

# java講座第1回 前半

情報科学部CS学科 2年網谷拓海

# 今回の講義の内容

- 前半

- ひな形
- 入出力
- 変数
- 演算子
- if文(条件分岐)
- for文(繰り返し)

- 後半(別スライド)

- while文(繰り返し)
- void型メソッド

# Eclipseの準備

- 「プロジェクト・エクスプローラー」枠内を右クリック、「新規」→「プロジェクト」
- 「Java」ダブルクリック→「Javaプロジェクト」ダブルクリック
- プロジェクト名を入力してEnter
- 作成したプロジェクトをダブルクリック→srcファイル右クリック→  
「新規」→「パッケージ」
- パッケージの名前を入力してEnter
- srcファイル内にパッケージがあるので右クリック  
「新規」→「クラス」
- クラス名を入力してEnter

※「プロジェクト・エクスプローラー」がウィンドウ内にはない場合  
「ウィンドウ」→「ビューの表示」→「プロジェクト・エクスプローラー」

# ひな形

```
ClassName.java ✖
1 package j1.lesson01;
2
3 public class ClassName {
4     public static void main(String[] args) {
5         new ClassName().start();
6     }
7
8     void start() {
9
10    }
11 }
12
```

- ClassNameと書かれている部分にクラス名を書く(2か所あります)
- これから書く命令はvoid start(){...}の中カッコ内部に書く
- カッコ内に書かれた命令は上から順番に実行される
- 1つの命令は、書き始めから;まで

# 出力

- System.out.println();

```
1 package j1.lesson01;
2
3 public class ClassName {
4     public static void main(String[] args){
5         new ClassName().start();
6     }
7     void start(){
8         System.out.println("Hello!!");
9         System.out.println(196184);
10        System.out.println(-1.1);
11        System.out.println("私は森進一です。" + 66 + "歳です");
12    }
13 }
```

コンソール

<終了> ClassName [Java アプリケーション] C:\Software\Java\jdk-1.7.0\_25\bin\javaw.exe (2014/05/10 22:12:09)

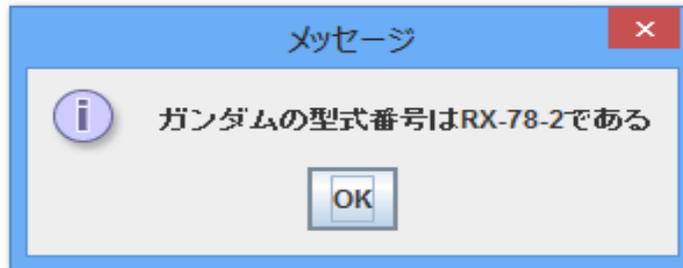
Hello!!  
196184  
-1.1  
私は森進一です。歳は66歳です

- ()内に入力した文字列( "" で囲むと文字列として判断される)、整数、少数を改行してコンソール内に出力する
- 文字列と他の文字列、整数、少数を合わせて出力したければ、+でつなぐ

# 出力

- JOptionPane.showMessageDialog(null, )

```
1 package j1.lesson01;
2
3 import javax.swing.JOptionPane;
4
5 public class ClassName {
6     public static void main(String[] args){
7         new ClassName().start();
8     }
9     void start(){
10        JOptionPane.showMessageDialog(null, "ガンダムの型式番号はRX-" + 78 + "-" + 2 + "である");
11    }
12 }
13
```

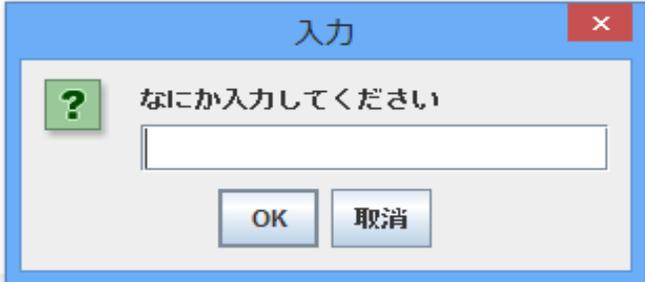


- ()内のnull,の横に入力した文字列、整数、少数をメッセージダイアログで出力する
- 入力の仕方はSystem.out.println()と同様
- 追加条件として、public class ClassName{...}の外側に  
import javax.swing.JOptionPane;の1文を書かなければいけない

# 入力

- JOptionPane.showInputDialog()

```
1 package j1.lesson01;
2
3 import javax.swing.JOptionPane;
4
5 public class ClassName {
6     public static void main(String[] args){
7         new ClassName().start();
8     }
9     void start(){
10        JOptionPane.showMessageDialog(null,
11            JOptionPane.showInputDialog("なにか入力してください"));
12        System.out.println(
13            JOptionPane.showInputDialog("入力しろ"));
14    }
15 }
```



- ユーザーに入力してもらったものを「文字列」として返す
- ()内に文字列、整数、少数を入力するとユーザーへの質問として入力ダイアログに表示させることができる
- 例ではこの命令をJOption.showMessageDialog(null,)とSystem.out.println()の入力として利用してみた  
Q. このプログラムがどのように動作するか考えてみよう

# 変数

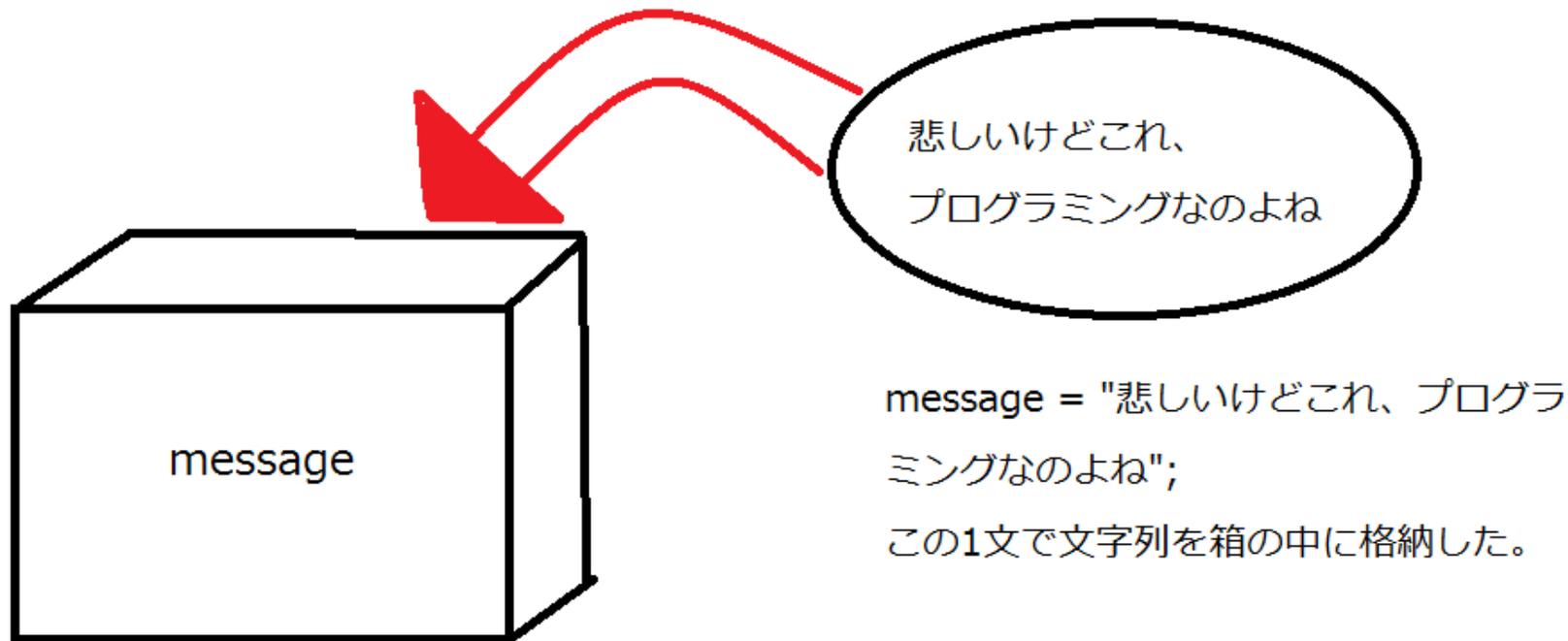
```
1 package j1.lesson01;
2
3 public class ClassName {
4     public static void main(String[] args) {
5         new ClassName().start();
6     }
7     void start() {
8         String message;
9         message = "悲しいけどこれ、プログラミングなのよね";
10        System.out.println(message);
11    }
12 }
```

コンソール

<終了> ClassName [Java アプリケーション] C:\Software\Java\jdk-1.7.0\_25\bin\javaw.exe (2014/05/11 13:03:02)  
悲しいけどこれ、プログラミングなのよね

- ・型名 変数名; この1文で変数を宣言できる(イメージ:データを入れる箱をつくる)。
  - ・変数名 = データ; この1文で変数に値を代入できる(イメージ:箱にデータをいれた)
- 例では型名をStringとした。これは文字列の型名である。  
文字列の変数だから、System.out.println()に変数をはめても動作するわけである。

## 変数のイメージ



```
String message;
```

この1文でmessageという箱ができた。

この箱は文字列を格納できる。

String message = “文字列”; と1つの文にまとめることも可

## • 変数の型

- int                    整数の型 (ex. -1)
- double                少数の型 (ex. 3.14)

Q. JavaにはString, int, double以外にも型が用意されている. どんなものがあるか調べてみよう.

## • 型の変換

- Integer.parseInt()
  - Double.parseDouble()
    - “-1” や “3.14” は“”で囲まれているので文字列であるが、“-1”はInteger.parseInt() の “3.14”はDouble.parseDouble() の()内にいれるとそれぞれ型が整数から文字列、少数から文字列に変換される。
- 入力のJOptionPane.showInputDialog()は「文字列」で入力を返してくるので、整数や少数として入力を取得できないが、この変換を利用すればユーザーから受け取ったデータを整数や少数として扱うことができる。

# 演算子

## 算術演算子

- + 加算(足し算)
- - 減算(引き算)
- \* 乗算(掛け算)
- / 除算(割り算)
- % 剰余(割り算の余り)

## 代入演算子

▪ +=, -=, \*=, /=, %=

a+=100だったら、 $a = a + 100$ の意味 そのほかも同様

## インクリメント・デクリメント演算子

▪ ++ --

a++だったら、 $a = a + 1$ の意味 --も同様 \*\*, //, %%はない

## 着目してほしい点

```
1 package j1.lesson01;
2
3 public class ClassName {
4     public static void main(String[] args) {
5         new ClassName().start();
6     }
7     void start() {
8         String inputA = "100";
9         String inputB = "51";
10        int a = Integer.parseInt(inputA);
11        int b = Integer.parseInt(inputB);
12        int c = a + b;
13        System.out.println(c);
14        System.out.println(a * b);
15        System.out.println("a / b = " + a / b);
16        System.out.println(b % a);
17    }
18 }
```

☰ コンソール ☒

<終了> ClassName [Java アプリケーション] C:\Software\Java\jdk-1.7.0\_25\bin\javaw.exe (2014/05/11 13:24:45)

```
151
5100
a / b = 1
51
```

- ・変数cの代入のように演算子を用いて、直接演算結果を代入できる

- ・System.out.println()の入力に、直接演算を書き込める

- ・文字列と演算結果を組み合わせて入力したい場合には、文字列と演算結果を組み合わせる + と演算子が混同してしまわないように演算する部分は()で囲む

- ・100 / 51の結果が1になっている (intは少数部分を切り捨てるため)

- ・割り算の余りはどのような値になっているか

## if文(条件分岐)

- if( 条件 ){...}
  - ()内の条件が満たされた場合は{}内の命令を行い、条件が満たされていない場合は{}内の命令は行われない
- if( 条件 ){A} else {B}
  - ()内の条件が満たされた場合は{A}内の命令を行い、条件が満たされていない場合は{B}の命令を行う
- if( 条件A ){A} else if( 条件B ){B} else if( 条件C ){C} else {D}
  - まず、1番初めの条件Aが満たされるかどうか確認し、満たされていれば命令Aを行い、そうでなければ、次の条件Bを確認し、そこも満たされなければ条件C、条件Cも満たされなければ命令Dが行われる

# 条件式の書き方

- 条件式は「関係演算子」をもちいて書く
  - == 左辺と右辺が等しい(=と間違えないように!)
  - != 左辺と右辺が異なる(!は否定の意味で使う)
  - > 左辺が右辺より大きい
  - < 左辺が右辺より小さい
  - >= 左辺が右辺より大きいか等しい
  - <= 左辺が右辺より小さいか等しい
- 2つ以上の条件を組み合わせるには「論理演算子」
  - && 「かつ」の意味 両方の条件が満たされなければいけない
  - || 「または」の意味 どちらかの条件が満たされればよい

Q. `!((A < B) && (C >= D))`はどのような条件か考えてみよう

- 文字列の比較
  - `A.equals(B)` 文字列Aが文字列Bと等しいか

```
1 package j1.lesson01;
2
3 public class ClassName {
4     public static void main(String[] args) {
5         new ClassName().start();
6     }
7     void start() {
8         int a = 100;
9         int b = 120;
10        int c = 2000;
11        int d = -1;
12        String messageA = "おはようございます";
13        String messageB = "こんにちは";
14
15        if(a >= b){
16            System.out.println(messageA);
17        }else if((b < c) && (c == d)){
18            System.out.println(messageB);
19        }else if(a % 15 == 0){
20            System.out.println("/パリポリ");
21        }else{
22            System.out.println("さようなら");
23        }
24    }
25 }
```

Q. コンソールにはどのような文字列が出力されるか考えてみよう

# for文(繰り返し)

```
1 package j1.lesson01;
2
3 public class ClassName {
4     public static void main(String[] args) {
5         new ClassName().start();
6     }
7     void start() {
8         for(int i = 0; i < 5; i++){
9             System.out.println(i);
10        }
11    }
12 }
```

コンソール

<終了> ClassName [Java アプリケーション] C:\Software\Java\jdk-1.7.0\_25\bin\javaw.exe (2014/05/11 1:04:21)

0  
1  
2  
3  
4

- System.outprintln()は1回しか書いてないのにコンソールには5回出力されたなぜ？

for(初期化処理; 繰り返し条件; 繰り返し時処理){...}

1. 初期化処理      繰り返しの最初に一回だけ呼ばれる処理  
ここで、「いま何週目の繰り返しかを記録する変数」を定義する。この変数をループ変数という
2. 繰り返し条件      ループ変数がこの条件を満たしていれば、  
{...}内の命令が行われる
3. 繰り返し時処理      {...}内の命令が最後まで行われたときに行われる処理  
通常は*i++*のようにループ変数を1増やす処理を書く



## • 演習1

- ユーザーに2つの整数を入力してもらい、それらの数を足した結果をコンソールに6回出力せよ。ただしfor文を用いること。
  - ヒント: for文のページのプログラムは5回出力していた。どこを変えれば6回になるか考えよう

## • 演習2

- ユーザーに100以上の整数を入力してもらい、その百の位、十の位、一の位の数をコンソールに出力せよ。
  - ヒント: 割り算と、余りの演算子を用いる。34103を1万で割った余りを1000で割ると4.103になる。これの少数部分を切り捨てると34103の千の位の数になる。少数部分を切り捨てる型は何であったか考えよう

## • 演習3

- 1から100の数をコンソールに出力せよ。ただし3の倍数の時は「パリ」、5の倍数の時は「ポリ」、15の倍数の時は「パリポリ」と出力するようにせよ。
  - ヒント: for文とif文を組み合わせる。for文のページのプログラムでは0から4の数をコンソールに出力させた。これを参考にすれば1から100までの数を出力するのは容易だろう。倍数の判定には割り算の余りを用いる。3の倍数ということは3で割ると余りは？複数の条件分岐には `else if(){...}` を用いるのがいいだろう。

## 参考文献

- ・中山清喬/国本大吾 著  
2011年出版「スッキリわかるJava入門」p40 - p133  
インプレスジャパン発行