代入	
*=	積の代入	
/=	商の代入	
%=	割り算の余りを代入	
++	1加える	(例) a=a+1; と a++; は同じ
--	1減らす	

(4) 次の画面のように、0~9までの乱数を5×10個表示させるプログラムを書きなさい。

```
C:\¥yy>java Zd
4      8      6      5      4
4      4      4      8      9
4      6      1      3      4
3      3      7      9      7
2      6      3      0      2
2      5      8      0      4
4      4      2      7      0
1      7      7      1      7
5      7      4      6      5
7      4      3      7      5
```

**ファイル名: Zd.java**

```
class Zd{
    public static void main(String args[]){
        int r, i, j;
        for(j=1; j<=10; j++){
            for(i=1; i<=5; i++){
                r=(int)Math.round(Math.random()*9);
                System.out.print(r+"¥t");
            }
            System.out.print("¥n");
        }
    }
}
```

**【ワンポイント】エスケープシーケンス**

Javaで扱える文字には特殊な文字も含まれる。改行やタブなどもSystem.out.print()メソッドで表記できる。そのような文字をエスケープシーケンスといい、次のような文字がある。

- ¥b      バックスペース
- ¥t      タブ
- ¥n      改行
- ¥f      カーソルを1段下に移動
- ¥r      カーソルを行頭に戻す
- ¥       " "
- ¥¥     ¥

**【ワンポイント】print () とprintln ()**

System.out.println(""); で書かれた文字や数字は改行される。それに対してSystem.out.print(""); では改行がされない。従って次のような場合では1行に書かれることになる。

```
System.out.print("a");
System.out.print("b");
System.out.print("c");
実行結果      abc
```

```
System.out.println("a");
System.out.println("b");
System.out.println("c");
実行結果      a
                  b
                  c
```