

Java プログラミング I

7回目 switch 文と論理演算子

● 条件判断文3 switch 文

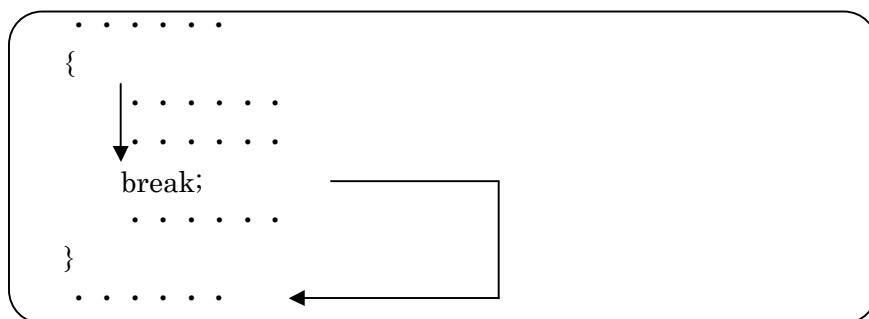
switch 文 式が case の値と一致した場合、そこから直後の break;までを処理し、
どれにも一致しない場合、default;から直後の break;までを処理する。
但し、式や値 1、値 2 は整数または文字である。

```
switch( 式 )
{
    case [値 1]: // ←コロン ":" です。セミコロン ";" と間違えないように！！
        [文 1];
        :
        break;
    case [値 2]:
        [文 2];
        :
        break;
        :
    default: // ←default 部分は省略できる
        [文 D];
        :
        break;
}
```

○ break 文

break 文

switch ブロックまたはループ内の処理を強制的に終了し、
現在のブロックから抜ける。



ソースコード例

ソースファイル名 : Sample7_1.java

```
// 入力値の判定
import java.io.*;

class Sample7_1
{
    public static void main(String[] args) throws IOException
    {
        // キーボード入力の準備
        BufferedReader br;
        br = new BufferedReader(new InputStreamReader(System.in));

        // キーボード入力
        System.out.println("整数を入力してください。");
        int i;
        i=Integer.parseInt(br.readLine());

        switch(i) // 変数 i により処理を分岐
        {
            case 1:// i が 1 のとき、
                System.out.println("1 が入力されました。");
                break;
            case 2:// i が 2 のとき、
                System.out.println("2 が入力されました。");
                break;
            default:// i が 1 でも 2 でもないとき、
                System.out.println("1 か 2 を入力してください。");
                break;
        }
    }
}
```

実行画面

```
>java Sample7_1
整数を入力してください。
1
1 が入力されました。
-- Press any key to exit (Input "c" to continue) --

>java Sample7_1
整数を入力してください。
2
2 が入力されました。
-- Press any key to exit (Input "c" to continue) --

>java Sample7_1
整数を入力してください。
3
1 か 2 を入力してください。
-- Press any key to exit (Input "c" to continue) --
```

ソースコード例

ソースファイル名 : Sample7_2.java

```
// 入力文字の判定
import java.io.*;

class Sample7_2
{
    public static void main(String[] args) throws IOException
    {
        // キーボード入力の準備
        BufferedReader br;
        br = new BufferedReader(new InputStreamReader(System.in));

        // キーボード入力
        System.out.println("a か b を入力してください。");
        String str;
        str = br.readLine();

        // 文字列から一文字を取り出す
        char c;
        c = str.charAt(0); // charAt(x) : x 文字目の文字を取り出す 但し、0 が先頭

        switch(c) // 変数 c により処理を分岐
        {
            case 'a': // c が'a'のとき、
                System.out.println("a が入力されました。");
                break;
            case 'b': // c が'b'のとき、
                System.out.println("b が入力されました。");
                break;
            default: // c が'a'でも'b'でもないとき、
                System.out.println("a か b を入力してください。");
                break;
        }
    }
}
```

実行画面

```
>java Sample7_2
a か b を入力してください。
a
a が入力されました。
-- Press any key to exit (Input "c" to continue) --

>java Sample7_2
a か b を入力してください。
b
b が入力されました。
-- Press any key to exit (Input "c" to continue) --

>java Sample7_2
a か b を入力してください。
c
a か b を入力してください。
-- Press any key to exit (Input "c" to continue) --
```

- switch 文で break;を省略したらどうなる？
- Sample7_1.java の break;を取り除いた場合の実行画面

```
>java Sample7_1
整数を入力してください。
1
1 が入力されました。
2 が入力されました。
1 か 2 を入力してください。
-- Press any key to exit (Input "c" to continue) --

>java Sample7_1
整数を入力してください。
2
2 が入力されました。
1 か 2 を入力してください。
-- Press any key to exit (Input "c" to continue) --

>java Sample7_1
整数を入力してください。
3
1 か 2 を入力してください。
-- Press any key to exit (Input "c" to continue) --
```

● 論理演算子

論理演算子

!, &&, ||

真(true)と偽(false)の間で論理演算を行う。
“～ではない”、“かつ”、“または”を表現する。

!	論理否定	<table border="1"> <thead> <tr> <th>a</th><th>!</th></tr> </thead> <tbody> <tr> <td>true</td><td>false</td></tr> <tr> <td>false</td><td>true</td></tr> </tbody> </table>	a	!	true	false	false	true	例えば、!(3<5)は false									
a	!																	
true	false																	
false	true																	
&&	論理積	<table border="1"> <thead> <tr> <th>a</th><th>b</th><th>a&&b</th></tr> </thead> <tbody> <tr> <td>true</td><td>true</td><td>true</td></tr> <tr> <td>true</td><td>false</td><td>false</td></tr> <tr> <td>false</td><td>true</td><td>false</td></tr> <tr> <td>false</td><td>false</td><td>false</td></tr> </tbody> </table>	a	b	a&&b	true	true	true	true	false	false	false	true	false	false	false	false	例えば、(1==0)&&(1<2)は false
a	b	a&&b																
true	true	true																
true	false	false																
false	true	false																
false	false	false																
	論理和	<table border="1"> <thead> <tr> <th>a</th><th>b</th><th>a b</th></tr> </thead> <tbody> <tr> <td>true</td><td>true</td><td>true</td></tr> <tr> <td>true</td><td>false</td><td>true</td></tr> <tr> <td>false</td><td>true</td><td>true</td></tr> <tr> <td>false</td><td>false</td><td>false</td></tr> </tbody> </table>	a	b	a b	true	true	true	true	false	true	false	true	true	false	false	false	例えば、(1==0) (1<2)は true
a	b	a b																
true	true	true																
true	false	true																
false	true	true																
false	false	false																

但し、a と b は boolean 型の変数

ソースコード例

ソースファイル名 : Sample7_3.java

```
// 論理演算子の真理値表
class Sample7_3
{
    public static void main(String[] args)
    {
        System.out.println("!true = " + (!true));
        System.out.println("!false = " + (!false));
        System.out.println("true && true = " + (true && true));
        System.out.println("true && false = " + (true && false));
        System.out.println("false && true = " + (false && true));
        System.out.println("false && false = " + (false && false));
        System.out.println("true || true = " + (true || true));
        System.out.println("true || false = " + (true || false));
        System.out.println("false || true = " + (false || true));
        System.out.println("false || false = " + (false || false));
    }
}
```

実行画面

```
>java Sample7_3
!true = false
!false = true
true && true = true
true && false = false
false && true = false
false && false = false
true || true = true
true || false = true
false || true = true
false || false = false
-- Press any key to exit (Input "c" to continue) --
```

ソースコード例

ソースファイル名 : Sample7_4.java

```
// 大文字・小文字の処理
import java.io.*;

class Sample7_4
{
    public static void main(String[] args) throws IOException
    {
        // キーボード入力の準備
        BufferedReader br;
        br = new BufferedReader(new InputStreamReader(System.in));

        // キーボード入力
        System.out.println("あなたは男性ですか？");
        System.out.println("Y か N を入力してください。");
        String str;
        str = br.readLine();

        // 文字列から一文字を取り出す
        char c;
        c = str.charAt(0); // charAt(x) : x 文字目の文字を取り出す

        if(c == 'Y' || c == 'y') // Y または y のとき、
        {
            System.out.println("あなたは男性ですね。");
        }
        else
        {
            if(c == 'N' || c == 'n') // N または n とき、
            {
                System.out.println("あなたは女性ですね。");
            }
            else
            {
                System.out.println("Y か N を入力してください。");
            }
        }
    }
}
```

実行画面

```
>java Sample7_4
あなたは男性ですか？
YかNを入力してください。
Y
あなたは男性ですね。
-- Press any key to exit (Input "c" to continue) --
```

```
>java Sample7_4
あなたは男性ですか？
YかNを入力してください。
y
あなたは男性ですね。
-- Press any key to exit (Input "c" to continue) --
```

```
>java Sample7_4
あなたは男性ですか？
YかNを入力してください。
m
YかNを入力してください。
-- Press any key to exit (Input "c" to continue) --
```

- switch 文での break; の省略をうまく利用して Sample7_4.java は次のように表現できる。
ソースファイル名 : Ext7_1.java

```
// 大文字・小文字の処理 2
import java.io.*;

class Ext7_1
{
    public static void main(String[] args) throws IOException
    {
        // キーボード入力の準備
        BufferedReader br;
        br = new BufferedReader(new InputStreamReader(System.in));

        // キーボード入力
        System.out.println("あなたは男性ですか？");
        System.out.println("Y か N を入力してください。");
        String str;
        str = br.readLine();

        // 文字列から一文字を取り出す
        char c;
        c = str.charAt(0); // charAt(x) : x 文字目の文字を取り出す

        switch(c)
        {
            case 'Y':
            case 'y': // Y または y のとき、
                System.out.println("あなたは男性ですね。");
                break;
            case 'N':
            case 'n': // N または n のとき、
                System.out.println("あなたは女性ですね。");
                break;
            default:
                System.out.println("Y か N を入力してください。");
        }
    }
}
```

● 条件演算子

条件演算子

? :

条件が true のとき式 1 が、
false のとき式 2 が本演算子の値となる。
条件とは、関係演算子で表現される式
例えば、 $a < b$ 、 $a != 5$ など

条件 ? [true のときの式 1] : [false のときの式 2]

ソースコード例

ソースファイル名 : Sample7_5.java

```
// 偶数・奇数の判定
import java.io.*;

class Sample7_5
{
    public static void main(String[] args) throws IOException
    {
        // キーボード入力の準備
        BufferedReader br;
        br = new BufferedReader(new InputStreamReader(System.in));

        // キーボード入力
        System.out.println("整数を入力してください。");
        int i = Integer.parseInt(br.readLine());

        // 偶数・奇数の判断
        char c;
        c = ((i%2==0) ? 'E' : 'O');

        System.out.println("入力された整数は "+c+" です。 (E)ven:偶数、(O)dd:奇数");
    }
}
```

実行画面

```
>java Sample7_5
整数を入力してください。
4
入力された整数は E です。 (E)ven:偶数、(O)dd:奇数
-- Press any key to exit (Input "c" to continue) --
```

```
>java Sample7_5
整数を入力してください。
5
入力された整数は O です。 (E)ven:偶数、(O)dd:奇数
-- Press any key to exit (Input "c" to continue) --
```