

プログラム例 1

```
public class Prog04_05
{
    public static void main(String[ ] args)
    {
        int a = 50;

        if(a >= 60) {
            if(a < 70) {
                System.out.println("その点数は範囲内です");
            }
            else{
                System.out.println("その点数は範囲外です");
            }
        }
        else{
            System.out.println("その点数は範囲外です");
        }
    }
}
```

また、if-else if-else 文を使って条件 2 を判別するには以下のようなプログラムが考えられる。

プログラム例 2

```
public class Prog04_06
{
    public static void main(String[ ] args)
    {
        int a = 50;

        if(a < 60) {
            System.out.println("その点数は範囲外です");
        }
        else if(a >= 70) {
            System.out.println("その点数は範囲外です");
        }
        else{
            System.out.println("その点数は範囲内です");
        }
    }
}
```

これらのプログラムは論理演算子を用いると簡潔になる。論理演算子は 2 つあり、AND 論理演算子 (&&) と OR 論理演算子がある。AND 論理演算子は「～でかつ～」という条件の判別に利用することができ、OR 論理演算子は「～または～」という条件の判別に用いることができる。これらを用いると先ほど示したプログラムは以下のように記述できる。

プログラム例 3 AND 論理演算子

```
public class Prog04_07
{
```

```

public static void main(String[ ] args)
{
    int a = 50;

    if(a >= 60 && a < 70) {
        System.out.println("その点数は範囲内です");
    }
    else{
        System.out.println("その点数は範囲外です");
    }
}
}

```

AND 論理演算子を用いる場合では演算の対象となる 2 つの式（ここでは、 $a \geq 60$ と $a < 70$ ）が両方とも真となった場合のみ、その式（ $a \geq 60 \ \&\& \ a < 70$ ）の評価が真となる。それ以外は偽である。

プログラム例 4 OR 論理演算子

```

public class Prog04_08
{
    public static void main(String[ ] args)
    {
        int a = 50;

        if(a < 60 || a >= 70) {
            System.out.println("その点数は範囲外です");
        }
        else{
            System.out.println("その点数は範囲内です");
        }
    }
}

```

OR 論理演算子を用いる場合では演算の対象となる 2 つの式（ここでは、 $a < 60$ と $a \geq 70$ ）のうち、少なくとも片方が真だった場合には、その式（ $a < 60 \ || \ a \geq 70$ ）の評価が真となる。偽になる場合は 2 つの式の両方が偽になった場合だけである。

このほか「～でない」という判別を行う演算子として否定演算子 (!) がある。この否定演算子を用いると以下のようなプログラムが書ける。

プログラム例 5 否定演算子

```

public class Prog04_09
{
    public static void main(String[ ] args)
    {
        int a = 50;

        if(!(a < 60 || a >= 70)) {
            System.out.println("その点数は範囲内です");
        }
        else{
            System.out.println("その点数は範囲外です");
        }
    }
}

```

```

    }
}
}

```

ここまでで示した5つプログラムは全て60~69の範囲の値が変数aに格納されている場合には「その点数は範囲内です」と表示し、それ以外の値が格納された場合には「その点数は範囲外です」と表示する。

同じ動作をするプログラムではあるが、いくつかの書き方が存在することが理解できたろうか。説明のためにこのような例を示したが、実際にこのような動作をさせるプログラムを作成する場合にはAND論理演算子を用いたプログラム例3を用いることが一般的である。プログラミングを行う上での方針として、

- ①記述量が少ないこと
- ②読みやすいこと

を満たすように心がけると良い。こうするとプログラミングの作成時間が短縮でき、バグも入りにくくなる。

論理演算子と否定演算子について表1にまとめておく。

表1 論理演算子と否定演算子

| | 条件式 | 意味 | プログラムでの表現 |
|-----------|--------|---------|---------------|
| AND 論理演算子 | a && b | a かつ b | x>=0 && x<5 |
| OR 論理演算子 | a b | a または b | x==1 y==3 |
| 否定演算子 | !a | a ではない | !(x>1 y>3) |

注) a や b は条件式である

また、条件式を a、b とした時の各演算子の真偽値表を以下に示しておく。ただし、偽=0 真=1 とする。

表2 真偽値表

| a | b | a && b | a b | !a |
|---|---|--------|--------|----|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | / |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | / |

条件式の値

ここまでは条件式を評価すると真か偽の結果が得られると示した。これは実は正確ではない。関係演算子や論理演算子も演算であるので、算術演算子と同様に値をもっている。しかし値と言っても通常の整数型や実数型のような「数値」ではなく、その演算結果は「論理値」となる。

「論理型」として「boolean」という変数があることは以前に学んだが、boolean型はこの「論理値」として、「true」と「false」を扱う変数である。

例えば a>1 という関係演算は a が 1 より大きい場合に評価結果が真となり、a が 1 以下の場合には評価結果が偽となるが、真の場合には a>1 の演算結果として boolean 型の「true」が得られる。偽であれば演算結果として boolean 型の「false」が得られる。例えば b を boolean 型の変数とした場合には

```

b = true;
b = false;

```

という代入を行うことができる。また、条件式を右辺に書いて、

```

b = (a > 1);

```

という代入もできる。結果は a が 1 より大きい場合には b には true が代入され、a が 1 以下の場合には b には false が代入される。よって、以下のような文も可能である。この場合では上の if 文は常に真になり、下の if 文は常に偽になる。

```
if(true) System.out.println("真");
if(false) System.out.println("偽");
```

演算子の優先順位

以前に算術演算子と代入演算子について演算の優先順位を示したが、新しい演算子を習ったので、改めて表 3 に示す。

表 3 演算子の優先順位

| 種類 | 演算子 |
|---------|-----------|
| 括弧 | () |
| 否定 | ! |
| 乗除余 | * / % |
| 加減 | + - |
| 比較 | < > <= >= |
| 等価 | == != |
| 論理的 AND | && |
| 論理的 OR | |

上位に書かれているものほど優先順位が高い。また、優先順位が等しい場合には式の左側から順に評価していく。

演習

プログラム例 1～5 を作成し、出力を確認しなさい。

課題 2 の続き

kadai2_4 以下について演算子の優先順位を考慮し、手計算を行った後、後に示したプログラムを参考にしてプログラミングし、実行することで結果を確認しなさい。

- ```
a = -4 b = -2 c = 0 d = 2 e = 4
```
- 1) a < b            [解答例] true
  - 2) a > b
  - 3) a+b > c+d
  - 4) a == -4 && c < d
  - 5) a != -4 || e != 4
  - 6) a < 0 || b < c+d
  - 7) a < 0 && c == 0 && d != 2
  - 8) (a < -5 || c <= 0) && d >= 0
  - 9) !(a+b > c)
  - 10) !(a+b < -5) || !((c+d) > 5)

### プログラミング例

```
public class Kadai02_04
{
 public static void main(String[] args)
 {
 int a, b, c, d, e;

 a = -4; b = -2; c = 0; d = 2; e = 4;

 System.out.println("a=" + a + " %tb=" + b + " %tc=" + c + " %td=" + d + " %te=" + e);
 System.out.println("1) a < b の結果は" + (a < b) + "です");
 }
}
```

|                                                        |
|--------------------------------------------------------|
| }<br>}                                                 |
| a = -4 b = -2 c = 0 d = 2 e = 4<br>1) a < b の結果はtrueです |

kadai2\_5 キーボードから月を入力し、対応する季節を表示するプログラムを作成しなさい。ただし、季節は以下に従うものとする。

3月～5月は春、6月～8月は夏、9月～11月は秋、12、1、2月は冬

実行例は以下である。

|                                              |
|----------------------------------------------|
| 実行例 (斜体は入力値)                                 |
| 月を入力してください : <i>11</i><br><i>11</i> 月の季節は秋です |

1～12以外の数値が入力されたら、「入力できる値は1から12までです」などの表示を行うこと。