

2-6 : 制御文—switch case

switch case 文は条件分岐させたいときに使う制御文です。

基本的な機能は if 文と一緒にですが、if 文のように「 \geq 」や「 \leq 」や「 $=$ 」をしません。

switch case の利点は $X=1, 2, 3, 4, 5, 6$ などそれぞれの条件文を書くときなどは if 文で書くより switch 文で書いたほうが見やすくなります。

<一般式>

```
switch(式) {  
    case 値 1:実行文 1;  
    break;  
    case 値 2:実行文 2;  
    break;  
    default:実行文 n;  
    break;  
}
```

注意：ここは;ではなく:。

式が値 1 なら実行文 1 を実行

式が値 2 なら実行文 2 を実行

どの条件にも当たらなかったら実行文 n を実行

<switch 例の説明>

$X=1$ のときは case 1 の実行文、 $Y += 1;$ が実行され、 $X=2$ のときは case 2 の実行文、 $Y += 2;$ が実行されます。 X の値がどちらにも当てはまらなければ、default の実行文、 $+= 100;$ が実行されます。

なぜ break があるの?と思われるかもしれませんが、この break は switch 文から抜け出すのに必要なものになります。

これにより break 以下の文は実行されなくなります。

```
//switch 例  
switch(X) {  
    case 1:Y += 1;  
    break;  
    case 2:Y += 2;  
    break;  
    default:Y += 100;  
    break;  
}
```

先ほどの switch 例を if 文に書き換えるとそのようになります。

```
if(X == 1) {
    Y += 1;
}
else if(X == 2) {
    Y += 2;
}
else{
    Y += 100;
}
```

この程度の数の条件分岐ならそれほど見にくくないのですが、これの数が膨大になると if 文よりも switch 文の方が断然見やすくなります。

ちなみに C++Builder では switch 文はよくキー入力の分岐に使われます。

例題

実行文 1~3 の実行文の内、どれが実行されるでしょうか？

```
int X = 3;

switch(X * 5 + 3) {
    case 3:実行文 1;
        break;
    case 18:実行文 2;
        break;
    default:実行文 3;
        break;
}
```