

2 条件分岐

計算機に処理させたい内容は、状況に応じて変化することがあります。「あるときは A を処理させたい」「またあるときは B を処理させたい」というように、時と場合に応じて分岐させたい、ということがあります。本章では、条件に応じて処理内容を分岐させるプログラムについて解説します。

2.1 【サンプルプログラム】おみくじ

早速ですが簡単なサンプルプログラムとして、「大吉」「中吉」「小吉」「凶」のいずれかをランダムに出力する、「おみくじ」のプログラムを紹介します。

このプログラムで 3 行目に用いられている `Math.random()` は、0.0 から 1.0 の間でランダムな値（乱数という）を生成するものです。プログラムの 3 行目では、この結果を変数 `value` に代入します。

続いて 4 行目以降では、その値の大小によって条件分岐します。4 行目の `if(value >= 0.75)` では、`value` に代入された値が 0.75 以上という条件が成立すれば、その後の { と } の間を実行します。具体的には "Daikichi" と画面表示します。

これが成立しないときには、7 行目の `else` に移動して、さらに `if(value >= 0.5)` という条件分岐に入ります。ここで「`value` に代入された値が 0.5 以上」という条件が成立すれば、その後の { と } の間を実行します。具体的には "Chukichi" と画面表示します。

これも成立しないときには、10 行目の `else` に移動して、さらに `if(value >= 0.25)` という条件分岐に入ります。ここで「`value` に代入された値が 0.25 以上」という条件が成立すれば、その後の { と } の間を実行します。具体的には "Shokichi" と画面表示します。

これも成立しないときには、13 行目の `else` に移動して、その後の { と } の間を実行します。具体的には "Kyo" と画面表示します。

```
public class Omikuji {
    public static void main(String[] args) {
        double value = Math.random();
        if(value >= 0.75) {
            System.out.println("Daikichi");
        }
        else if(value >= 0.5) {
            System.out.println("Chukichi");
        }
        else if(value >= 0.25) {
            System.out.println("Shokichi");
        }
        else {
            System.out.println("Kyo");
        }
    }
}
```

では早速、このプログラムを `Omikuji.java` というファイル名で保存して、繰り返し実行してみてください³。"Daikichi", "Chukichi", "Shokichi", "Kyo" のいずれかがランダムに表示されるのではないかと思います。

³ここでいう「繰り返し実行する」とは、`javac` コマンドを 1 回だけ実行して、そのあと `java` コマンドを繰り返す、という意味だということがおわかりでしょうか。

2.2 if文による条件記述

では、サンプルプログラムで多用された「if」を用いた以下のプログラムを紹介します。

```
public class If {
    public static void main(String[] args) {
        int a = 10;
        if(a % 2 == 0) {
            System.out.println("even");
        }
    }
}
```

このプログラムでは、まず3行目で変数 a を10に初期化します。

続いて4行目のif以下は、条件を指定します。具体的に言うと、ifの次にある () の中に記述された条件が成立する時、その後にある { と } の間の命令を実行します。つまり、

もし であれば を実行する

という内容をプログラムとして記述すると、

```
if(      ) {      }
```

というようになるわけです。

さて、この4行目の () の中にある条件について解説します。左半分の $a \% 2$ は前述の通り、 a を2で割った剰余を示します。続いてその後ですが、 $== 0$ というのは「左辺が0であるかどうか」を判断する条件文を表します。つまり、 a を2で割った剰余が0であれば、() の中にある条件は満たされ、{ と } の間、つまり5行目にある `System.out.println("even");` が実行されます。

では、このプログラムを `If.java` というファイル名で保存して、実行してみましょう。以下のような表示結果が表示されれば、正常に動作しているといえましょう。

```
even
```

では、このプログラムを以下のように書き換えて下さい。

```
public class If {
    public static void main(String[] args) {
        int a = 9;
        if(a % 2 == 1) {
            System.out.println("odd");
        }
    }
}
```

これを実行したとき、以下のような表示結果が表示されれば、正常に動作しているといえましょう。⁴

```
odd
```

条件を記述する算術記号

さて、if 文の () の中に記述する条件として、前述のプログラムでは == という記号を使いましたが、条件を記述する算術記号は他にも数種類あります。表 3 に、その代表的な記号を列挙します。

表 3: Java 言語における条件記述。

算術記号	条件の説明
a == b	a と b が等しい
a != b	a と b が等しくない
a < b	a が b より小さい
a > b	a が b より大きい
a <= b	a が b 以下 (等しくてもよい)
a >= b	a が b 以上 (等しくてもよい)

2.3 if 文と else 文による条件分岐

続いて if に加えて else という文を加えた例を示します。else は、

```
もし 条件 であれば 実行する
さもなければ x x を実行する
```

という内容を

```
if( 条件 ) { 実行する } else { x x }
```

というように記述するために用います。

では早速、else 文を使ったプログラムの例を示します。

```
public class IfElse {
    public static void main(String[] args) {
        int a = 10;
        if(a % 2 == 0) {
            System.out.println("even");
        }
        else {
            System.out.println("odd");
        }
    }
}
```

⁴even とは英語で偶数、odd とは英語で奇数を意味します。

```
}  
}
```

このプログラムの前半は、前述のプログラムと同様に、まず変数 a に 10 を代入し、続いてそれを 2 で割った剰余を求め、それが 0 であるかを判定します。この条件が成立するときには、その直後の { と } の間を実行します。つまり画面には even と表示されるはずですが、

一方、この条件が成立しないときには、else の直後の { と } の間を実行します。つまり画面には odd と表示されるはずですが、

それでは、このプログラムを IfElse.java というファイル名で保存して、実行してみましょう。以下のような表示結果が表示されれば、正常に動作しているといえましょう。

```
even
```

また、プログラムの 3 行目の代入文を、10 の代わりに奇数にしてみてください。以下のような表示結果が表示されれば、正常に動作しているといえましょう。

```
odd
```

2.4 条件の論理積と論理和

if 文の条件に、2 つ以上の条件を組み合わせたい、という場合があります。Java 言語では例えば、2 つ以上の条件を同時に満たす、あるいは 2 つ以上の条件を最低一方満たす、というように条件を設定することができます。

2 つの条件を同時に満たす、つまり「かつ」という条件を記述するには、&& という記号を用いて、

```
if(条件1 && 条件2)
```

というように記述します。

2 つの条件の最低一方を満たす、つまり「または」という条件を記述するには、|| という記号を用いて、

```
if(条件1 || 条件2)
```

というように記述します。

ここで「かつ」というように、複数の条件の両方、という記述を論理積といいます。また「または」というように、複数の条件の一方、という記述を論理和といいます。

上述の && および || のように、論理積や論理和などの演算を記述する記号を、論理演算子といいます。論理演算子の代表的なものを、表 4 に列挙します。

表 4: Java 言語における論理演算子。

記号の種類	記号の説明
&&	かつ
	または

論理演算子を用いたプログラムの例を、以下に示します。このプログラムでは4行目のif文にて、 $a \% 2 == 0$ および $a \geq 5$ の両方の条件を満たした場合に限り、OK と表示します。どちらか片方でも満たさなければ、NG と表示します。

```
public class IfElse {
    public static void main(String[] args) {
        int a = 10;
        if(a % 2 == 0 && a >= 5) {
            System.out.println("OK");
        }
        else {
            System.out.println("NG");
        }
    }
}
```

このプログラムでは論理積の記号 `&&` を例にしましたが、論理和の記号 `||` も、用法は全く同じです。