

情報処理演習 5

kamahara@port.kobe-u.ac.jp

関数（メソッド）

- ◎ 関数は、複数の命令をまとめたもの
- ◎ ブロックとの違い
 - ◎ 引数で値が渡せる： $f(x)$ や $f(x,y)$ の x や y が引数
 - ◎ 戻り値がある： $y=f(x)$ の y が戻り値

JAVAの関数の書式

- ◎ 関数（メソッド）はクラスの中にある
- ◎ 引数と戻り値のないシンプルな形
 - ◎ `public static void FUNCTION() { ... }`
- ◎ 引き数だけある場合
 - ◎ `public static void FUNCTION( int a ) { ... }`
- ◎ 戻り値だけある場合
 - ◎ `public static int FUNCTION() { ... }`

足し算をする関数

- ◎ 引き数 2 つと、戻り値がある場合
- ◎ 引き数には「型」を指定する必要がある
- ◎ 戻り値も型を指定する
- ◎ $x+y$ の結果を変数resultに入れて、それを戻す

```
public static double add( double x, double y ) {  
    double result = x + y;  
    return result;  
}
```

MOJI.JAVAを開く

```
C:\ コマンド プロンプト
Z:\>cd joho

Z:\joho>dir
ドライブ Z のボリューム ラベルは OS です
ボリューム シリアル番号は A2B4-B072 です

Z:\joho のディレクトリ

2011/12/01  11:21    <DIR>          .
2011/12/01  11:21    <DIR>          ..
2011/11/17   11:31             438 HelloWorld.class
2011/11/17   11:31             123 HelloWorld.java
2011/12/01   14:36          1,327 Moji.class
2011/12/01   14:56             619 Moji.java
2011/11/17   14:50             638 Sum.class
2011/11/17   14:49             163 Sum.java
               6 個のファイル                3,308 バイト
               2 個のディレクトリ  24,730,103,808 バイトの空き領域

Z:\joho>notepad Moji.java

Z:\joho>■
```

関数を追加

```
Moji.java - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

import java.io.*;
public class Moji {
    public static void main(String[] args) {
        String str1 = "お名前は?";
        String str2 = "さん、こんにちは";
        String name = "鎌原";

        try {
            BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
            System.out.print("名前は?");
            name = br.readLine();
            System.out.print("a=");
            str1 = br.readLine();
            System.out.print("b=");
            str2 = br.readLine();
        } catch (IOException e) {

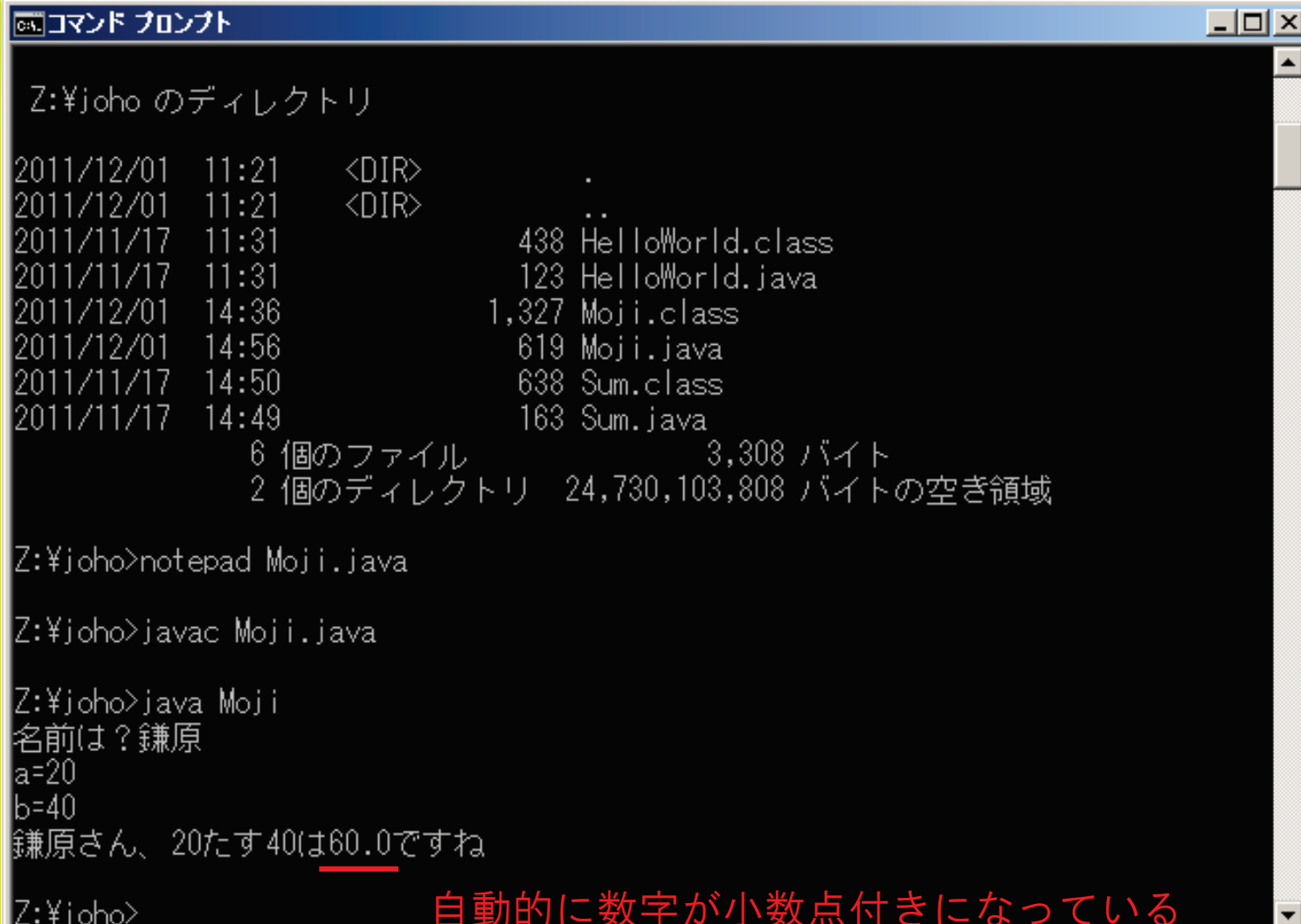
        }

        int a = Integer.parseInt(str1);
        int b = Integer.parseInt(str2);

        System.out.println(name+"さん、 "+a+"たす"+b+"は"+add(a,b)+"ですね"      );
    }

    public static double add( double x, double y ) {
        double result;
        result = x + y;
        return result;
    }
}
```

実行してみる



The screenshot shows a Windows Command Prompt window titled "コマンド プロンプト". The current directory is "Z:\¥joho のディレクトリ". A directory listing shows files: "HelloWorld.class" (438 bytes), "HelloWorld.java" (123 bytes), "Moji.class" (1,327 bytes), "Moji.java" (619 bytes), "Sum.class" (638 bytes), and "Sum.java" (163 bytes). It also shows "6 個のファイル" and "3,308 バイト". Below this, it says "2 個のディレクトリ 24,730,103,808 バイトの空き領域". The user then runs "notepad Moji.java", "javac Moji.java", and "java Moji". The output of the Java program is: "名前は？鎌原", "a=20", "b=40", and "鎌原さん、20たす40は60.0ですね". The number "60.0" is underlined. The prompt "Z:\¥joho>" is visible at the bottom.

```
コマンド プロンプト

Z:\¥joho のディレクトリ

2011/12/01  11:21    <DIR>          .
2011/12/01  11:21    <DIR>          ..
2011/11/17   11:31             438 HelloWorld.class
2011/11/17   11:31             123 HelloWorld.java
2011/12/01   14:36          1,327 Moji.class
2011/12/01   14:56             619 Moji.java
2011/11/17   14:50             638 Sum.class
2011/11/17   14:49             163 Sum.java
           6 個のファイル                3,308 バイト
           2 個のディレクトリ 24,730,103,808 バイトの空き領域

Z:\¥joho>notepad Moji.java

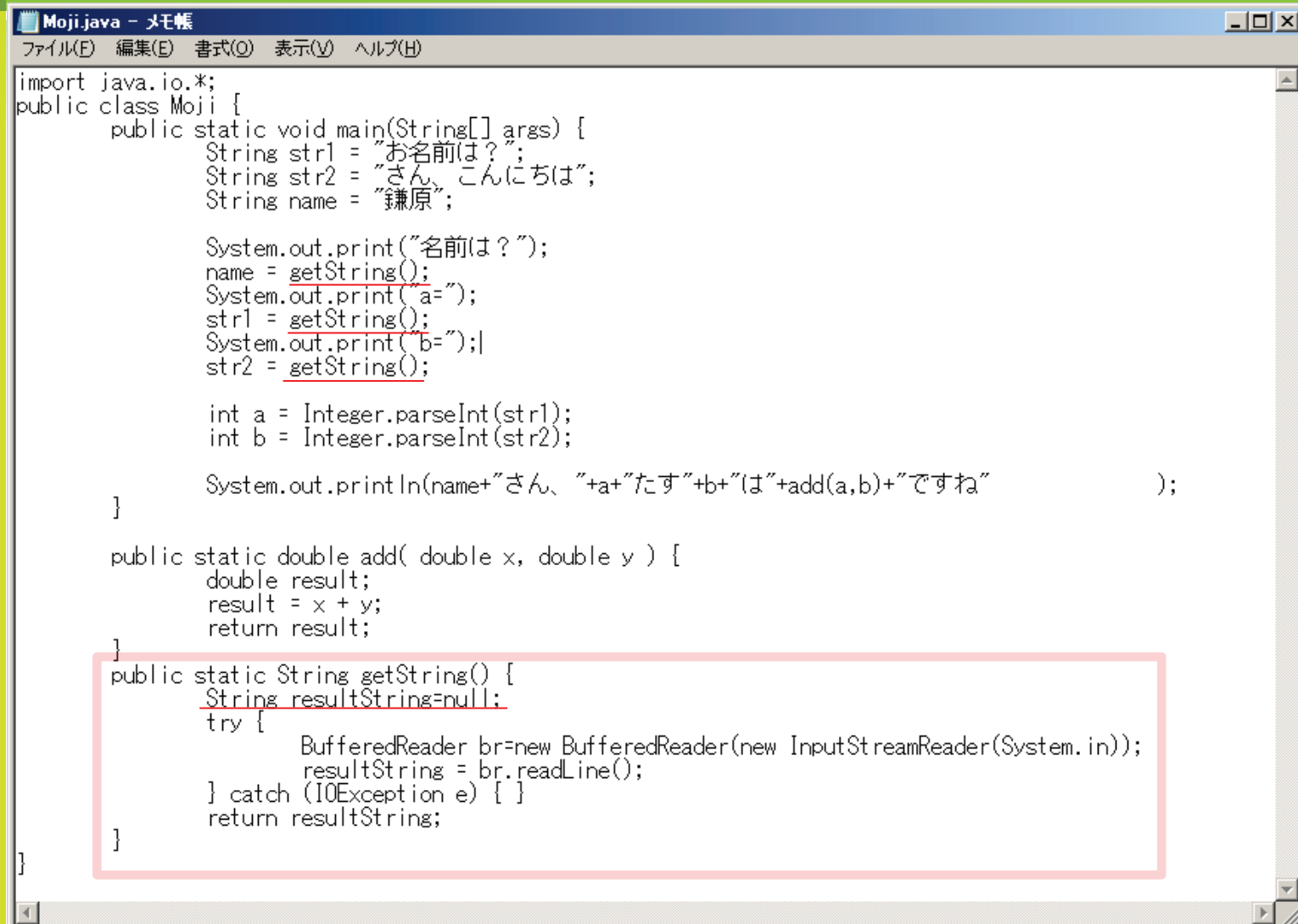
Z:\¥joho>javac Moji.java

Z:\¥joho>java Moji
名前は？鎌原
a=20
b=40
鎌原さん、20たす40は60.0ですね

Z:\¥joho>
```

自動的に数字が小数点付きになっている

関数は繰り返しを まとめることができる



```
Moji.java - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

import java.io.*;
public class Moji {
    public static void main(String[] args) {
        String str1 = "お名前は?";
        String str2 = "さん、こんにちは";
        String name = "鎌原";

        System.out.print("名前は?");
        name = getString();
        System.out.print(" a=");
        str1 = getString();
        System.out.print(" b=");
        str2 = getString();

        int a = Integer.parseInt(str1);
        int b = Integer.parseInt(str2);

        System.out.println(name+"さん、 "+a+"と"+b+"は"+add(a,b)+"ですね" );
    }

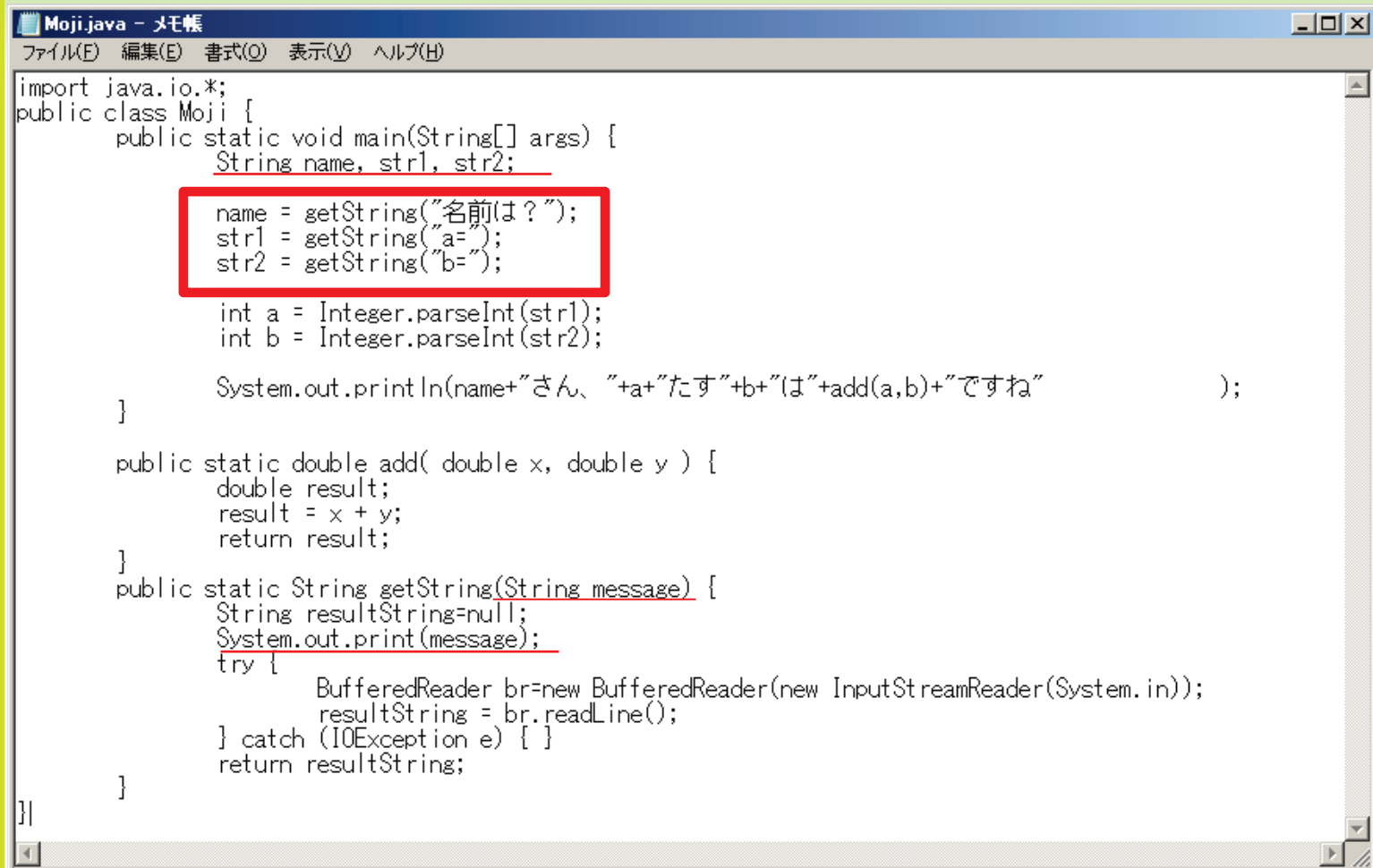
    public static double add( double x, double y ) {
        double result;
        result = x + y;
        return result;
    }

    public static String getString() {
        String resultString=null;
        try {
            BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
            resultString = br.readLine();
        } catch (IOException e) { }
        return resultString;
    }
}
```


文字列の初期値はNULL

- ◎ String resultString = null; で初期化する

出力もまとめる



```
Moji.java - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

import java.io.*;
public class Moji {
    public static void main(String[] args) {
        String name, str1, str2;

        name = getString("名前は何?");
        str1 = getString("a=");
        str2 = getString("b=");

        int a = Integer.parseInt(str1);
        int b = Integer.parseInt(str2);

        System.out.println(name+"さん、 "+a+"たす "+b+"は "+add(a,b)+"ですね" );
    }

    public static double add( double x, double y ) {
        double result;
        result = x + y;
        return result;
    }

    public static String getString(String message) {
        String resultString=null;
        System.out.print(message);
        try {
            BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
            resultString = br.readLine();
        } catch (IOException e) { }
        return resultString;
    }
}
```

乱数で値を生成する

- ◎ `Math.random()` で0.0から1.0未満の数値が得られる
- ◎ 0から9までの整数が欲しい時は、10倍して切り捨て
 - ◎ 例: `int rand = (int)(Math.random()*10);`
- ◎ 1から10までの整数が欲しい時は+1する
 - ◎ 例: `int rand = (int)(Math.random()*10)+1;`

足し算ドリル

```
Moji.java - メモ帳
ファイル(E) 編集(E) 書式(O) 表示(V) ヘルプ(H)

import java.io.*;
public class Moji {
    public static void main(String[] args) {
        String name, str1, str2;

        name = getString("名前は何?");
        int a = (int)(Math.random()*10)+1;
        int b = (int)(Math.random()*10)+1;

        int c = Integer.parseInt(getString(a+" "+b+"=?"));
        System.out.println(name+"さん");
        if ( c==add(a,b) ) {
            System.out.println("正解です");
        } else {
            System.out.println("間違いです。正解は"+add(a,b)+"です");
        }
    }

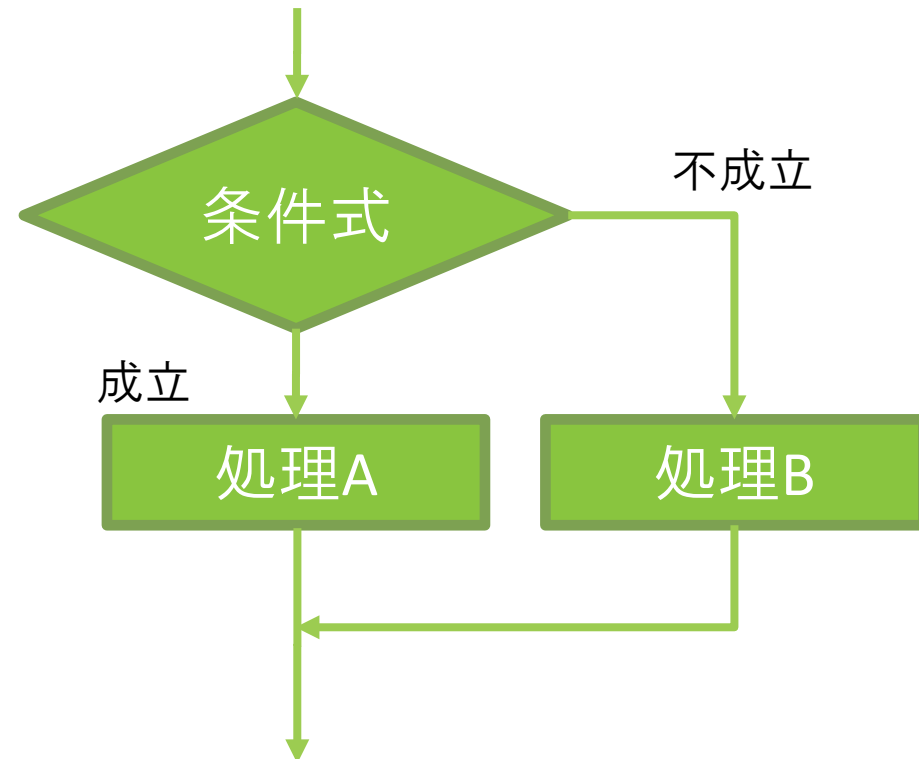
    public static double add( double x, double y ) {
        double result;
        result = x + y;
        return result;
    }

    public static String getString(String message) {
        String resultString=null;
        System.out.print(message);
        try {
            BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
            resultString = br.readLine();
        } catch (IOException e) { }
        return resultString;
    }
}
```

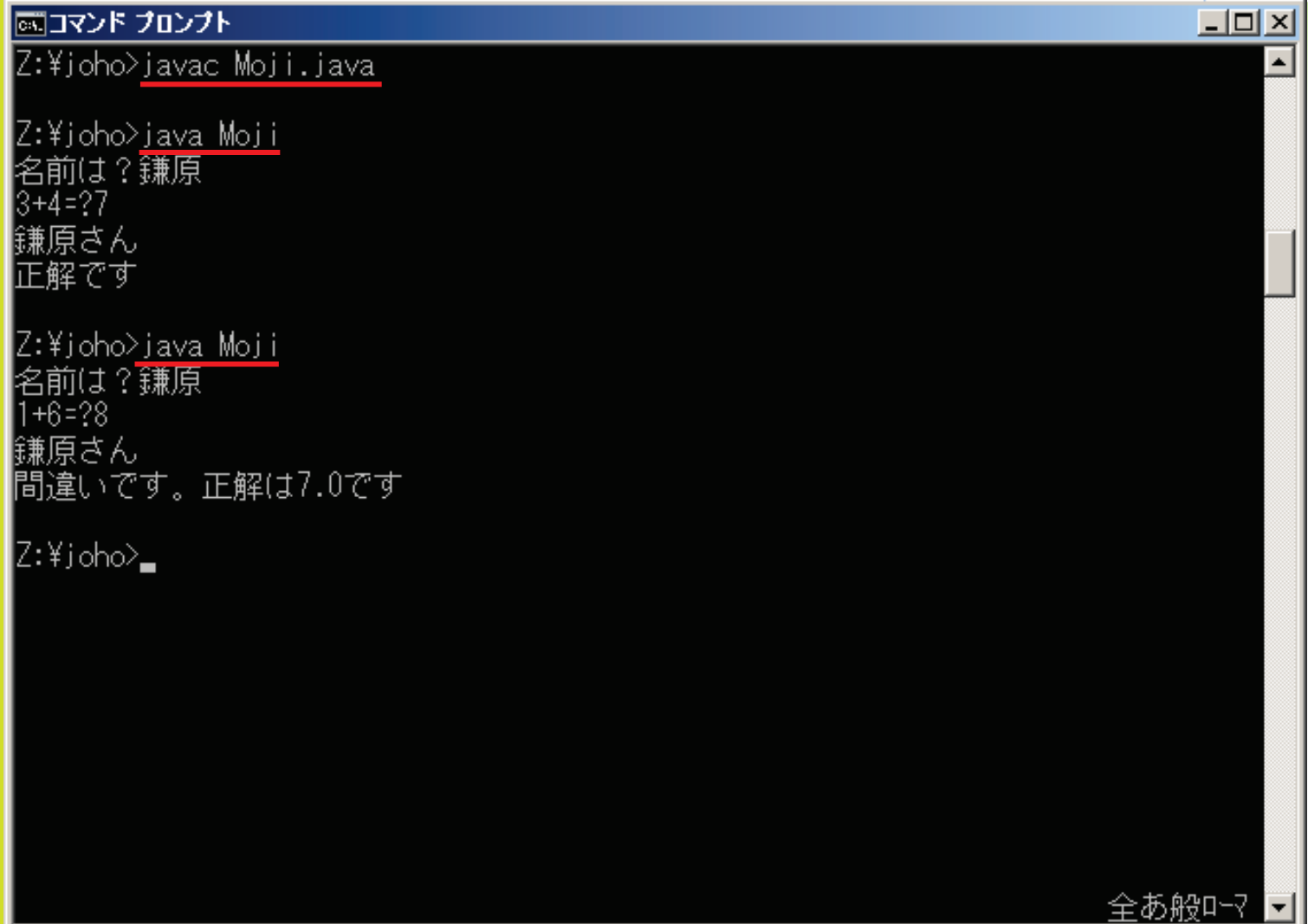
IF文

- ◎ 条件判断をする

```
if ( 条件式 ) {  
    処理A;  
} else {  
    処理B;  
}
```



実行してみる



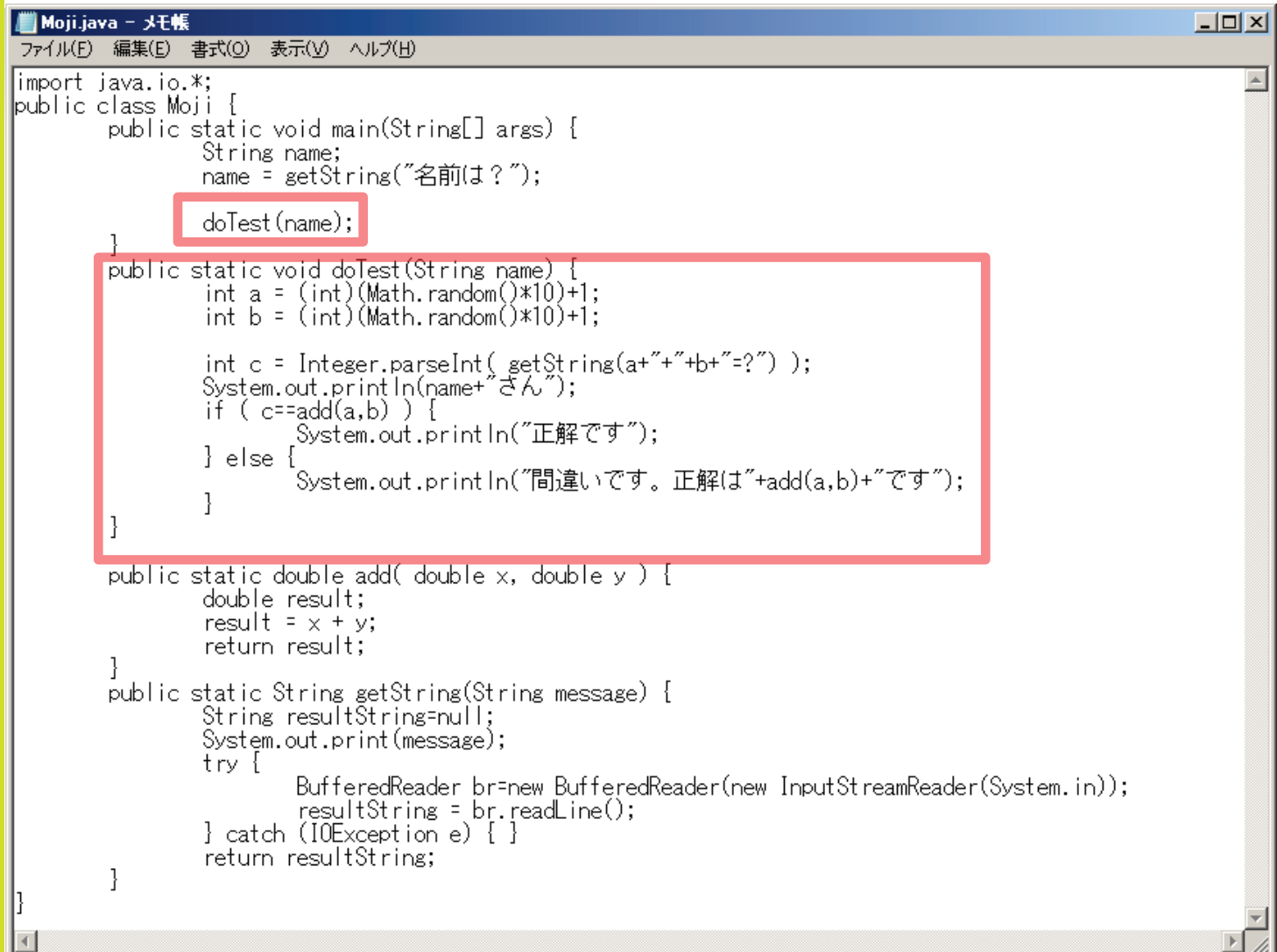
```
C:\> コマンド プロンプト
Z:\¥joho> javac Moji.java
Z:\¥joho> java Moji
名前は？鎌原
3+4=?7
鎌原さん
正解です

Z:\¥joho> java Moji
名前は？鎌原
1+6=?8
鎌原さん
間違いです。正解は7.0です

Z:\¥joho> _
```

全あ般0-7

問題をdoTest()にまとめる



```
Moji.java - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

import java.io.*;
public class Moji {
    public static void main(String[] args) {
        String name;
        name = getString("名前は何?");
        doTest(name);
    }
    public static void doTest(String name) {
        int a = (int)(Math.random()*10)+1;
        int b = (int)(Math.random()*10)+1;

        int c = Integer.parseInt( getString(a+" "+b+"=?") );
        System.out.println(name+"さん");
        if ( c==add(a,b) ) {
            System.out.println("正解です");
        } else {
            System.out.println("間違いです。正解は"+add(a,b)+"です");
        }
    }
    public static double add( double x, double y ) {
        double result;
        result = x + y;
        return result;
    }
    public static String getString(String message) {
        String resultString=null;
        System.out.print(message);
        try {
            BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
            resultString = br.readLine();
        } catch (IOException e) {}
        return resultString;
    }
}
```

課題

- ◎ forループを使って、問題を5回繰り返すプログラムに改造しなさい
- ◎ 追加課題：その際、正解なら戻り値を1、不正解なら戻り値が0になるようにして、正解の数を最後に表示するようにすること（正解を4回と、間違いを1回で実行すること）
- ◎ 実行結果とプログラムを添付でメールすること

課題の提出

- ◎ コマンドプロンプトの画像とMoji.javaを添付ファイルとして提出する
- ◎ 件名：情報処理演習 5 学籍番号 名前
- ◎ 宛先：kamahara@port.kobe-u.ac.jp