

教科書第5章「場合に応じた処理」を理解し、次の間に答えられるようにする。

- 条件の書き方（関係演算子）を理解し、様々な条件が書けるようになる
- if文の動作、書き方を理解し、条件に応じた処理を書けるようになる
- 複数の条件の判断、つまりif ~ else if ~ else の形と、switch文の形を理解する
- 論理演算子による条件の組合せを理解し、様々な条件が書けるようになる

## 1. 前回の小テストのコメント

[2] で、変数を宣言すると同時に値を初期化する書き方（`int a = 3;`のように）をすることができます。このときに、型が違うとどうなるでしょうか？ たとえば `double num1 = 2;` とするとどうでしょうか？

- 変数の値は、`double num1;` と宣言した後、別の文で `num1 = 2;` と代入したのと同じ効果があります。つまり、double型の変数 `num1` に、整数の値2を代入するので、自動的な型変換（正式名称は「暗黙の型変換」）が起こり、整数2はdouble型の値2.0に変換されてから、`num1` に入れられ（初期化）ます。なので、`println(num1);` とすれば、表示は2.0（整数値2ではなくて、小数の値2.0）となります。
- たとえば⑤で、`double num1=7;` `double num2=32;` `double num3=5;` のように宣言 + 初期値設定すると、初期値を設定するときにdouble型に変換されて、それぞれ7.0、32.0、5.0になって、それを `((num1+num2)/num3)` とすると全体にdouble型で計算されます。
- なお、逆の場合、つまり、`int num1=3.14;` と書くと、これは小数（double型・float型によらず）を整数に代入するのと同じことになって、コンパイルの時に <エラー: 精度が低下している可能性 `int num1 = 3.14;`> と言われます。
- こんな書き方をした人もいました。たとえば③で、`double a = (double) 5 / 3;` や、`double a = 5 / (double) 3;` と書く場合です。この式を計算するときに、それぞれの部分で型がどう決まってゆくのか、きちんと説明できるようにしておいてください。
- ④は、`30÷7` の「余り」を要求しています。だから、`30/7` ではありません。「余り」は剰余演算子%を使って計算できます。もちろん、自分で計算しても構いませんが、`(a%b = a - (a/b) * b)` で計算できます。)

[4] で、`int num1 = Integer.parseInt(str1); int num2 = Integer.parseInt(str2);` としておいて、この`num1`と`num2`を使って、`println( num1 * num2 / 2 )` とした場合、結果はどうなりますか？ 型について考えてみましょう。

人によっていろいろなパターンがありました。 たとえば、

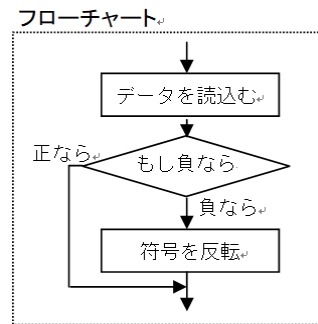
- `println( (double)num1 * (double)num2 / (double) 2 )`
- `println( (double) (num1 * num2) / (double) 2 )`
- `println( (double) (num1 * num2) / 2 )`
- `println( num1 * num2 / (double) 2 )`

どれも動くでしょう。ただ、自分が何をやっているのか、（つまりの変換の目で見ると、何が起きているか）、説明できるようにしておいてください。「わからないけれど動いたからいい」というのは、（次にやって、トラブルした時に）問題です。

念のために、`println( (double) ( num1 * num2 ) / 2 )` だと、何が起こるでしょうか？

## 2. 「場合に応じた処理」の全体像

図で描くと ⇒ 「流れ図」 (フローチャート、flow chart) ～～ 教科書ではp 114に出てくるが、あまり説明がない? ～～



ひし形の部分 ⇒ 条件を判断して場合に応じた処理をする

ポイント

- ① 「条件」をどうやって指定するのか
- ② 場合に分かれた後、それぞれの処理をどう書く (指示する) のか?
- ③ 複雑な条件・ケースをどう書くか?

### 3. 条件の仕組 — 論理値 (真か偽か、真偽値、True or False、論理値、.....) で決まる。

教科書p 110: 「良い成績ならば」 ⇒ どのようなときに「成績が良い」と判断するのか? ⇒ 点数が80点以上か? 本格的には、(たぶん?) 情報数理の授業で習う。アーキテクチャでもカバーする。

プログラムや情報数理で扱うのは、真か偽かの判断が決まる問だけ。決め方自体は守備範囲外!! 「あの人はいいい人だ」 ← 判断基準?? 「今日は良い天気だ」 ← 判断基準??

プログラムで扱う<真偽値の元> ～ たとえば「関係」 (関係演算子) (他にあるかなあ? 定数true, false) 関係演算子: 表5-1 2つのもの(式)のあいだの、等しい等しくないとか大きい/小さいとか という関係 関係式: <何かの式> <関係演算子> <何かの式> 例:  $(x+y) > q$  とか  $5 > 3$  とか 関係式を「評価」する (=真trueか偽falseかを定める)  $x$ が3、 $y$ が5、 $q$ が0であれば  $(x+y) > q$  は真true

脱線: 表5-1は覚えてください。いくつか注意点。

- 「等しい」は「==」 ← 「=」は代入の意味で、「等しいか?」の意味ではない。
- 「!=」は「等しくない」。p 135に出てくる「論理演算子」でも「!」がnotとして出てくるがこれとは違うもの。
- 「>=」や「<=」は、数学の「 $\geq$ 」「 $\leq$ 」の意味だが、コンピュータに扱える文字の中に $\geq$ や $\leq$ がないので代用。

[確認の問題]

- 変数xの値が3のとき、 $x < 2$ の値は (真・偽) である。
- 変数pの値が9、変数qの値が4のとき、 $p - q > 1$ の値は (真・偽) である。

### 4. if文 (p 121、5.3節)

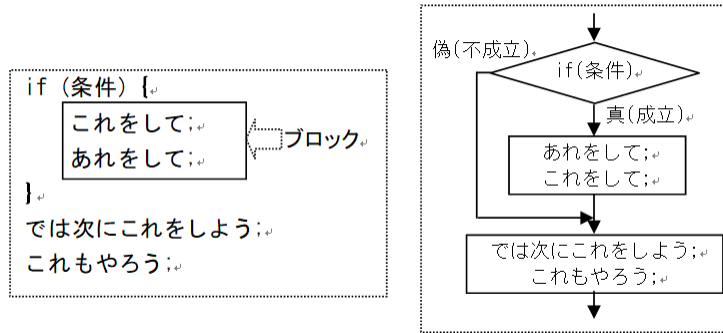
条件の値 (真trueか偽falseか) によって、真のときだけ、指定した処理をする。図5-3参照。Sample1より単純な例 (やっていることに意味はない...)

```
class sample5_0 {
    public static void main(String[] args) {
        int x = 3;
        if (x == 3)
            System.out.println("等しい");
    }
}
```

Sample1は、xの代わりに、キーボードからの入力（p 64復習）を変数resに代入し、1と比較している。

図5-4で、trueの時とfalseの時の動作の違いを理解する。

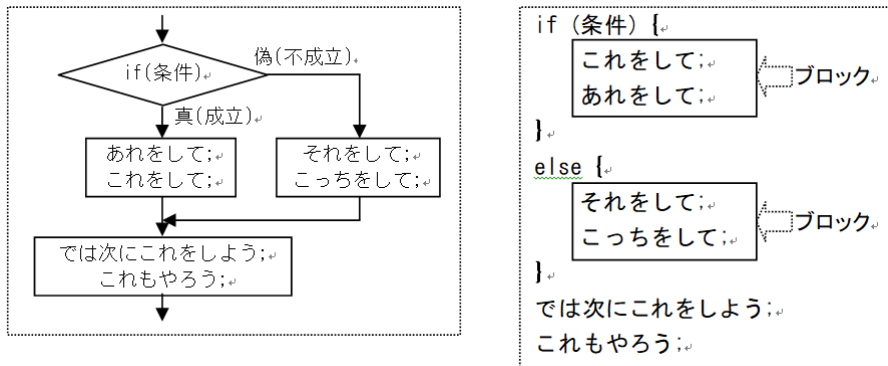
複数の文 ⇒ ブロック（かたまり）



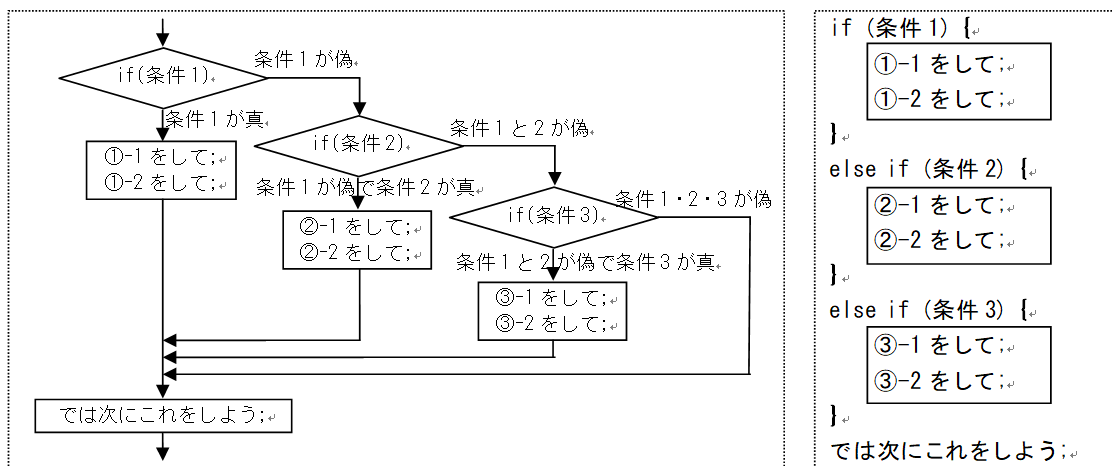
{ } で囲んで「ブロック」だと示す p 119 ブロックにしないと？

### 5. if ~ else 文 (p 121、5.3節)

条件の値（真trueか偽falseか）によって、異なる処理（真のときにする処理、vs 偽のときにする処理）をする。



### 6. 複数の条件 (p 125、5.4節) ~ else if のスタイル



### 7. switch文 (p 129、5.5節) switchとcaseとbreak (とdefault)

値によって選択する。教科書p 129の構文。 1行目switch(式)の式の値によって、以下のcaseのどれかが選択されて実行される。

右の例（教科書p 131のSample5.javaに準ずる）では、変数xの値はキーボードから入力されたint型だとする。その時は、**case 1:**のように書く。

p 133 Sample6.javaでは、キーボードから入力された「文字」(char型)を見て振り分ける。(注: readLine()で入力された値は文字列(String)だが、その中の第0文字目(=先頭の文字)を charAt(0)で取出して「文字」にしている。)このときの case の直後に書くのは **case 'a':** のように「文字型」にする。

もし**case "a":** とすると、一致しない。(switchの引数との間で型の不一致) (注 なお、String型がswitchの引数に認

められるようになったのはDK1.7以降という話である。)

```
xの値 = キーボードから入力:
switch(x) {
  case 1:
    値が1の時の処理;
    break;
  case 2:
    値が2の時の処理;
    break;
  case 3:
    値が3の時の処理;
    break;
  default:
    それ以外の時の処理;
    break;
}
```

### 8. 論理演算子 (p 135、5.6節) ～～ もっと複雑な条件を書きたい

「論理式」が書ける。論理式 ~ 論理値 (比較演算の結果や、論理定数true、false) や他の論理式 (条件) を組合せて論理値を計算する。(成績優) かつ (財産>百万円) (身長>180) または (出身地千葉) ((成績優) かつ (財産>百万円)) または ((身長>180) かつ 否定(出身地千葉))

論理演算子 要するに、かつ (AND)、または (OR)、否定 (NOT) かつ (AND、 $\wedge$ )  $\Rightarrow$  **&&** または (OR、 $\vee$ )  $\Rightarrow$  **||** 否定 (NOT、 $\neg$ )  $\Rightarrow$  **!** と書く (注: **&** 1つや **|** 1つは、別の意味があるので注意。 ビットAND / ビットOR)

論理演算の細かい性質や、演算の練習は、別の科目(たとえば情報数理?)でやってください。

で、if文の後の条件の部分には、論理式 (=論理演算子で繋いだ式) が書ける。 `if ( x == 3 || x == 4 ) System.out.println("3か4です");`

### おまけっばい話 「条件演算子」 ～～ 「論理」演算子じゃないよー

if文  $\Rightarrow$  条件によって、異なる処理をする。 処理の流れを変える(分ける)。 条件演算子  $\Rightarrow$  条件によって、異なる(式の)値を出す。 式を評価すると、条件によって異なる値が返る。

「もし整数jokenの値が3だったら値7000を返し、それ以外だったら値200を返す」 `kotae = (joken == 3) ? 7000 : 200;` この代わりにif文で書くとどうなるか?

## 第5回 小テスト (今日の復習)

1. 次の条件を、論理演算子を使って記述してください。

- ① 変数nenreiに年齢データ (整数) が入っているとき、「10代である」という条件 (10代の時だけ真になる論理式)
- ② 変数sinchoに身長データ (double) が入っているとき、「180以上である」という条件
- ③ 変数mojiにキーボード入力されたアルファベット大文字 (char) が入っているとき、「ローマ字で見て母音」という条件

2. キーボードから整数の値を入力させて、その値によって次のような出力を表示するプログラムを作り、Jcpadで試してください。 (正しく動いたかどうかを、TAの人に確認してもらってください。)

```
import java.io.*;
class Sample5_EX1
{
    public static void main(String[] args) throws IOException
    {
        System.out.println("整数を入力してください。");
        BufferedReader br =
            new BufferedReader(new InputStreamReader(System.in));
        String str = br.readLine();
        int res = Integer.parseInt(str);

        .....
        .....
        .....
        .....
    }
}
```

3. キーボードからアルファベットを1文字入力させて、その文字がローマ字で見て母音なら「母音です」、それ以外なら「母音ではない」、と表示するプログラムを作り、Jcpadで試してください。 (TAの人に確認してもらってください。)

```
import java.io.*;
class Sample5_EX2
{
    public static void main(String[] args) throws IOException
    {
        System.out.println("アルファベットを入力してください。");
        BufferedReader br =
            new BufferedReader(new InputStreamReader(System.in));
        String str = br.readLine();
        char res = str.charAt(0);

        .....
        .....
        .....
        .....
    }
}
```