

0 から始める Java 入門 ～5.引数を利用する～

(1) 引数の回数だけ「おはよう」と繰り返し表示するプログラムを書きなさい。



ファイル名: Ha.java

```
class Ha{
    public static void main(String args[]){
        int i,j;
        j=Integer.parseInt(args[0]);
        for(i=0;i<j;i++){
            System.out.println("おはよう");
        }
    }
}
```

【ワンポイント】引数の利用

Java では上記のように、クラス名のあとに文字列や数字を入力し、プログラム内で利用することが可能です。

`args[0]`に渡されますが、利用の際には、文字列オブジェクトであることに留意します。

(`public static void main(String args[]`)
したがって、数値として利用したい場合には、次のようにして数値に変換します。

```
int i=Integer.parseInt(args[0]);
```

このようにすると引数として渡され `args[0]`に代入された値を数値として利用できるようになります。

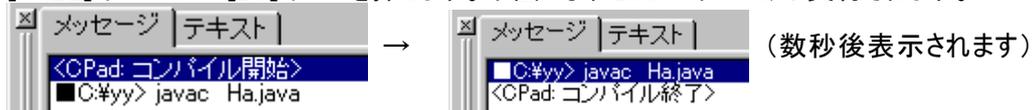
【ワンポイント】jcpadでの引数利用

jcpad では[F9]キーを押して、プログラムを[コンパイル+実行]します。しかし、その場合には引数を渡さず実行してしまいます。

したがって意図的に引数を渡して実行したい場合は、コンパイルと実行を別々に行います。

①コンパイルのみを実行

[Ctrl]キー + [F9]キー を押します。下図のようにコンパイルのみが実行されます。



②実行

コマンド欄に次の書式で入力し、[Enter]キーを押します。
java クラス名 引数



実行確認後は、`exit` と入力し、[Enter]キーを押すと、DOS 画面が消えます(手動で画面を消さなくてはなりません)。



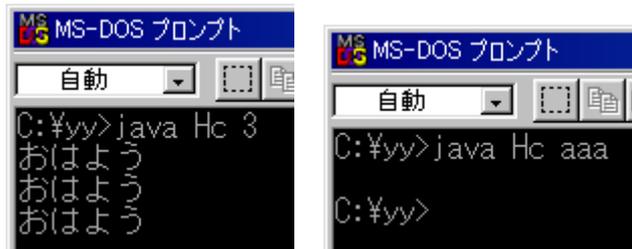
(2) 引数で渡した文字列を表示させるプログラムを作りなさい。



ファイル名: Hb.java

```
class Hb{
    public static void main(String args[]){
        System.out.println(args[0]);
    }
}
```

(3) (1)の問題で引数に文字列が渡されたとき、エラー表示させないようにするようプログラムを変更しなさい。



ファイル名: Hc.java

```
class Hc{
    public static void main(String args[]){
        int i,j;
        try{
            j=Integer.parseInt(args[0]);
            for(i=0;i<j;i++){
                System.out.println("おはよう");
            }
        } catch(Exception e){}
    }
}
```

【ワンポイント】例外処理

プログラム上、予期しない例外的な状況が発生した場合に、対応する処理を用意しておけるのが try{}catch() です。次のような文法になります。

```
try{ ① } catch(Exception e){ ② }
```

…①の処理をしているときに例外が発生したら、②を実行する。

上記の例では、引数に数字以外が渡されると Integer 型に変換できないためエラーになる。そこで、文字列が渡されたときは例外処理ということで try{}catch()を使う。

```

MS-DOS プロンプト
自動
C:\¥yy>java Hd 1
こんにちは

C:\¥yy>java Hd 2
こんばんわ

C:\¥yy>java Hd 0
おはよう

C:\¥yy>java Hd 99
さようなら

C:\¥yy>

```

(4) 引数を次のように渡すと、表示されるメッセージが変わるプログラムを組みなさい。

引数 0	おはよう
引数 1	こんにちは
引数 2	こんばんわ
それ以外	さようなら

ファイル名: Hd.java

```

class Hd{
    public static void main(String args[]){
        int i;
        try{
            i=Integer.parseInt(args[0]);
            switch(i){
                case 0:
                    System.out.println("おはよう");break;
                case 1:
                    System.out.println("こんにちは");break;
                case 2:
                    System.out.println("こんばんわ");break;
                default:
                    System.out.println("さようなら");break;
            }
        }
        catch(Exception e){}
    }
}

```

【ワンポイント】switch(変数){...}

条件分岐をさせるときに、0の時は...、1の時は...、2 のときは...というように規則性があるときなどは、if()を使うよりも、switch を使うほうが便利です

```

switch(変数) {
    case 0: ...①...;break;
    case 1: ...②...;break;
    case 2: ...③...;break;
    .
    .
    case n: ...④...;break;
    default:..⑤..;
}

```

変数が 0 のときは①、1 のときは②、、、もしそれ以外ならば⑤が実行される、、、という意味になります。

```
C:\¥yy>java He 0
こんにちは、 はなこさん。

C:\¥yy>java He 1
こんにちは、 たろうさん。

C:\¥yy>java He 2
こんにちは、 じろうさん。

C:\¥yy>java He 3
こんにちは、 あかねさん。
```

(5) 引数を次のように渡すと、「こんにちは、〇〇さん」と表示されるメッセージが変わるプログラムを組みなさい。

引数	0	はなこ
引数	1	たろう
引数	2	じろう
引数	3	はなこ

ファイル名: He.java

```
class He{
    public static void main(String args[]){
        int i;
        String a[]={"はなこ", "たろう", "じろう", "あかね"};
        try{
            i=Integer.parseInt(args[0]);
            System.out.println("こんにちは、"+a[i]+"さん。");
        }
        catch(Exception e){}
    }
}
```

【ワンポイント】文字列の配列

文字列を複数利用(2バイト以上)するときには、String オブジェクトを使用します。
char 型の配列では2バイトまでしか代入することができません。
書式は次のようになります。

```
String a[]={"はなこ", "たろう", "じろう", "あかね"};
```

上記のようにしますと、a[0]にははなこ、a[1]にはたろう、・・・がそれぞれ入ります。

【ワンポイント】文字列の比較

文字列を比較するときは、==ではなく、文字列オブジェクトの equals()メソッドを使います。
たとえば、次のような場合はそれぞれ、true、false となります。

```
class He{
    public static void main(String args[]){
        System.out.println("たろう".equals("たろう"));
        System.out.println("たろう".equals("じろう"));
    }
}
```

【実行結果】

```
C:\¥yy>java He
true
false
```

①.equals(②) ...①と②が等しければ true、①と②が等しくないときは false が返されます。