

2012年4月26日（木）開催

計算技術研究会

第2回JAVA講座

～条件分岐・ループ・擬似乱数～

本日の担当者

根岸 拓郎

CS科 2年B組 数値解析班所属

GBC SA

小藺 絢介

CS科 2年B組 アプリ班所属

前回の復習

- ◎ そもそもJAVAって何？

⇒ プログラミング言語のひとつであり、
携帯のアプリケーション、
ウェブページのアニメーション等
様々な場面で活用されている。



前回の復習

◎ プロジェクト、パッケージ、クラス



- ・ プロジェクト
- ・ パッケージ
- ・ クラス

プロジェクトは、
Eclipseが企業向けに管理しやすいシステムとして導入している
のでEclipseでやるときの準備作業工程と
思っただいて結構です。

パッケージは作品のまとめり。

クラスはプログラムの中身となります。
(クラス名の最初は大文字)

前回の復習

◎ プログラムの流れ（雛形）

```
*Test.java x
1 package test;           //所属パッケージ名
2
3 import javax.swing.JOptionPane; //中身使いたい道具を出す。
4                             //JOptionPane
5
6 public class Test {     //クラス名
7
8     public static void main(String args[]) { //クラス実行に必要な
9
10        new Test().start(); //中身(start)の導入
11
12    }
13
14    void start() {
15        /*
16         * プログラムの中身
17         */
18    }
19
20 }
21
```

ワンポイント

◎ コメント（注釈）をつけよう！

スラッシュ2回（//）でその行の最後まで
の部分にはプログラム外として扱われる。

また、/* */で囲まれた部分（複数行にも有効）もプログラム外として扱われる。

これらを利用して、変数の解説（どのように扱っているのか）、プログラムの解説などをいれ、わかりやすいようにすると良いでしょう。

前回の復習

- ◎ 変数の種類

String 型 文字列（ “ ” 内の文字）

int 型 整数

double 型 実数

（小数点がつかない場合でも「.0」が補われます。）

- ◎ 生成方法

型 （変数） = 変数に記憶させたいもの；

例： **String word = “hello!”;**

int number = 7;

前回の復習

◎ プログラムの中身で使った命令

```
String (変数) = JOptionPane.showInputDialog(“文字列を入力”);
```

⇒入力欄を表示させて**文字列**を記憶させる。(入力)

※(変数)にあたる部分は自分がわかりやすいものにする。

例：message, comment, x, y ,inputWords など。

ただし、スペースを含む、数字から始まる、記号を含むものは使えません。

前回の復習

◎ プログラムの中身で使った命令

`JOptionPane.showMessageDialog(null, “表示させたい文字列”);` または、

`JOptionPane.showMessageDialog(null, (変数));` または、

`JOptionPane.showMessageDialog(null, (数値));`

⇒ **文字列**や**変数**を表示させる。(出力)

※`””`内に変数を入れても変数名自体が表示されるので注意。

※`””`と**(変数)**をあわせるときは、**+**でつなげる。

(演算と同じ記号なので注意)

例：`JOptionPane.showMessageDialog(null, “xの値は” + x + “です。”);`

前回の復習

```
void start() {  
  
    double x=0;  
    x=9; ← 変数  
  
    JOptionPane.showMessageDialog(null, x);  
}
```

メッセージ

i 9.0

了解

変数

前回の復習

◎ プログラムの中身で使った命令

```
int (変数) = Integer.parseInt( (変数 '));
```

⇒ (変数 ') を **整数** の型に変換する。

```
double (変数) = Double.parseDouble( (変数 '));
```

⇒ (変数 ') を **実数** の型に変換する。

(変数 ') は入力されたものを用いている。

なぜ、このようなことをするのかというと、入力に関しては全てString型として扱われてしまうためであり、これらの行為を行わないと整数、実数に変換することができない。

前回の復習

- ◎ プログラムの中身で使った命令変換の例

```
void start() {  
  
    String inputNumber  
        = JOptionPane.showInputDialog("数字を入力");  
  
    int x = Integer.parseInt(inputNumber); // 整数に変換  
  
    JOptionPane.showMessageDialog(null, x); // 整数の表示  
}
```



全角数字でも可。
ただし、文字が入力されると
エラーがでる。

前回の復習

◎ 演算方法

足し算 +

引き算 -

掛け算 *

割り算 /

余り %

※ = (イコール) はいりません。

例 : `int x = 10;`

`int y = 5;`

`JOptionPane.showMessageDialog(null, x+y);`

⇒これでx(10)+y(5)の結果である15が表示される。

※intでの割り算は小数点以下切捨てになります。

クラスを作しましょう

- ◎ 前回「k3java1」プロジェクトを作りましたね
- ◎ その中に「lesson02」パッケージを作しましょう
- ◎ そこに「Example01」クラスを作しましょう
- ◎ クラス名には日本語を使わないようにしましょう

復習問題

- ◎ 整数を入力して、その値を5乗した結果を表示するプログラムを作りましょう！

本日の内容

- ◎ 条件分岐(if文)
- ◎ ループ(for文、 while文)
- ◎ 擬似乱数(Math.random();)

条件分岐(if文)

- If文は、「もし～だったら、...をする」の機能を持ちます。
- それ以外の条件を定めるときは、**else (if)** を使う。
※ifは、パターンをまだ増やす場合に使う。
- ～の部分に当たるところは、**比較演算子**を用いて定めます。

比較演算子

日本語	数学表記	JAVA表記
AとBは等しい	$A = B$	<code>A == B</code>
AとBは等しくない	$A \neq B$	<code>A != B</code>
AよりBが大きい	$A < B$	<code>A < B</code>
AよりBが小さい	$A > B$	<code>A > B</code>
AがB以上	$A \geq B$	<code>A >= B</code>
AがB以下	$A \leq B$	<code>A <= B</code>
AかつB	$A \cap B$	<code>A && B</code>
AまたはB	$A \cup B$	<code>A B</code>

* 「！」はNOTの役割を果たしている！

余談 &と|が2つあると、A次第でBの判定は行わない。
&&なら、Aがfalseなら、Bの判定をせずにfalseが出力、
||なら、Aがtrueなら、Bの判定をせずにtrueが出力される。
ちなみに、1個だけだと、両方チェックするので速度が遅い！

例えば...

```
10 void start() {  
11     int a=0;  
12     int b=0;  
13     if (a=b) {  
14  
15     }  
16 }
```

aとbを比較するときには「a=b」ではいけません
「=」 1個だと代入の意味になってしまいます
なので、「=」を2個にしましょう！

条件分岐(if文)

if文内の命令が1個なら{}は省略できます

```
void start() {  
    String input = JOptionPane.showInputDialog("整数を入力してください");  
    int x = Integer.parseInt(input);  
    if(x >= 100) {  
        JOptionPane.showMessageDialog(null, "高得点です!!");  
    }  
    else if(x >= 80) {  
        JOptionPane.showMessageDialog(null, "そこそこの点です");  
    }  
    else if(x >= 60) {  
        JOptionPane.showMessageDialog(null, "ギリギリ合格点です");  
    }  
}
```

else ifは何個でも追加
することが出来ます

elseをつけないことも
できます

else ifの用途

- ◎ 下のよう^にelse ifのelseを省いて実行してみましょう。

```
void start(){
    String input=JOptionPane.showInputDialog("点数を入力。");
    int x=Integer.parseInt(input);
    if(x>=100){
        JOptionPane.showMessageDialog(null,"高得点です！");
    }if(x>=80){
        JOptionPane.showMessageDialog(null,"そこそこの点です");
    }if(x>=60){
        JOptionPane.showMessageDialog(null,"ギリギリ合格点です");
    }
}
```

else ifの用途・注意

- 100以上の数値を入力したら、ダイアログが3個出てきました。
- Ifは、前の条件などはすべて無視して新たな条件をつくる。
- else ifは、前のifの条件以外かつ、else ifの条件を満たすときに適用されます。

具体例 1 — 1

```
String xNumber =  
JOptionPane.showInputDialog(“xの数字を入力”);  
String yNumber =  
JOptionPane.showInputDialog(“yの数字を入力”);  
String zNumber =  
JOptionPane.showInputDialog(“zの数字を入力”);  
  
int x = Integer.parseInt(xNumber);  
int y = Integer.parseInt(yNumber);  
int z = Integer.parseInt(zNumber);
```

具体例 1 - 2

```
if(x == y && y == z){  
    JOptionPane.showMessageDialog(null,  
        "3つの数は同じ数です");  
}
```

```
else if(x == y || y == z || z == x){  
    JOptionPane.showMessageDialog(null,  
        "1数のみ仲間はずれがあります。");  
}
```


具体例 1 — 3

```
else {  
    JOptionPane.showMessageDialog(null,  
        "3 数すべて異なります。");  
}
```

3数以上の比較の注意

- ◎ 3数以上（具体例の x, y, z ）の比較のときに $x == y == z$ とまとめて使うことができない！
- ◎ 3数以上の比較をするときは $\&\&$ や $||$ を活用する！

例題

- ◎ 「Example02」クラスを作って…
- ◎ if, else を使って
- ◎ 何か整数を入力させて、その値が偶数か奇数か判断させてみましょう

- ◎ JOptionPane.showMessageDialog() を使って
- ◎ 「～は奇数です」「～は偶数です」とかメッセージを出してみましょう

String型の比較1

```
void start(){
    String a = "Hel";
    String b = "lo";
    if(a + b == "Hello"){
        System.out.println("同じ");
    }
}
```

上のコードを打ってみましょう。

結果はどうなるでしょう？

String型の比較2

- ◎ どちらも”Hello”なのにif文の中身が実行されません。
- ◎ Stringで「==」を使うと、中身の文字列ではなく文字列が保存されている場所(アドレス)を比較してしまいます。

※ちなみに、”(文字列)”==“(文字列)”は使えるが、不安定なので、非推奨。

String型の比較3

```
void start(){
    String a = "Hel";
    String b = "lo";
    if((a + b).equals("Hello")){
        System.out.println("同じ");
    }
}
```

文字列比較には「<文字列>.equals(文字列)」を使います。

上のようになるとif文の中身が実行されます。

繰り返し文 (for文)

- for文は、ある変数がある値からある値までの回数分繰り返す機能を持つ。
- for文は、繰り返す回数分かっているときに使う。
- for((始める値);(継続条件);(値の加算)) {(繰り返したい命令)}で実現できる。

例 : for(int i=0;i<5;i+=1){...}

⇒ 新規定数*i*を生成、*i*が0からスタート、
{ }内の命令が一通り終わったら、*i*に1を加算、
*i*が5より少ないなら、継続。
∴ *i*が0～4のときの計5回繰り返す命令。

具体例

```
int total = 0;
for (int i = 0; i < 5 ; i ++){
    total = total + i;
    JOptionPane.showMessageDialog(null,total);
}
```

⇒ 0 + 0 の結果表示、0 + 1 の結果表示、
1 + 2 の結果表示、3 + 3 の結果表示、
6 + 4 の結果表示を順番にしている。

ちなみに、i++ と i+=1 は、同じ意味！

繰り返し文 (for文)

`total = total + i;`

の部分は

`total += i;`

と書くことも出来ます。

もちろん「`-=`」 「`*=`」 「`/=`」 「`%=`」 も使えます。

例題

- ◎ 復習で作った、「入力した整数を5乗した数を示す」プログラムを、for文を使って書き換えてみましょう
- ◎ 「Example01」クラスの内容を書き換えましょう

繰り返し文 (while文)

- ◎ while文は、ある条件を満たすまで永遠に処理を繰り返す機能を持つ。
「無限ループってこわいよね...」
- ◎ while文は繰り返す回数が分からないときに使われるので多くの場面で用いられる。
- ◎ while((条件)){ ... }ででき、(条件)を満たすまで{...}内の命令を繰り返す。
- ◎ 条件をtrue(真)にすると無限ループが完成する。
* 脱出にはbreak;を使う。

while文の例(条件編)

```
例 : int i = 1;  
      int total = 0;  
      while(i < 5){  
  
          total = total + i;  
          i++;  
  
      }  
      JOptionPane.showMessageDialog( null,total);
```

※0 + 1、1 + 2、3 + 3、6 + 4 を内部で
やり、6 + 4の結果である **10** が出力される。

無限ループになったら・・・

コンソールを開き、赤い■をクリック。

The screenshot shows an IDE with a Java file named `Test.java`. The code contains a `while(true)` loop that prompts the user for input. A dialog box titled "入力" (Input) is displayed, asking for a number to end the program (0). The console window at the bottom shows the command prompt output: `Test [Java アプリケーション] C:\Program Files (x86)\Java\jre6\bin\javaw.exe (2012/04/12 18:12:51)`. A red square icon in the console window's toolbar is highlighted with a purple arrow, indicating where to click to stop the program.

ここをクリック！

例題

- ◎ 「Example03」のクラスを作ります
- ◎ while文を使って「何か整数を入力させてダイアログで表示」を繰り返すプログラムをつくってみましょう。
- ◎ 「-1」が入力されたら、終了するようにしましょう

擬似乱数 (Math.random())

- 擬似乱数とは、でたらめに数字を出すことであり、これを用いることでサイコロ振るなどのプログラムを作ることができる。
- Math.random()は、0.0以上1.0未満の数字を適当に出す命令です。

擬似乱数の例

```
void start(){  
    double random = Math.random() * 6;  
    int dice = (int) random + 1;  
    JOptionPane.showMessageDialog(null,dice);  
}
```

× 6 することによって、0 以上 6 未満の数字を出している。ただ、サイコロは 1 ~ 6 の出目なので + 1 をしている。(int) で小数点以下を切り捨てている。

演習問題1 (さいころ)

- ◎ 2人でさいころ (乱数) を投げ、どちらが勝ったのかを判定するプログラムを作りましょう。
- ◎ 同じ目の場合は、再度振らせ、勝敗が決定するまで繰り返してください。

演習問題2 (じゃんけん)

- プレイヤーに**数字入力**によってじゃんけんで出すものを数値で選ばさせ、**コンピュータ (乱数)** と勝負させる。
- あいこの場合は、**再度選ばさせ、勝敗がつくまで行う。**
- できたら、「**3勝するまで続ける**」等の改変をやってみましょう。

余談・・・

- ◎ 自分がゲーの時間が完成したら、チョコキ、パーのときも、コピペで追加して出来るはずです。
- ◎ これ以外に、もっと分かりやすい・簡単な書き方もできると思います。
- ◎ 自分なりに考えて書いてみると更にいいですね！！

次回について

- ◎ 前ページの例でプログラム組むと、

結構長いですよね

そして**見づらい****分かりづらい**

そんな見づらさを解消する

メソッド

について次回は学びます

次は4/30(月)ですよ！