Java講座 第5回
やる夫(CV:ゆっくり)と学ぶ
配列・クラス編

情報科学部 DM 2年
海藤優弥

出典一覧

- あのAAどこ？
  http://dokoaa.com/yaruo.html
- だからニュー速でやるお！のガイドラインまとめ
  http://yaruo.xxxxxxxx.jp/aa/yaranaio.html
- オタク.com -オタコム-
  http://Otaku.livedoor.biz/archives/3972906.html
- ニコニコ大百科(仮)
  http://dic.nicovideo.jp/a/%E3%81%AC%E3%82%8B%E3%81%BD
- 達人プログラマーを目指して
  http://d.hatena.ne.jp/ryoasai/20111217/1324139249
- プログラミング入門1 オンライン教材
  http://java2010.cis.k.hosei.ac.jp/10-2/material-10/
- 音声合成ソフト・・・SofTalk
  http://www35.atwiki.jp/softalk/
- Puella Magi Wiki・・・フォント
  http://wiki.puella-magi.net/Deciphering_the_runes

人物紹介

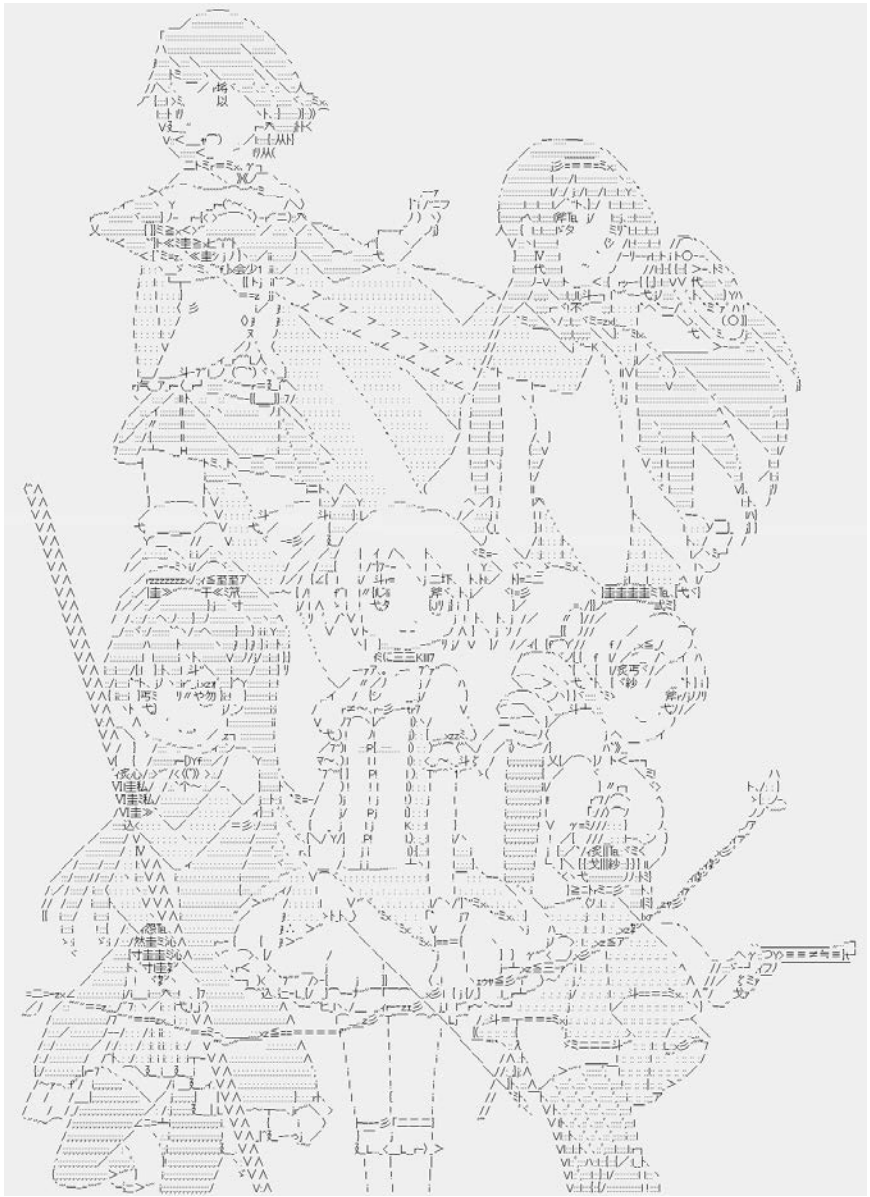
やる夫

- 法政大学 情報科学部 1年
- 計算技術研究会所属
- プログラミング勉強中
- (ry

やらない夫

- 法政大学 情報科学部 1年
- 計算技術研究会所属
- FBIのサーバーに侵入したこともあるスーパーハカー
- (ry

スペシャルゲスト



- 某魔法少女の皆さん
- まどマギの再放送を見ながらのスライド作成のため登場
- どうやら彼女らもプログラミングを勉強中らしい

配列編

どの言語にもある配列という機能
これを知らずにプログラムを作るのは、多分不可能
でも、つまずく人が多いらしい？

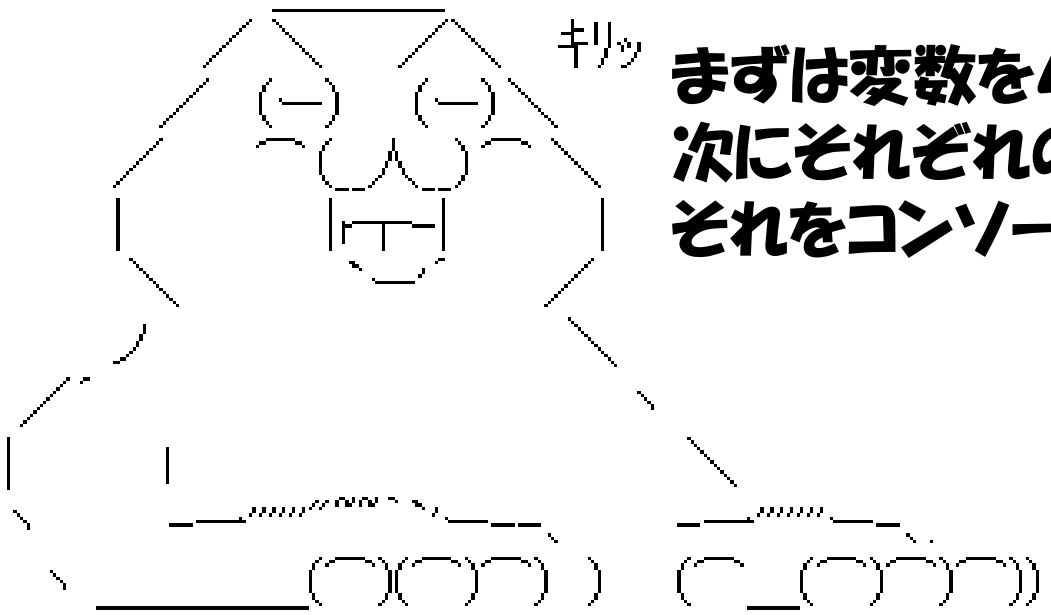
変数、条件分岐、メソッドなどを
順調にクリアしたやる夫

プログラミングなんて簡単だお！キリッ
これならやらない夫のようなスーパーハカーになる日もそう遠くないお

今日もプログラミングの勉強をするお！

問. 40人のクラスの中間テストの結果を表示するプログラムを作りなさい

こんな問題、今のやる夫にかかれば朝飯前だお



まずは変数を40個用意するお
次にそれぞれの得点を変数に代入するお
それをコンソールに出力すれば完成だお！

```
int score_01 = 80;
int score_02 = 75;
int score_03 = 90;
int score_04 = 50;
    .
    .
    .
```

自分の才能が憎いお
やらない夫にも見せに行くお

最近頑張って勉強してるみたいだな
どれどれ・・・

新しいプログラムが完成したから、見てほしいお

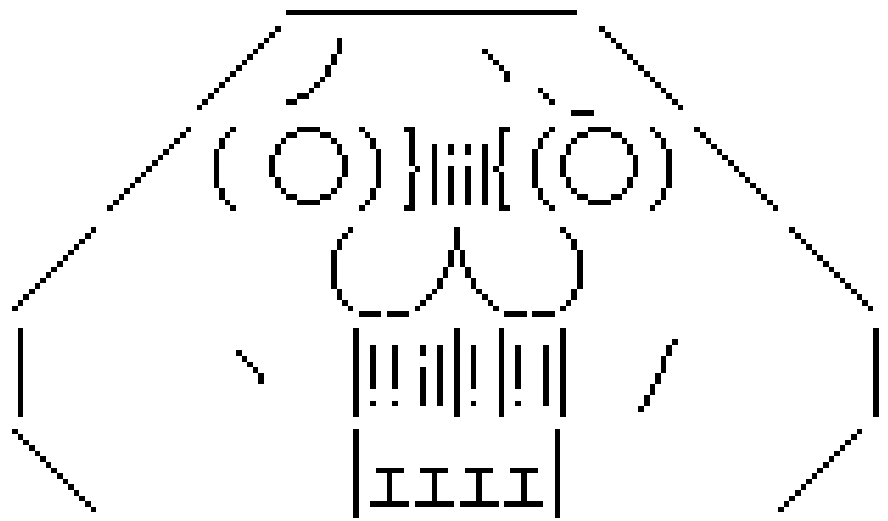
やる夫、これ全部打ち込んだのか・・・
お疲れ様だな

・・・でも、実はもっと簡単なやり方があるんだ

そうだったのかお・・・

まさかスーパーハカーのみぞ知る秘策なのかお？

早くやる夫にも教えろお！！！



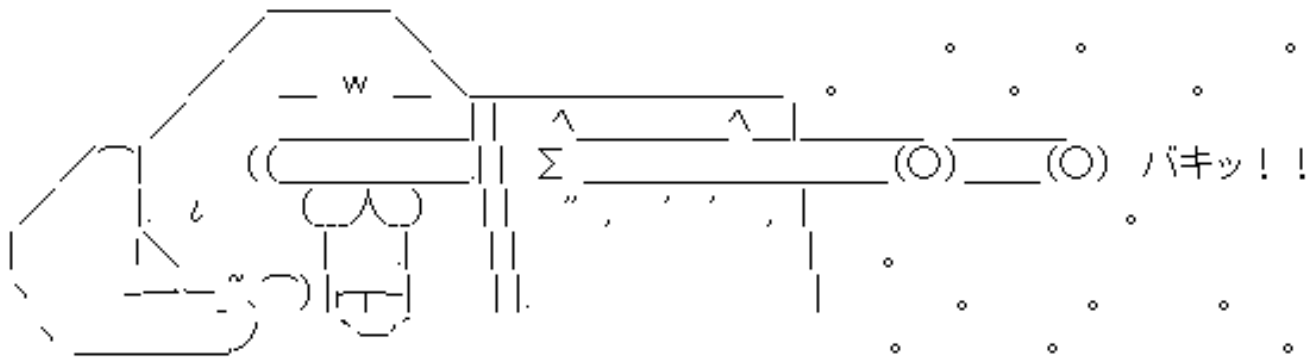
いや、そんなたいそうなものではないぞ
むしろどの言語にもある基本的なものだ

やる夫は生徒の得点を管理するために40個の変数を使っただろ？

別にそれでも構わないんだが、プログラムを書くときには生徒の得点といった、似ているものは「**配列**」というものを使ってまとめて管理するのが普通なんだ

口で説明しづらいから、Java 配列でググれ

こんなシークレット機能があったのかお！！！！！



シークレット機能なわけないだろjk

プログラムを書くうえで、配列を知らないと
話にならないから覚えておいた方がいいぞ

配列

- 似たような変数を一括りで管理するための機能
- 普通の変数と同じように扱うことが出来るよ！

- 記述方法

  〈型名〉[] 〈変数名〉 = new 〈型名〉[〈長さ〉];

  - int[] array = new int[10];
  - double[] array = new double[10];
  - String[] array = new String[10];

配列

- 配列の値の参照

  〈変数名〉[〈インデックス〉]

  - int[] value = array[5];
  - System.out.println(array[5]);

- インデックスについての注意

  配列のインデックスは0から始まることに注意！

  int[] array = new int[10];

  array[0] ～ array[9];

問題

- 長さが10のint型の配列を生成して、それぞれにインデックスと同じ整数を代入せよ
- 上の配列の0から9を順番に出力せよ
  (JOptionPaneで1個づつ出力してね☆彡)

- ✓ for文を使うと簡単かもね～
- ✓ 配列の長さは、直接入力してもいいけど、array.lengthで配列の長さが取得可能！

簡単杉ワロスって人は、逆順(9から0)で配列を出力とかやってみてね☆彡

補足

- String str = “やる夫,やらない夫”;

この文字列から、やる夫・やらない夫という2つの文字列を抜き出したい場合、どうしたらいいか

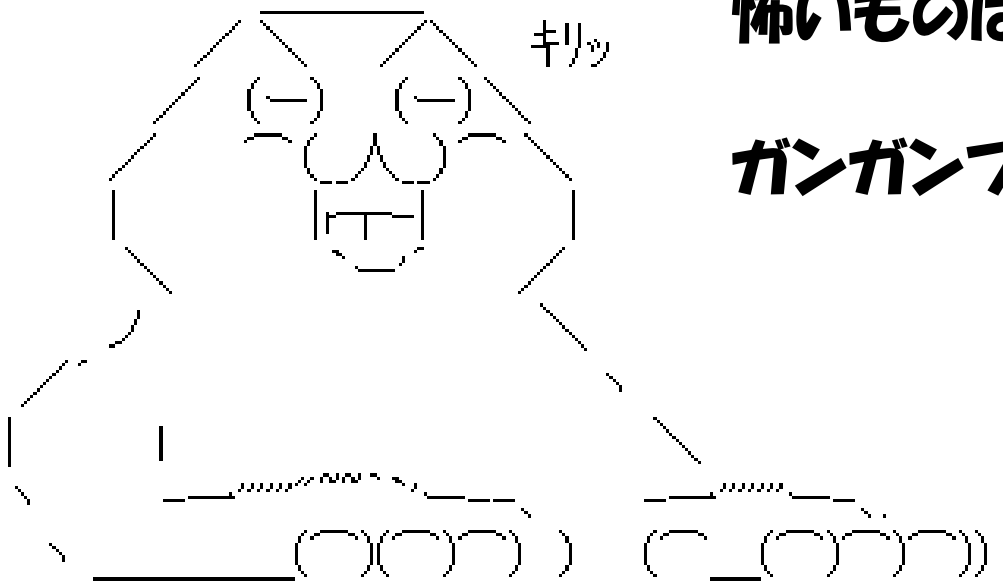
- String[] array = str.split(“,”);

- split()は引数に指定した文字で文字列を分割する
- split()の戻り値はString型の配列
- 区切り文字は、ほかの文字でもおk
- Eclipseで実例をやりまーす

クラス編

オブジェクト指向という概念の基本となるクラスだが、
情報科学部は2年になってからクラスを扱うというクソ仕様

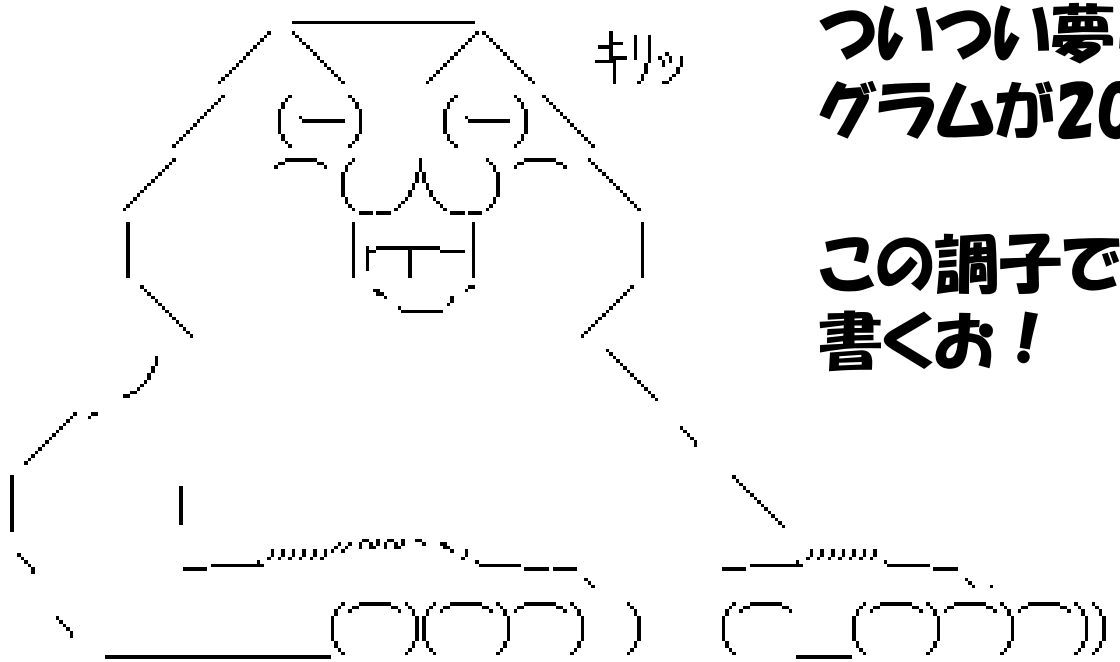
配列をマスターしたやる夫



配列をマスターしたやる夫にもう
怖いものはないお

ガンガンプログラムを書くお

```
2239 ¤¶
2240 »   public·boolean·onTouchEvent(MotionE
2241 »   »   switch·(scene)·{¤¶
2242 »   »   //·初期画面¤¶
2243 »   »   case·0:¤¶
```

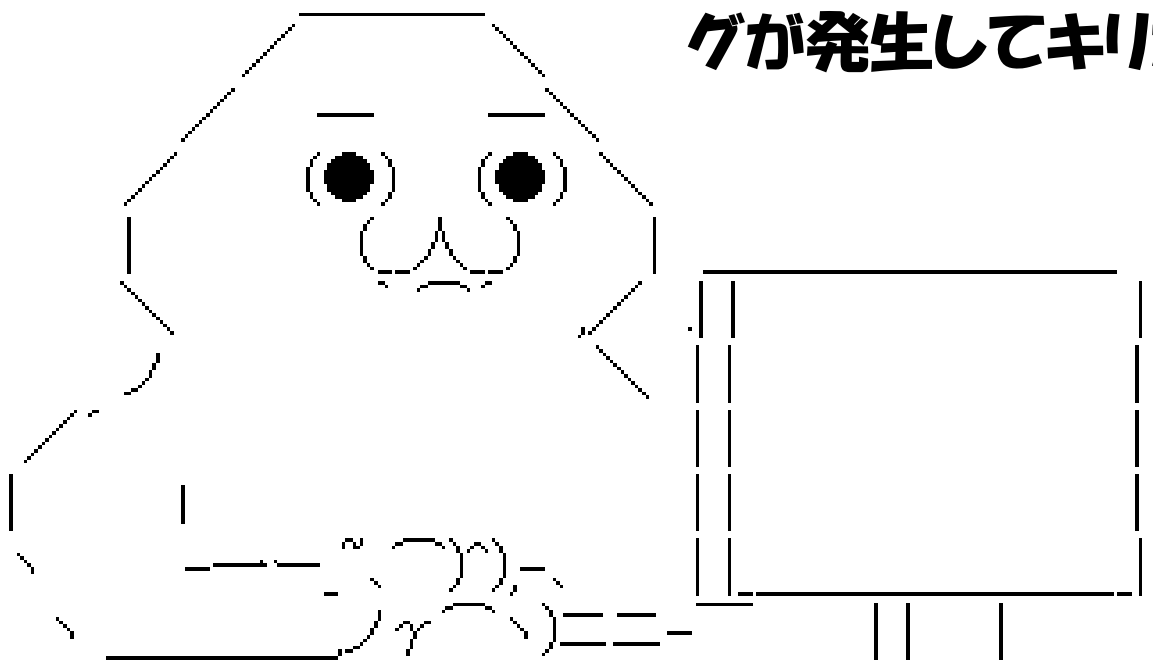


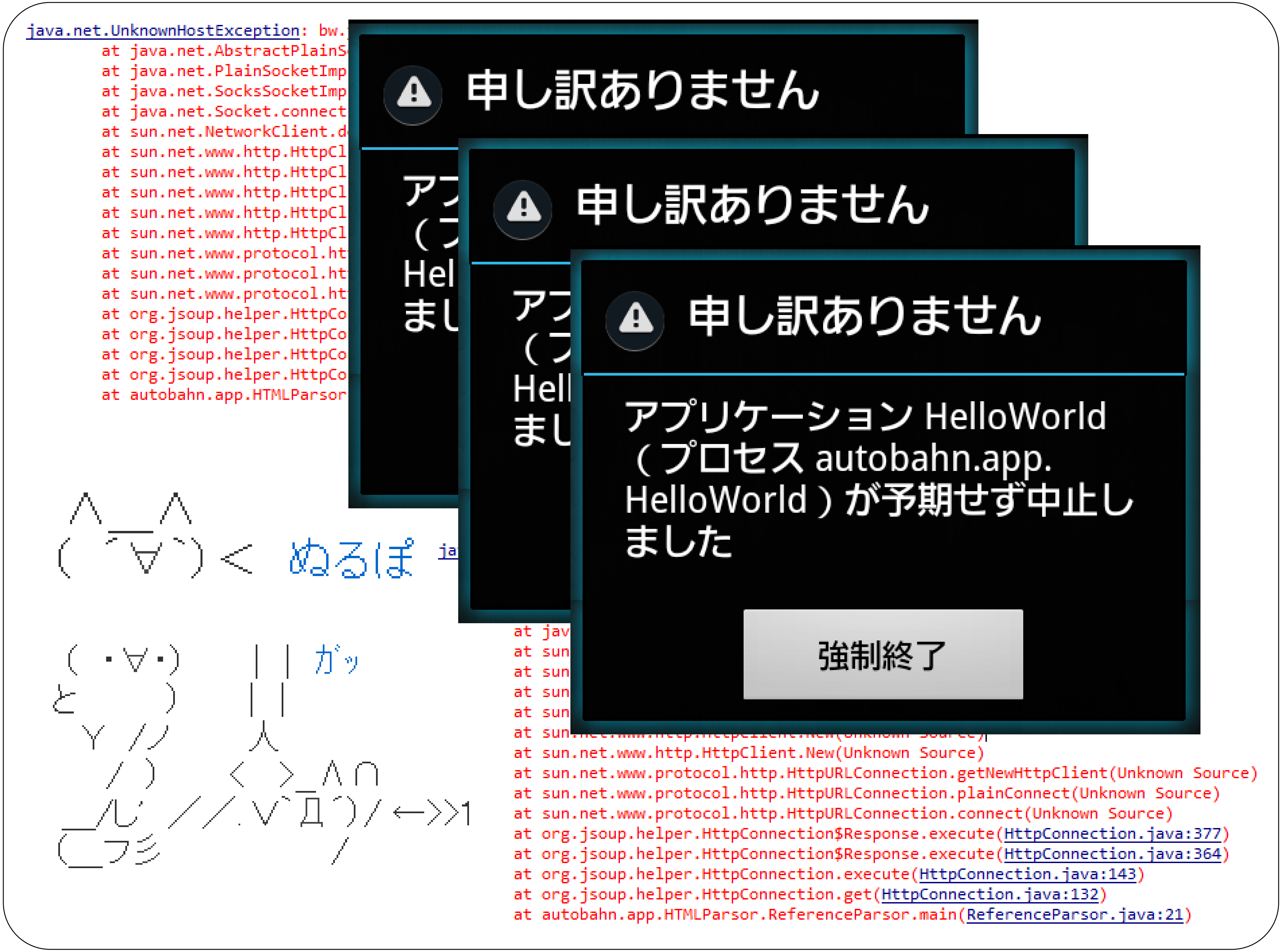
ついつい夢中になってるうちにプログラムが2000行を超えたお

この調子でもっともっとプログラムを書くお！

うーん、またバグかお・・・
一体何処を間違えたんだお

直しても直しても、また新しいバグが発生してキリがないお・・・

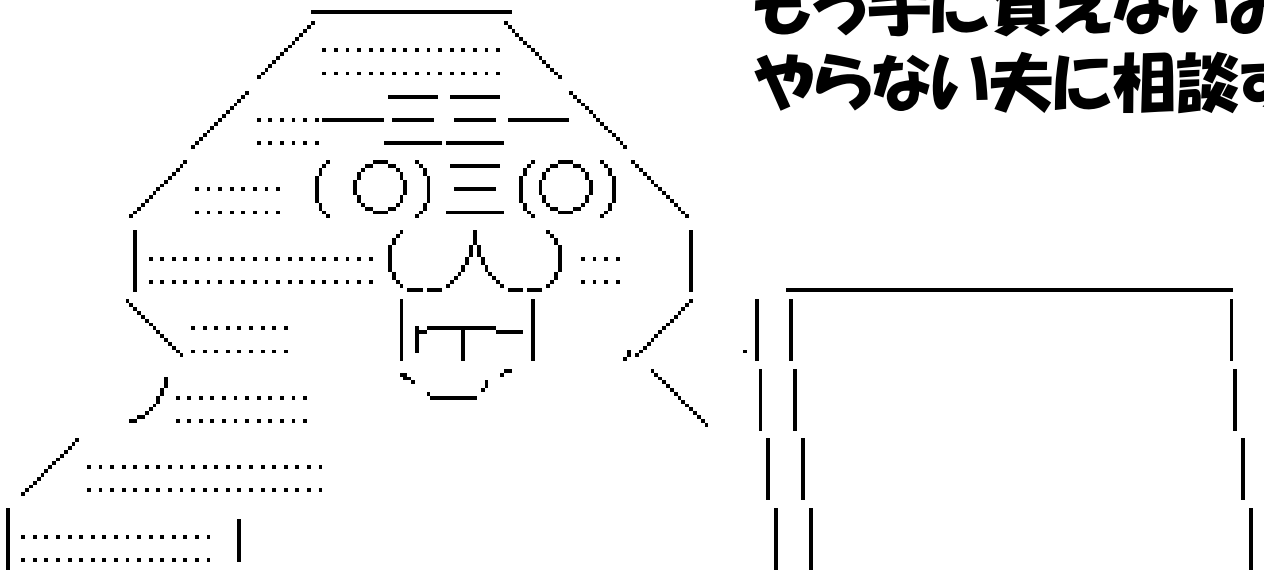




java.net.UnknownHostException: bw.
at java.net.AbstractPlainS
at java.net.PlainSocketImp
at java.net.SocksSocketImp
at java.net.Socket.connect
at sun.net.NetworkClient.d
at sun.net.www.http.HttpCl
at sun.net.www.http.HttpCl
at sun.net.www.http.HttpCl
at sun.net.www.http.HttpCl
at sun.net.www.http.HttpCl
at sun.net.www.protocol.ht
at sun.net.www.protocol.ht
at sun.net.www.protocol.ht
at org.jsoup.helper.HttpCo
at org.jsoup.helper.HttpCo
at org.jsoup.helper.HttpCo
at org.jsoup.helper.HttpCo
at autobahn.app.HTMLParsor
申し訳ありません
申し訳ありません
申し訳ありません
アプリケーション HelloWorld（プロセス autobahn.app.HelloWorld）が予期せず中止しました
強制終了
ぬるぽ
ガッ
at sun.net.www.http.HttpClient.New(Unknown Source)
at sun.net.www.protocol.http.HttpURLConnection.getNewHttpClient(Unknown Source)
at sun.net.www.protocol.http.HttpURLConnection.plainConnect(Unknown Source)
at sun.net.www.protocol.http.HttpURLConnection.connect(Unknown Source)
at org.jsoup.helper.HttpConnection$Response.execute(HttpConnection.java:377)
at org.jsoup.helper.HttpConnection$Response.execute(HttpConnection.java:364)
at org.jsoup.helper.HttpConnection.execute(HttpConnection.java:143)
at org.jsoup.helper.HttpConnection.get(HttpConnection.java:132)
at autobahn.app.HTMLParsor.ReferenceParsor.main(ReferenceParsor.java:21)

うわあああああああああああああ
あああああああああああああああ
あああああああああああああああ

こりゃどうなってるんだお・・・
もう手に負えないお・・・
やらない夫に相談するお



どうしたんだ？

ちょっと、助けてほしいお・・・

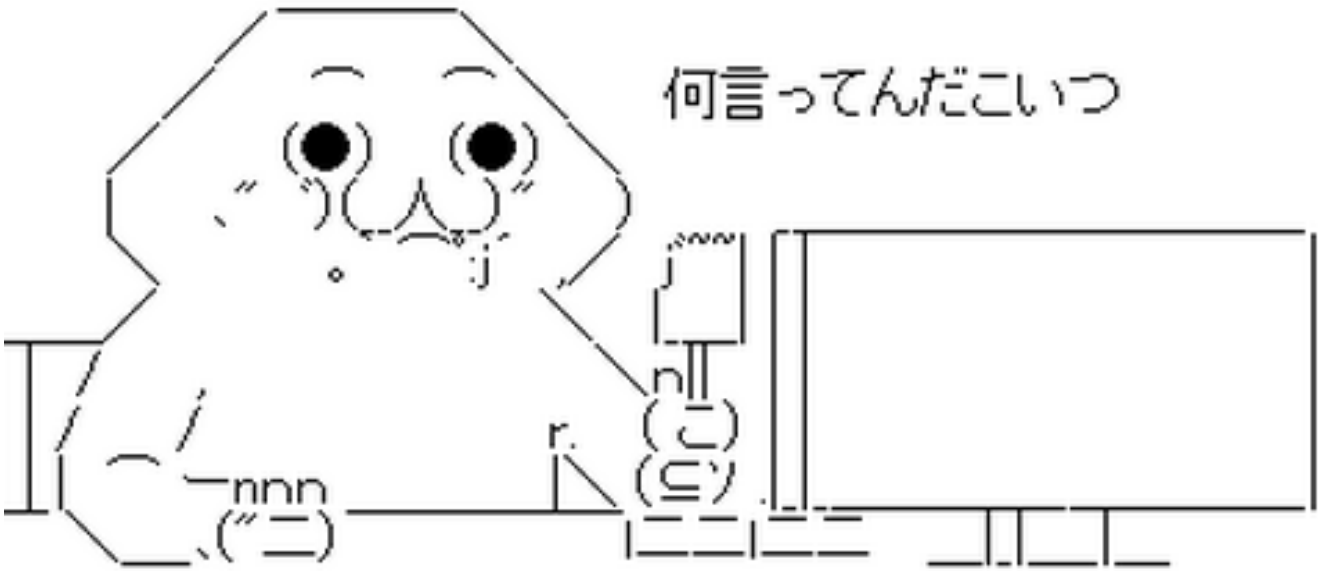
どれどれ・・・

プログラムのバグが直らないお・・・
ちょっと見てほしいお

うーん、どこからツッコめばいいのやら・・・

とりあえず、プログラムの書き方に問題があるな

このメソッドとこのメソッドをクラス①に、この変数をクラス②のフィールドとして、（ry


何言ってんだこいつ

クラス

情報科学部1年の授業では、ほとんどクラスは扱われない

↓

でも、Javaで開発をする上で知っておくべき

↓

じゃあ、解説するしかないじゃない！！！

↓

でも、全部解説しきれないので、授業分＋αみたいな感じ
もっと解説して欲しい人がいれば、追加授業も(俺が)？
それか、夏休みに集中講座的な感じ？

クラス

- 1つのモノ(オブジェクト)を表すJavaにおけるファイル単位
- そのモノが持つステータス(変数)などを1つのまとまりとして定義することが出来る
- 当然メソッドも定義出来る
- クルマというオブジェクトを考えた場合
    - 車種名(変数)
    - カラー(変数)
    - 前進(メソッド)
    - ストップ(メソッド)

✓ ちなみに、これらはメンバとか呼ばれている(たぶん)

✓ 変数はフィールドとも呼ばれる(たぶん)

クラス

- 大きなプログラムを作る際は機能ごとにクラスで分割して作るのが望ましい

- RPGを作る場合
  - プレイヤークラス
  - 敵モンスタークラス
  - 画面の描画を行うクラス
  - データベースを扱うためのクラス
  - 内部処理を行うクラス

✓ これはあくまで一例であって、他の設計でも勿論OK

クラス

- なんで、クラスで分割するんですか？

1. 共通のロジックを流用しやすくする（再利用）
2. 大規模なプログラムを分割して扱い易くする（モジュール化）
3. 変化や拡張に強いプログラムにする（拡張性）
4. 設計のアイデアを共有する（パターン）
5. ビジネスルールをプログラムとしてわかりやすい形で表現する（モデル化）

要は、読みやすいし、バグが見つけやすいってことだと思う

クラス

- 記述方法

```
public class Player() {
        String name = “やる夫”;
        int hp;                          ←メンバ変数(フィールド)
        int mp;

        String getName() {
                return name;             ←メンバメソッド
        }
}
```

インスタンス

- クラスは設計図のようなものなので、実際にプログラム内でクラスを使うには、インスタンスというものを使う
- インスタンスはクラスの実体を表している

- インスタンスの生成

  〈クラス名〉〈変数名〉 = new 〈クラス名〉();

  - Player player = new Player();
  - Parson hogehoge = new Parson();

クラス

- クラスのメンバへのアクセス
  〈インスタンス変数名〉.〈メンバ変数名〉
  〈インスタンス変数名〉.〈メンバメソッド名〉

  - player.name = “やる夫”;
  - System.out.println(player.getName());

問題

1. 実行クラス(public static void main()があるクラス)とは別に人間を表すParsonクラスを作りましょう
2. メンバ変数として、name、ageを宣言しましょう
3. メンバメソッドとして、showInfo()というJOptionPaneで情報を表示するメソッドを作りましょう
4. 実行クラスからname、ageに自分の名前、年齢を代入しましょう
5. 実行クラスからshowInfo()を呼び出してみましょう

- ✓メンバ変数へのアクセスは、parson.nameだよ！
- ✓メンバメソッドへのアクセスは、parson.showInfo()だよ！

余談

ここからは、ちょっと発展的なお話
でも、今日知ってもらいたいのは実はここからだったり
クラス、配列はプログラムを書いていれば誰でも覚えます

- プリミティブ型とクラス型
- 多次元配列(2次元配列)

プリミティブ型とクラス型

普段は何気なく使っている変数の型に関するお話

プリミティブ型とクラス型

プリミティブ型

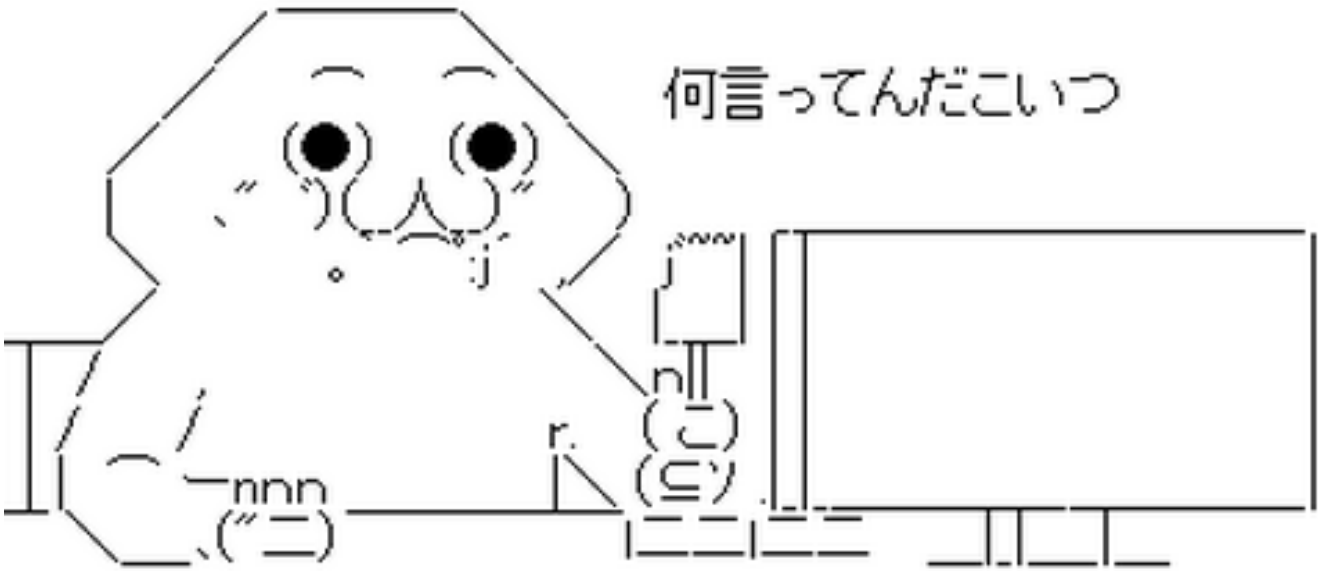
- int
- double
- boolean

✓変数に直接値が入っている

クラス型

- String
- Math
- 自作クラス(Parson等)

✓変数には値が格納されているメモリのアドレスが入っている


何言ってんだこいつ

プリミティブ型とクラス型

```
void start() {
    int value = 100;
    modify(value);
    System.out.println(value);
}

void modify(int value) {
    value = 200;
}
```

プリミティブ型とクラス型

```
void start() {
    int value = 100;
    modify(value);
    System.out.println(value);
}

void modify(int value) {
    value = 200;
}
```

正解は、100

プリミティブ型とクラス型

Q. なんで？

A. intはプリミティブ型だからです

- プリミティブ型は変数に値そのものが入っている
- start()からmodify()に値が渡された時点で、完全に別物になっているから
- modify()で値を変更してもstart()の値は変わらない

プリミティブ型とクラス型

```
void start() {
    Parson parson = new Parson();
    parson.name = "かいどうゆうや";
    parson.age = 19;
    modify(parson);
    parson.showInfo();
}
void modify(Parson parson) {
    parson.name = "やる夫";
}
```

プリミティブ型とクラス型

```
void start() {
    Parson parson = new Parson();
    parson.name = "かいどうゆうや";
    parson.age = 19;
    modify(parson);
    parson.showInfo();
}
void modify(Parson parson) {
    parson.name = "やる夫";
}
```

正解は、やる夫：19

プリミティブ型とクラス型

Q. うそつけ

A. 本当です

- Parsonはクラス型なので、変数parsonに値そのものは入ってません
- 変数parsonには、値が格納されているメモリ上のアドレスが入っていて、start()からmodify()には、そのアドレスが渡されています
- 1つのインスタンスを共有しているような感じ
- ちなみに、配列もクラス型と同じ振る舞いをします

プリミティブ型とクラス型

Q. アドレス、アドレスってお前は出会い厨か！

A. 違います

Q. じゃあ、アドレスって何よ？

A. 次のスライドで解説します

プリミティブ型とクラス型

```
void start() {
    Parson parson = new Parson();
    System.out.println(parson);
}
```

これを実行すると・・・

j2.lesson01.Parson@188edd79

これがアドレスです
詳しくは知りません（キリッ

プリミティブ型とクラス型

Q. その理屈だと、Stringはプリミティブ型じゃね？

A. いいえ、クラス型です

- Stringはプログラム内で頻繁に利用されるので、例外的にプリミティブ型と同様に使っていいことになっています
- 当然、String str = new String(“やる夫”);もおk
- これは、String str = “やる夫”;

- 大文字から始まっているものは基本的にクラス型と考えておk

多次元配列

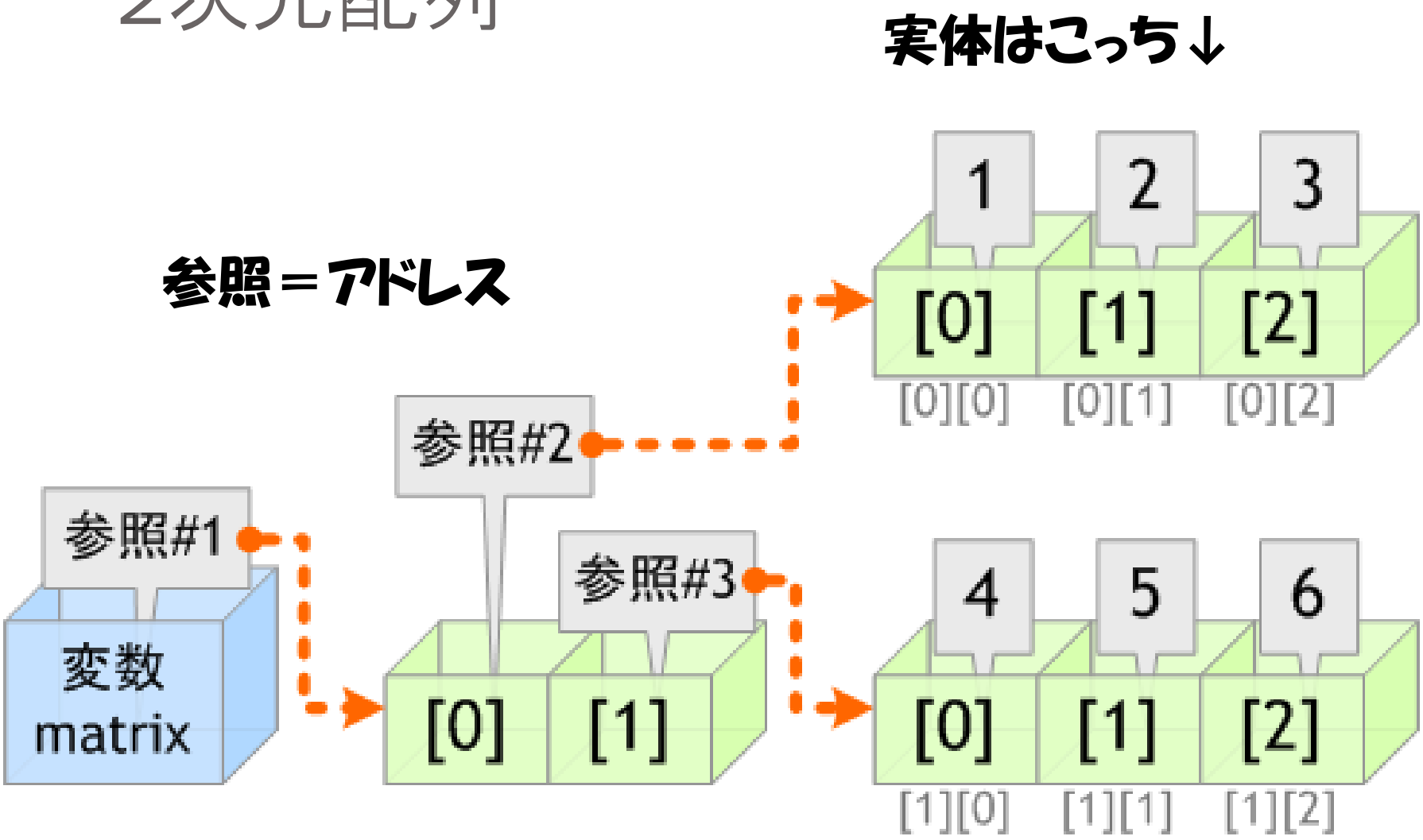
配列は何次元でも増やすことが出来ちゃいます

多次元配列

- 配列は1次元に限らず、2次元、3次元、4次元といった感じで、いくらでも増やせます
- ただ、3次元以上は管理が大変なので、あまりやらないのかな？

- 結論：2次元最高！！杏子ちゃん！！！

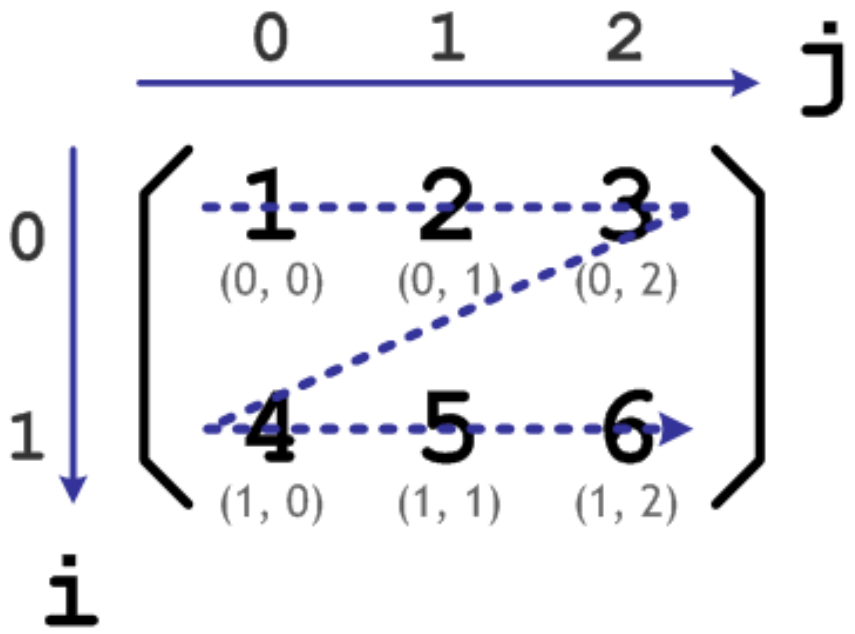
2次元配列


実体はこっち↓
参照＝アドレス
1
2
3
[0]
[1]
[2]
[0][0]
[0][1]
[0][2]
参照#2
参照#1
参照#3
変数
matrix
[0]
[1]
4
5
6
[0]
[1]
[2]
[1][0]
[1][1]
[1][2]

2次元配列

```
String[][] matrix = new String[2][3];
for (int i = 0; i < matrix.length; i++) {
        for (int j = 0; j < matrix[i].length; j++) {
                matrix[i][j] = "value";
        }
}
```

アイを固定した状態で、
ジェーを動かす



問題

- 時間があったらやる

1. String[][] matrix = new String[2][3];を宣言しよう
2. 行,列という形で番地のような文字列をそれぞれに代入しよう
3. それを

こんな感じでコンソールに出力しよう

- ✓ 改行のない出力はSystem.out.print();
- ✓ System.out.println();で、ただの改行になる
- ✓ 頭の中でプログラムがどういう動きになるか考えながら打ち込もう！これ大事！
- ✓ これが出来たら、プログラミング入門1でA+余裕

お疲れ様でした！！！

以上で今日の講座は終わりです

ご清聴ありがとうございました

少しでも役に立てたなら嬉しいです

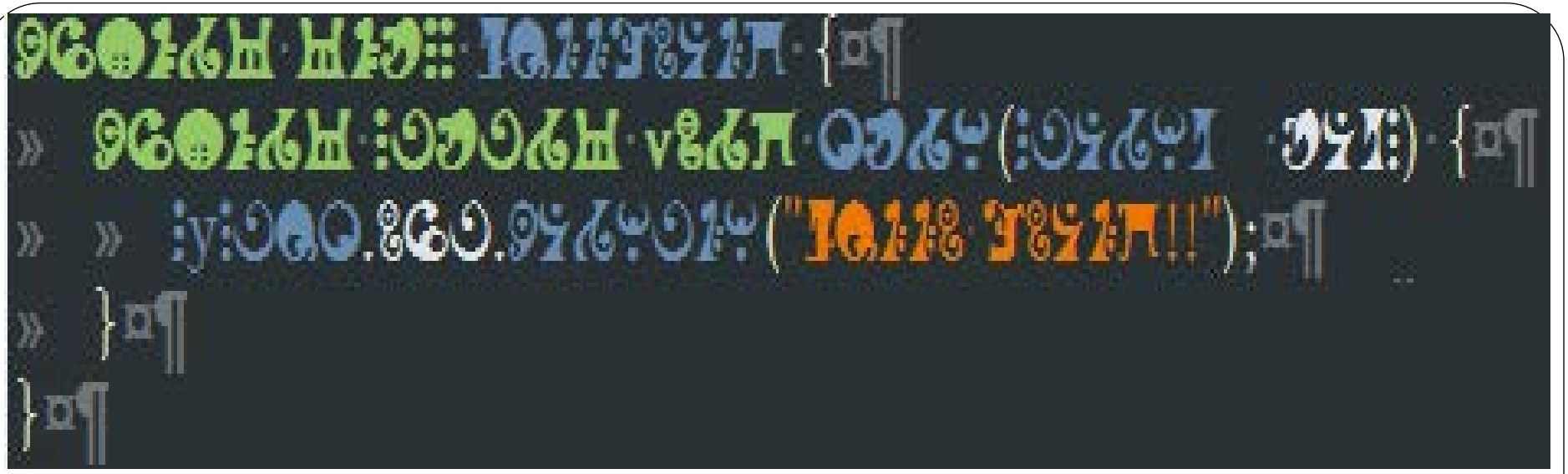
余談の部分はちょっと難しいと感じたかもしれませんが、何度もやってるうちに覚えるので、大丈夫！！

Q. 魔法少女出てきてなくね？

A. あんまりいいネタが思い浮かばなかったので・・・

Q. なんでEclipse黒いの？

A. Eclipse Color Themeでググってくださいな



魔女文字でのHello World!!です
これくらいしか思いつかなかったので、登場してません
フォントは探せばフリーのが落ちてます

画像載せても、キモオタ乙wwってなるので、やめました

やる夫で学ぶシリーズは結構勉強になるものも多いので、
ググるよろし
高校、大学数学シリーズとかもあるよ！

今度こそ終わりです
お疲れ様でした！！！