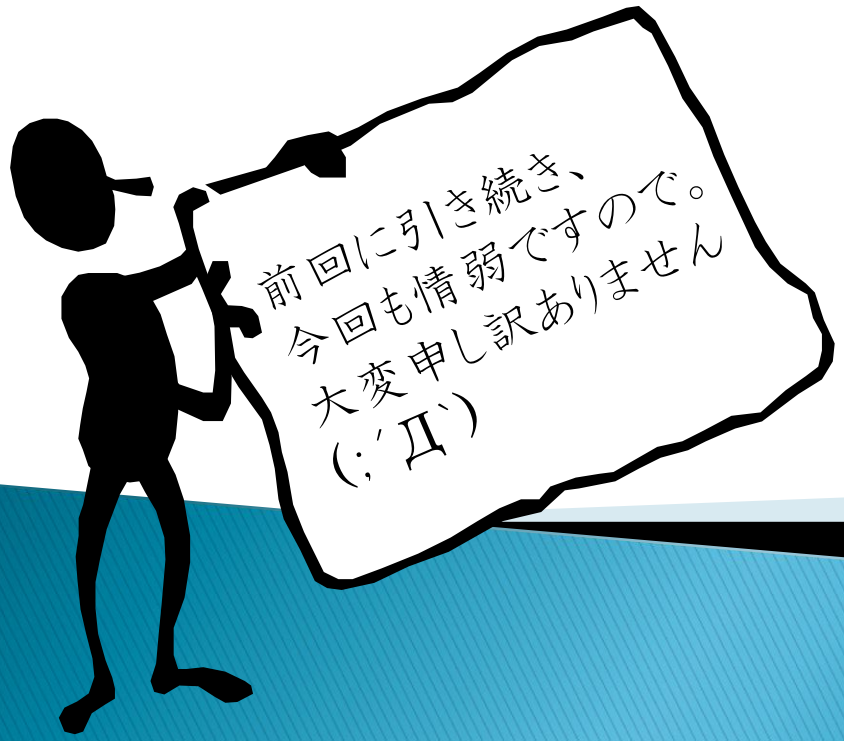
JAVA講座　第４回



情科2年　　鈴木

～とりあえず準備～

Eclipseを起動して
初回で作ったクラス「K3java1」に
新しくパッケージ「lesson04」を作成、(*´∀｀)ノ

メソッドってなんだろう？(復習)

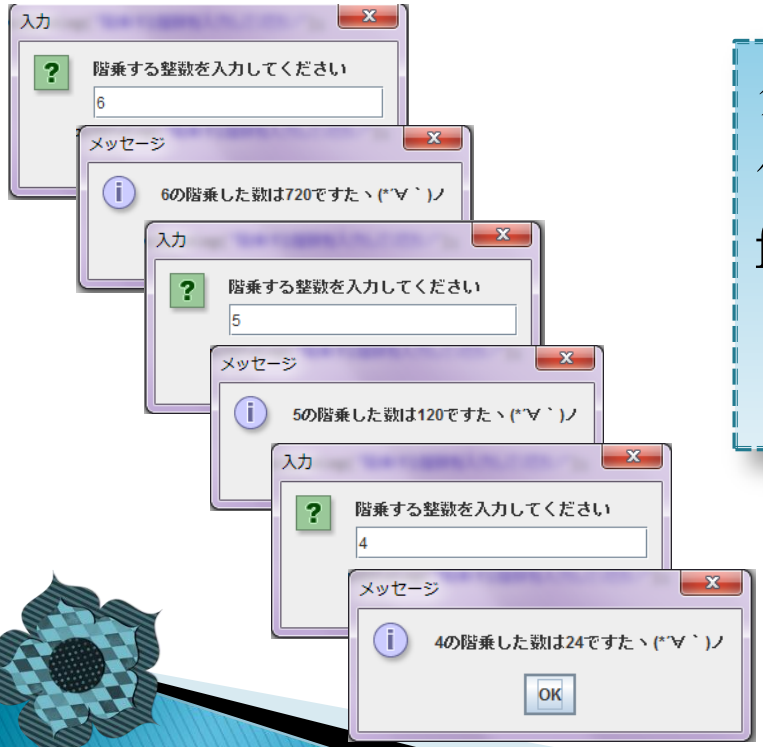
メソッドとは・・・
プログラムが難しくなると、それぞれ
処理をするプログラムをバラバラに
して書いたほうが効率的になる。
そんな時に使うのがメソッド！

例えば、100個の指定された数の合計を
計算したりするときに、1つ1つ「○＋●」
の形で書くのは面倒！

とりあえず～

復習

3回、数字を入力してその数字の階乗を
表示するプログラムを書きなさい。



クラス名「Factorial1」
使うメソッド:
factorial()…
入力ダイアログを表示し、その
階乗を求めて表示するメソッド

←「6」「5」「4」で実行した時の例

(解答例)復習：Factorial1

Factorial1.java | Add.java | Quiz.java | heapsort.java

```
1  package lesson04;
2
3  import javax.swing.JOptionPane;
4
5  public class Factorial1 {
6
7      public static void main(String[] args) {
8          new Factorial1().start();
9      }
10
11     void start() {
12         for (int i = 0; i < 3; i++) {
13             factorial();
14         }
15     }
16
17     void factorial() {
18         String input = JOptionPane.showInputDialog("階乗する整数を入力してください");
19         int n = Integer.parseInt(input);
20         int result = 1;
21         for (int i = 1; i <= n; i++) {
22             result = result * i;
23         }
24         JOptionPane.showMessageDialog(null, n + "の階乗した数は" + result + "ですたぃ (*´∀`)ノ");
25     }
26 }
```

ここで動作をメソッド
factorial()に移す！！

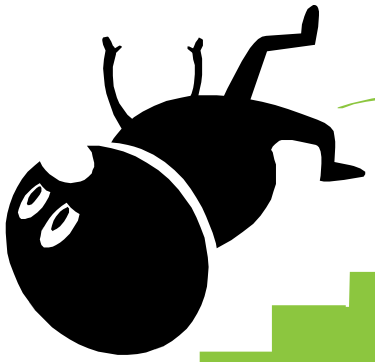
で、、、今回は(*´∀`*)

引数(ひきすう)と**戻り値**を用いる
複数のメソッドを利用したプログラムを書いてみよう！

引数ってなに？？

引数とは・・・
プログラム中で関数やメソッドなどを
呼び出すときに渡す値のことです！
例えば、最初のメソッドで入力した値を次のメソッドに引き継ぎたい際に利用するのが引数だ(´∀`*)
引数には**実引数**と**仮引数**があるのです♪

引き継ぐときはしっかりと
しないとダメだよ～！！



～実引数と仮引数～

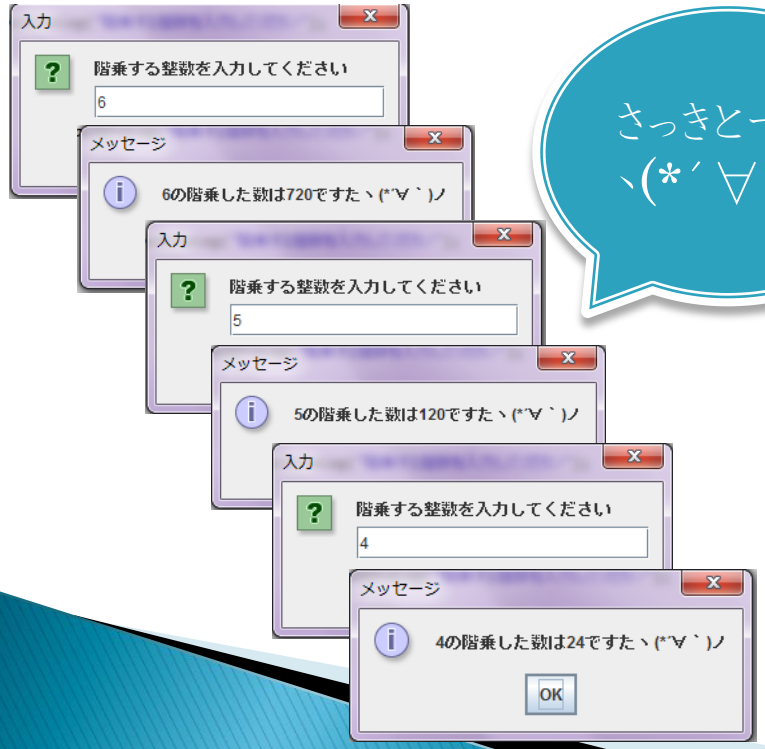
実引数とは…
今実行してるメソッド内の値で次のメソッドで利用したいときに引数として利用する変数や値のこと！

仮引数とは…
引き継がれたメソッド変数を再度宣言するときに利用する変数や値のこと！

```
void start( ) {

    メソッド名(実引数);

}
void  メソッド名(仮引数)  {

    プログラム内容

}
```

では、引数を使ってさっきの問題～

例1・・・
3回、数字を入力してその数字の階乗を表示するプログラムを書きなさい。



さっきと一緒
ヽ(*´∀｀)ノ

クラス名「Factorial2」
使うメソッド:
factorial(int n)…
仮引数nの階乗を求めて表示するメソッド

←「6」「5」「4」で実行した時の例

(解答例)例1：Factorial2

Factorial2.java | Factorial1.java | Add.java | Quiz.java | heapsort.java

```
1  package lesson04;
2
3  import javax.swing.JOptionPane;
4
5  public class Factorial2 {
6
7      public static void main(String[] args) {
8          new Factorial2().start();
9      }
10
11     void start() {
12         for (int i = 0; i < 3; i++) {
13             String input = JOptionPane.showInputDialog("階乗する整数を入力してください");
14             int n = Integer.parseInt(input);
15             factorial(n);
16         }
17     }
18
19     void factorial(int n) {
20         int result = 1;
21         for (int i = 1; i <= n; i++) {
22             result = result * i;
23         }
24         JOptionPane.showMessageDialog(null, n + "の階乗した数は" + result + "ですたぃ(*´∀｀)ノ");
25     }
26 }
```

ここで入力した整数nをメソッドfactorial(int n)に引き継ぐ！

まぁこんなかんじで。。。。

まだ、プログラムが長くないので
あんまりありがたみが感じないかもしれませんが
だんだんとわかってくると思うので、
スルーさせてもらいます(o^-^)(このあたりが情弱)

みんなハイチュウ食べるべし♪

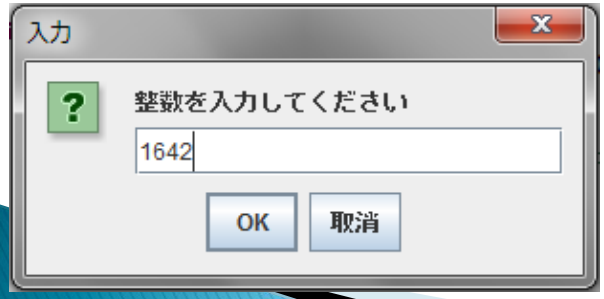
では、チャレンジ(´∀`*)♪

演習1

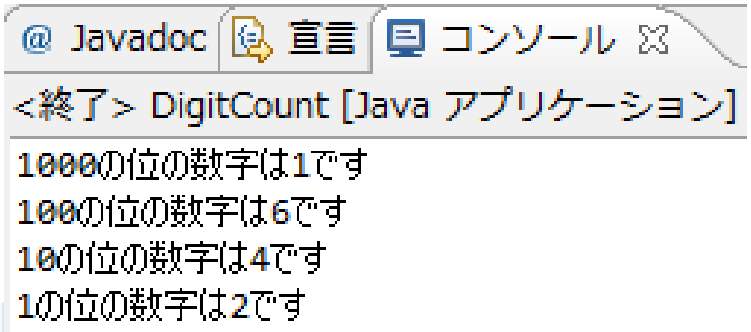
適当な整数を入力してその数字の千の位、百の位、十の位、一の位の数字を、それぞれコンソールに表示する再帰を利用したプログラムをつくってみよう！

クラス名「DigitCount」
使うメソッド:
digit(int n, int dig)…
位とその位の数を計算し、コンソールに表示するメソッド

初回の講座の最後にやった問題と同じ考え方を利用するよ♪
覚えてるかな？？？



(1642を入力した場合)→



ヒント(≧∇≦)/ 絶対にこうじゃなきゃいけないわけじゃないからね？

digit(int n, int dig)についての説明。
まず、digitはdigが0より大きい時処理を実行する。
nは引き継いだ値、digは桁を表す値になるようにする。
桁を調べる値が1642の場合、
再帰の最初の引数は
n = 1642, dig = 1000となり、digは0より
多いのでdigit内の処理が実行され、
1000の位の数字は1です
という文字がコンソールに表示される。
最後に再帰させる為、dig(変数1,変数2);を書く。

ここで引き継ぐ変数1,2は
自分で考えてみて‼

(解答例)演習1:DigitCount

DigitCount.java | DigitCount1.java | Calculation.java | Heapsort

```java
package practice;

public class DigitCount {

    public static void main(String[] args) {
        new DigitCount().start();
    }

    void start() {
        int n = 1642;
        digit(n, 1000);
    }

    void digit(int n, int dig) {
        if (dig > 0) {
            int much = n % dig;
            int result = (n - much) / dig;
            System.out.println(dig + "の位の数字は" + result + "です");
            digit(much, dig / 10);
        }
    }
}
```

特定の数という事で、
最初に数字を直接プログラムに打ち込んでもだいじょーぶ(#^.^#)

もし、入力ダイアログから入れた値でこのプログラムを実行したい場合は
次のページを参考に！

(別解)演習1：DigitCount

こんなふうに書くと、
何桁の数字を入力しても
それぞれの桁の値が出
るようになっていますが、
そこまでしなくても
だいじょーぶです^^；

●ちょこっと知識●

int 変数名 =
Integer.toString(n).length();

これでnの数字の桁数がわかる！

DigitCount.java | Calculation.java | Heapsort.java | MyMaxHeapify.java

```
1  package practice;
2
3  import javax.swing.JOptionPane;
4
   public class DigitCount {

       public static void main(String[] args) {
           new DigitCount().start();
       }

       void start() {
           String input = JOptionPane.showInputDialog("整数を入力してください");
           int n = Integer.parseInt(input);
           int keta = Integer.toString(n).length();
           // ketaでは桁数を求めています！今はよくわからなくても大丈夫です＞＜
           int length = 1;
17         for (int i = 1; i < keta; i++) {
18             length = length * 10;
19         }
20         digit(n, length);
21     }
22
23     void digit(int n, int dig) {
24         if (dig > 0) {
25             int much = n % dig;
26             int result = (n - much) / dig;
27             System.out.println(dig + "の位の数字は" + result + "です");
28             digit(much, dig / 10);
29         }
30     }
31 }
```

メソッドの型～

さっきまで作っていたプログラムでは、
当然のことのようにvoid メソッド名()
と書いていましたが、

実はvoid 以外の型を利用しても
メソッドを作ることができます！

しかし。。。
void型以外でメソッドを作った場合、
戻り値を返さなければならない！！

な・・・なん(;゜Д゜)ですとー!!

戻り値って？？～

戻り値とは、そのメソッド内の処理を実行したあとに
実行するメソッドへ引き継ぐ値のこと！
void型のメソッドの場合は戻り値が不要ですが、
それ以外の型の場合はメソッドの型と同じ型の変数を
返さなければなりません！
そして、返すときにreturnというのを使います!

例えば

String型　→　戻り値　文字列
int型　→　戻り値　整数
double型　→　戻り値　実数

などなど…

では、練習(((o(*ﾟ▽ﾟ*)o)))

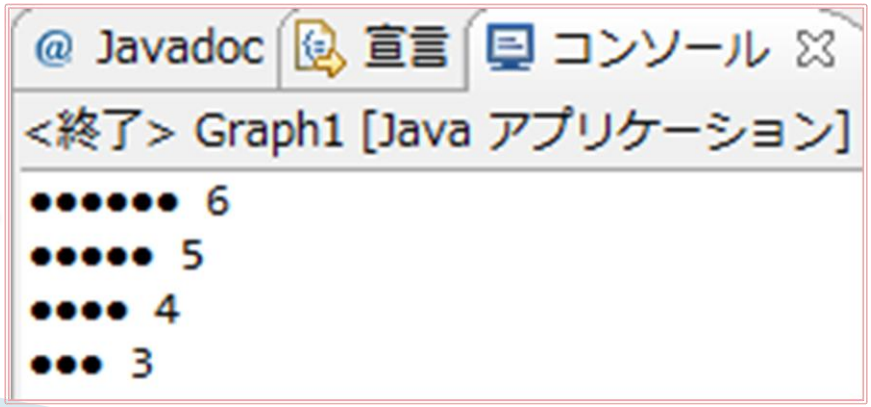
例2・・・

5回整数を入力してもらい、入力された数字を使ってコンソールに以下のような表示が出るように作ってみよう！

クラス名「Graph1」

使うメソッド：graph(int n)…

仮引数n個分の●を並べた文字列を返すメソッド



(解答例)例2:Graph1

String型のメソッドなので、
文字列を
こんな感じで
Return する

Graph1.java | Factorial2.java | Factorial1.java | Ado... | Quiz.j

```
1  package lesson04;
2
3  import javax.swing.JOptionPane;
4
5  public class Graph1 {
6
7      public static void main(String[] args) {
8          new Graph1().start();
9      }
10
11     void start() {
12         for (int i = 0; i < 5; i++) {
13             String input = JOptionPane.showInputDialog("整数を入力してください");
14             int a = Integer.parseInt(input);
15             String result = graph(a);
16             System.out.println(result + " " + a);
17         }
18     }
19     String graph(int n) {
20         String graph = "";
21         for (int i = 0; i < n; i++) {
               graph = graph + "●";
           }
24         return graph;
25     }
26 }
```

まぁ実際に書いてみよう！

演習2
4つの整数を入力させて、
それらの合計を表示するプログラム
を書いてみよう！

クラス名「Add」
利用するメソッド
add(int a, int b)…
　　2つの仮引数a, bを
　　足した値を返す。

この場合、
仮引数はaとb
になる！

(解答例)演習2： Add

2つの整数を足した値を返すメソッドaddを利用することで、2つずつそれぞれ足して、最後にそれぞれの合計を足すという右のような書き方ができる！

Add.java | Quiz.java

```
1  package lesson04;
2
3  import javax.swing.JOptionPane;
4
5  public class Add {
6
7      public static void main(String[] args) {
8          new Add().start();
9      }
10
11     void start() {
12         String input1 = JOptionPane.showInputDialog("1つ目の整数を入力してください");
13         String input2 = JOptionPane.showInputDialog("2つ目の整数を入力してください");
14         String input3 = JOptionPane.showInputDialog("3つ目の整数を入力してください");
15         String input4 = JOptionPane.showInputDialog("4つ目の整数を入力してください");
16         int a = Integer.parseInt(input1);
17         int b = Integer.parseInt(input2);
           int c = Integer.parseInt(input3);
           int d = Integer.parseInt(input4);
20         int result = add(add(a, b), add(c, d));
21         JOptionPane.showMessageDialog(null, "全ての合計は" + result + "なんだ！");
22     }
23
24     int add(int a, int b) {
25         return a + b;
26     }
27 }
28
```