

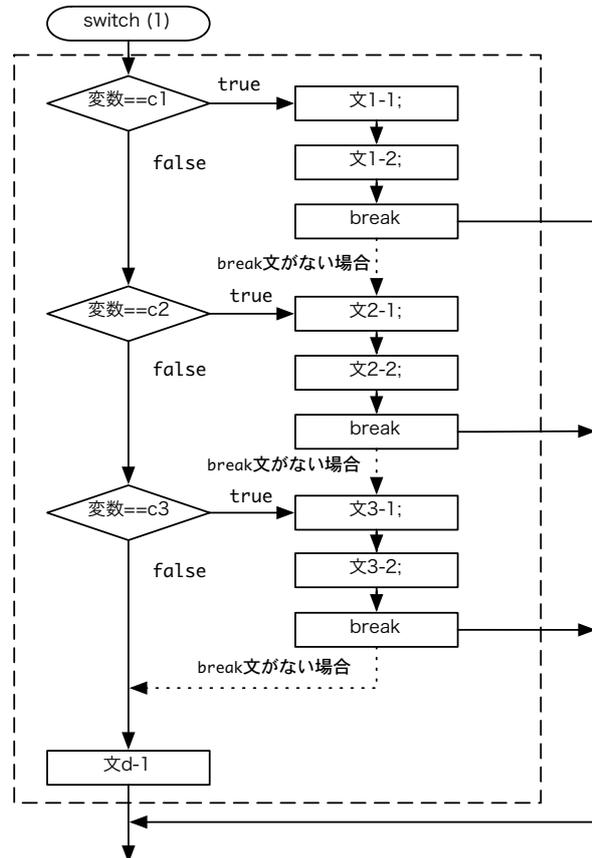
制御構造とフローチャート (2)

switch 文 (条件判断)

switch文は、変数の値が特定の値と等しい時、特別な処理をする場合に使われる条件判断の形式である。
 break 文はブロックから脱出する働きをする。switch 文の中で実行されると switch文の{}ブロック全体から抜け出して、その次の命令が実行される。break 文が無い場合は、次の case 内の文も実行される。
 default 文は省略できる。省略した場合、どの case にも変数の値が当てはまらないときは、何も実行されない。

形式 (1)

```
switch (変数) {
    case c1:
        文1-1;
        文1-2;
        ...
        break;
    case c2:
        文2-1;
        文2-2;
        ...
        break;
    case c3:
        文3-1;
        文3-2;
        ...
        break;
    ...
    default:
        文d-1;
        ...
}
```



形式 (2)

変数が c1 または c2 に等しいときに、同じ処理が行われる。また、どれとも等しくない場合は何もしない。

```
switch (変数) {
    case c1:
    case c2:
        文2-1;
        文2-2;
        ...
        break;
    case c3:
        文3-1;
        文3-2;
        ...
        break;
    ...
}
```

(練習1) 以下のプログラム switch-1.c を入力して、コンパイルし、入力の数字を0から6まで変化させて実行してみなさい。

```
1 /* switch-1.c */
2
3 #include <stdio.h>
4
5 int main()
6 {
7     int month;
8
9     printf( "季節の判断 : 何月ですか? (1-12) : " );
10    scanf( " %d", &month );
11
12    printf( "%d 月は, ", month );
13
14    switch (month) {
15        case 12:
16        case 1:
17        case 2:
18            printf( "冬です。¥n" );
19            break;
20        case 3:
21        case 4:
22        case 5:
23            printf( "春です。¥n" );
24            break;
25        default:
26            printf( "わかりません。¥n" );
27    }
28
29    return 0;
30 }
```

[課題1] 上記 switch-1.c を変更して 1-12 の数を入力すると、季節を判断するプログラム switch-2.c を作成しなさい。

[課題2] 以下の場合、if と switch のどちらを使えばよいか、理由を考えなさい。

- 整数変数 v の値が 10 より大きいかわりに小さいかで処理を分ける。
- 整数変数 v の値が 1 の場合と 2 の場合で処理を分ける。
- 整数変数 v の値が 1-10 の整数の場合、それぞれの値ごとに処理を分ける。
- 整数変数 v の値が整数変数 w, x, y, z それぞれの値と等しい場合ごとに処理を分ける。

[課題3] switch-2.c のフローチャートを書きなさい。

while 文 (繰り返し)

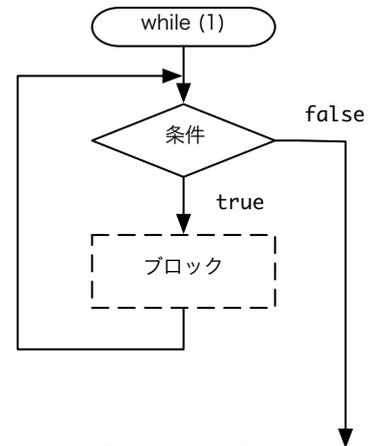
while 文は条件が真の場合はブロックを繰り返し続け、条件が偽になると繰り返しを抜けて次の文を実行する。

while 文は最初に条件判断が行われるので、ブロックが一度も実行されないこともあり得る。

while 文のブロック中で continue 文が実行されると、残りの文を実行せずに条件判断に戻る。また、break 文が実行されると、残りの文を実行せずにブロックを抜ける。

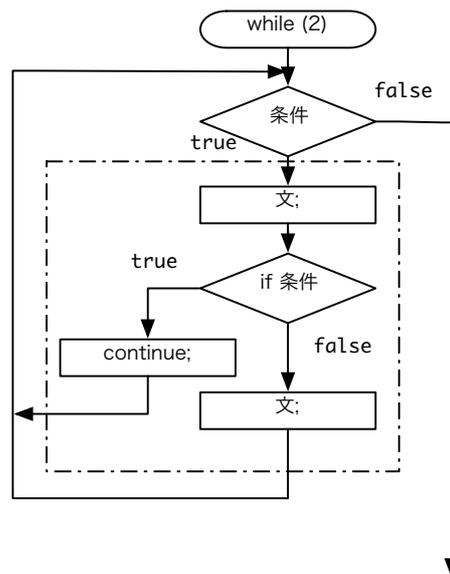
形式 (1)

```
while (条件) {
    文1-1;
    文1-2;
    ...
}
```



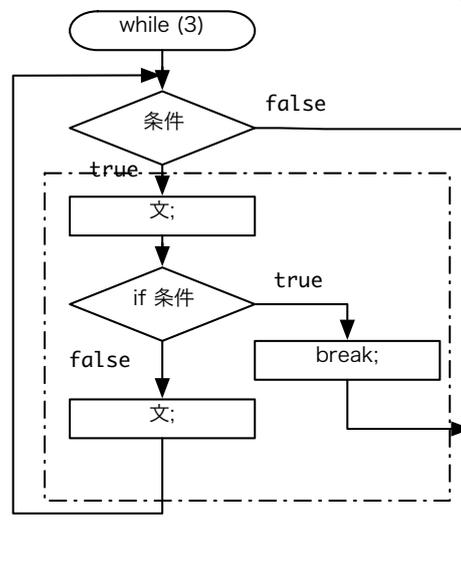
形式 (2)

```
while (条件) {
    ...
    if (条件) {
        ...
        continue;
    }
    ...
}
```



形式 (3)

```
while (条件) {
    ...
    if (条件) {
        ...
        break;
    }
    ...
}
```



(練習2) 以下のプログラム while-1.c を入力して、コンパイルし、実行してみなさい。また初期値として負の数を入力するとどうなるか確かめなさい。

```
1 /* while-1.c */
2
3 #include <stdio.h>
4
5 int main()
6 {
7     int count;
8
9     printf( "カウントダウンの初期値を入力してください : " );
10    scanf( " %d", &count );
11
12    while ( count >= 0 ) {
13        printf( "%10d !¥n", count );
14        count = count - 1;
15    }
16    printf( "終了!¥n" );
17
18    return 0;
19 }
```

[課題4] 上記 while-1.c のフローチャートを書きなさい。

[課題5] 上記 while-1.c を参考にして、a から z までを出力するプログラム while-2.c を作成しなさい。ヒント：9, 10 行目はいらぬ。文字変数は char c='a'; と宣言する。printf 文字出力指定は %c, 文字変数に1を足すと次の文字のコードになる。

do while 文 (繰り返し)

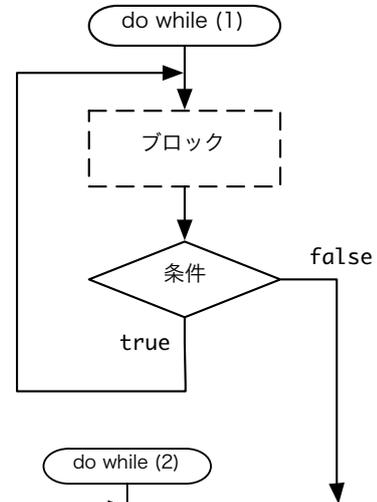
do while 文は条件が真の場合はブロックを繰り返し続け、条件が偽になると繰り返しを抜けて次の文を実行する。

do while 文は最後に条件判断が行われるので、最低一回はブロックが実行される。

do while 文のブロック中で continue 文が実行されると、残りの文を実行せずにブロックの先頭に戻る。また、break 文が実行されると、残りの文を実行せずにブロックを抜ける。

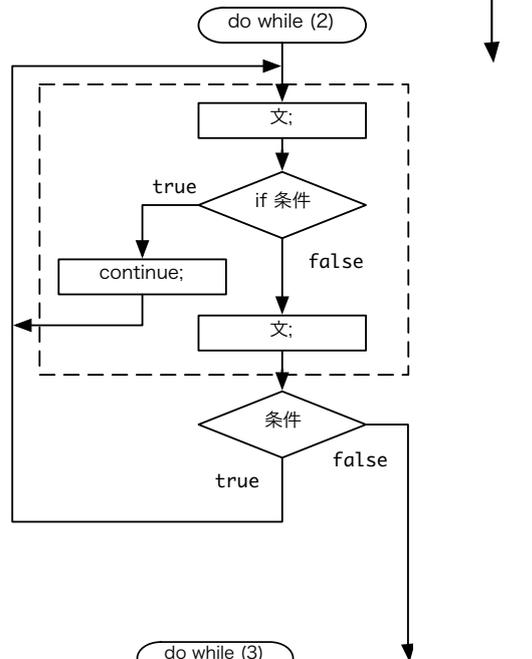
形式 (1)

```
do {
    文1-1;
    文1-2;
    ...
} while (条件);
```



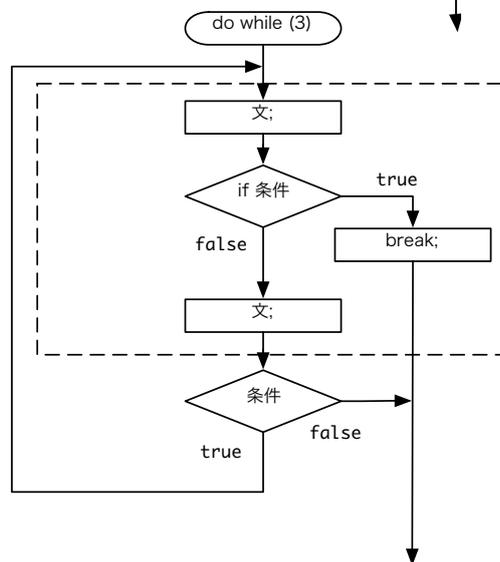
形式 (2)

```
do {
    ...
    if (条件) {
        ...
        continue;
    }
} while (条件);
```



形式 (3)

```
do {
    ...
    if (条件) {
        ...
        break;
    }
} while (条件);
```



(練習3) 以下のプログラム dowhile-1.c を入力して、コンパイルし、実行してみなさい。

```
1 /* dowhile-1.c */
2
3 #include <stdio.h>
4
5 int main()
6 {
7     int n, sum = 0;
8
9     do {
10        printf( "総和 : 数値を入力してください : " );
11        scanf( " %d", &n );
12
13        sum = sum + n;
14    } while ( n != 0 );
15
16    printf( "総和は %d です, ¥n", sum );
17
18    return 0;
19 }
```

[課題6] 上記 dowhile-1.c を参考に、初期値が 10 で減算していき 0 か負になったら終了するプログラム dowhile-2.c を作成しなさい。

[課題7] 0 から 9 までの数を入力すると、ひらがなで読み方を表示するプログラム yomiage.c を作成しなさい。

0: ぜろ, 1: いち, 2: に, 3: さん, 4: よん, ..., 9: きゅう

```
/* yomiage.c (一部) */
#include <stdio.h>

int main()
{
    int n, juu, ichi;

    printf( "整数を入力してください:" );
    scanf( " %d", &n );

    juu = n / 10;
    ichi = n % 10;

    if ( ichi == 0 ) {
        printf( "ぜろ" );
    } else {
        switch ( ichi ) {
            case 1:
                printf( "いち" ); break;
            case 2:
                printf( "に" ); break;
        }
    }

    printf( "¥n" );

    return 0;
}
```

【課題8*】1から99までの数を入力すると、ひらがなで読み方を表示するプログラム yomiage2.c を作成しなさい。

0: ぜろ, 1: いち, 2: に, 3: さん, 4: よん, ..., 9: きゅう, 10: じゅう, ..., 99: きゅうじゅうきゅう

【課題9*】 yomiage2.c をもとに、3の倍数の時には読みの後に！マークをつけるプログラム yomiage3.c を作成しなさい。

【課題10】 初期値, 増分, 終了値 (いずれも正の整数) を入力すると等差数列を表示するプログラム tousa-for.c, tousa-while-c を作成しなさい。それぞれ for 文, while文を使うこと。ただし初期値<終了値とし、値が終了値を超えたら停止するものとする。。

【課題11】 月の数字 (1月は1) を入力すると、その月の日にち (たとえば1と入力すると31) を表示するプログラム month-day-if.c と month-day-case.c を作成しなさい。それぞれ if文, switch文を使うこと。ただし2月は28日とする。

【課題12】 実数を入力するごとに、それまでの合計を表示するプログラム goukei.c を作成しなさい。ただし、0が入力されたらプログラムが終了するようにすること。

[課題13*] 1から10までの原子番号を入力すると元素記号を表示したい。プログラムに必要な適切な制御構造は何か。また、そのプログラム atom.c を作成しなさい。

(ヒント) 文字列変数は `char*` で宣言できる。例えば文字列変数 `str` の定義と代入、印字は以下の通り

```
char* str;  
str = "文字列";  
printf( "%s\n", str );
```