ループ処理 (while と for，do–while)

山本昌志*

2006 年 8 月 30 日

概 要

条件により，同じ文を何回も実行させる繰り返し文 (ループ文) について，学習する．while と for，do–while の文法を理解して，練習問題でその使い方をマスターする．

1 本日の授業内容

1.1 前回の復習

前回は教科書 [1] の 4 章 制御の流れの p.117–131 が範囲であった．多分岐についての学習であった．多分岐に使われる構文，which と if–elese if–else の使い方を学んだ．具体的な学習内容は以下の通りで，絶対に理解し，憶えなくてはならない．

- プログラム中で,「もし a=1 ならば○○する，a=2 ならば△△する，a=5 ならば□□する，さもなければ◎◎する」というような処理をしたい場合，switch という命令を使う．このように，値により処理が複数に分岐することを多分岐と言う．条件の部分が整数 (int)，あるいは文字 (char) で表される場合，switch を使う．この構文のフローチャートと書式を図 1 に示す．
- 「もし○○ならば ⊛⊛ する．さもなければ，もし□□ならば ⊠⊠ する．さもなければ，もし △△ ならば ▽▽ する．さもなければ，◎◎ する.」というよう構文を書きたい場合がある．条件に合致しなければ，次々と条件を変化させる．このような構文には，if〜else if〜else 文を使う．この構文のフローチャートと書式を図 2 に示す．

*独立行政法人秋田工業高等専門学校電気工学科

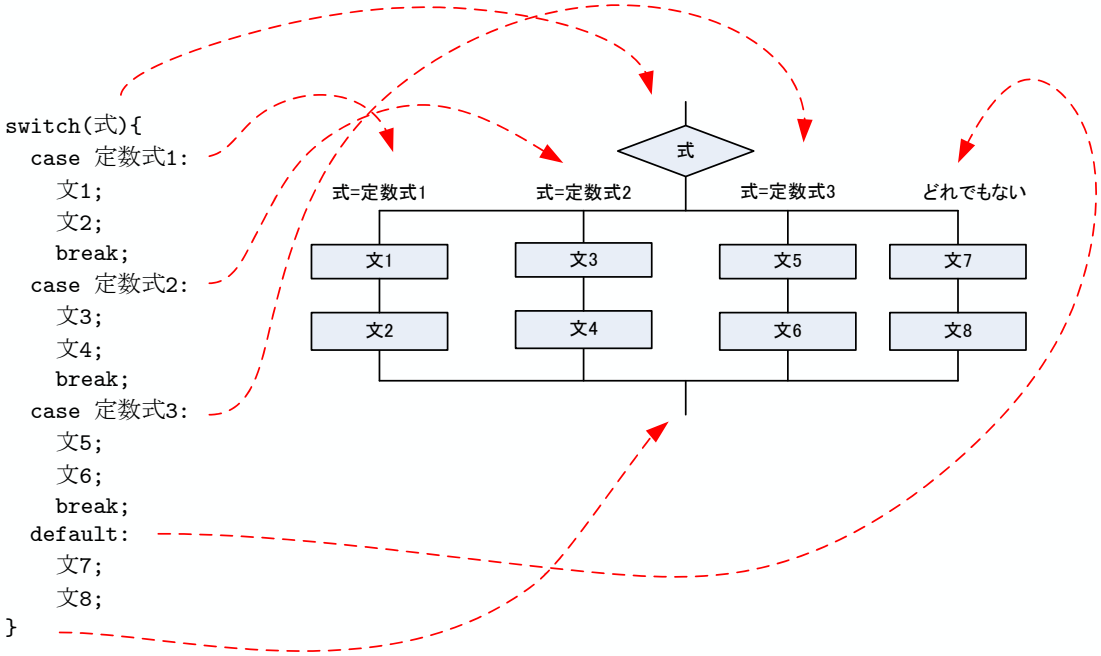

図 1: switch 文を使った構文．多くの選択肢がある場合．

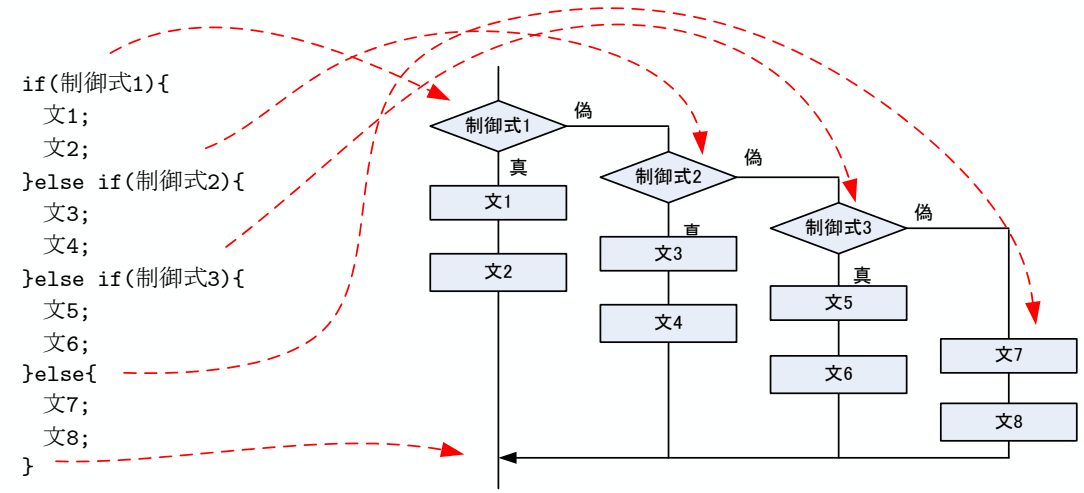

図 2: if〜else if〜else を使った多分岐

1.2 本日の学習内容

今週は同じ文—プログラムの命令—を何回も実行する繰り返し文 (ループ文) を学習する．教科書 [1] の 4 章の p.132–152 の学習を行う．そこで，諸君が身に付けるべきことを以下に示す．

- `while` を使ったループ処理が書ける．これはループ処理に先立って，判断を行い，それが正しい限り処理を続行する．
- `for` を使ったループ処理が書ける．通常，この文は繰り返し回数が予め分かっているような処理に使う．
- `do–while` を使ったループ処理が書ける．これは，一度ループ処理を行い，処理の後に繰り返しの判断を行う．

C 言語では 3 通りの繰り返し文が用意されているが，プログラムが分かりやすいものを使えば良い．

2 繰り返し処理

ここでは，先に示した 3 通りのの繰り返し処理の構文を学習する．3 通りの構文で同じ内容のプログラムを示し，その違いを学習する．ここで，例に示しているプログラムは，全て同一で，以下のようになっている．

- 「1 からいくつまで加算しますか?」と質問を行い，ユーザーはそれに答える．
- そして，1 からユーザーが答えた整数までの和を計算する．すなわち，以下の sum を計算する．

$$sum = 1 + 2 + 3 + 4 + 5 + 6 + 7 + \cdots + \text{ユーザーが入力した整数}$$

2.1 `while` 文

2.1.1 プログラム例

リスト 1 に `while` 文を使った例を示す．その動作は，以下のとおり．

1. 「1 からいくつまで加算しますか?」と質問を行い，ユーザーはそれに答える．入力した値は，`imax` に格納される．
2. 変数 `sum` と `i` に 0 と 1 を代入している．これを初期化と言う．
3. `while` を使ったループ処理のブロックでは，`i <= imax` が正しい限り，以下の文を繰り返し実行する．ただし，一度実行すると，`i <= imax` が正しいか否か，再度判断を行う．
   - `sum` に `i` の値を加えている．`sum += i` は `sum = sum+i` と同じ．教科書 [1] の p.87 を見よ．
   - `i` の値を 1 増加させている．`i++`は `i=i+1` と同じ．これをインクリメントと言う．教科書 [1] の p.135–137 を見よ．

リスト 1: while 文を使った整数の加算プログラム

```
1  #include <stdio.h>
2
3  int main(void){
4    int i, imax;
5    int sum;
6
7    printf("1からいくつまで加算しますか?\t");
8    scanf("%d", &imax);
9
10   sum = 0;
11   i = 1;
12
13   while(i<=imax){
14     sum += i;
15     i++;
16   }
17
18
19   printf("sum = %d\n", sum);
20
21   return 0;
22 }
```

2.1.2 while 文の書き方

これは，前判定繰り返しである．ループのブロックを実行する前に判断を行う—のでそのように呼ばれる．次に述べる for 文も前判定繰り返し文であるが，予め繰り返し回数が分からないときには，while 文が使われることが多い.「条件式が正しければ，ループ繰り返す．条件式が誤りになれば，そのループから抜け出す」という構文である．次のように，書く．

書式

```
while(継続条件式){
   文 1;
   文 2;
   文 3;
}
```

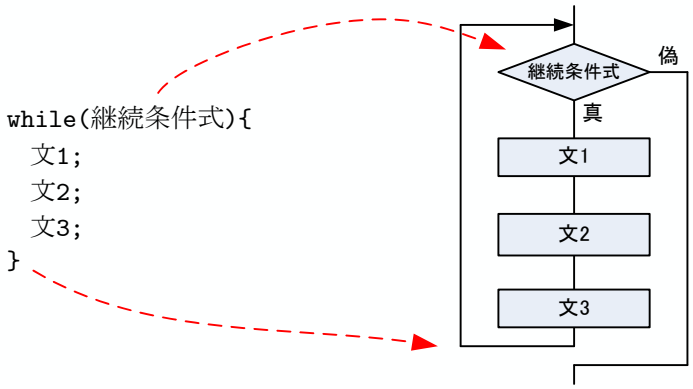

図 3: while を使ったループ処理

2.2 for 文

2.2.1 プログラム例

リスト 2 に for 文を使った例を示す．その動作は，以下のとおり．

1. 「1 からいくつまで加算しますか?」と質問を行い，ユーザーはそれに答える．入力した値は，imax に格納される．
2. 変数 sum に 0 を代入し，初期化を行っている．
3. for を使ったループ処理のブロックでは，次のように動作する．
   - 初期値として，i に 1 を代入している．
   - 継続条件式 i <= imax が正しければ，中括弧で囲まれた繰り返しのブロックを実行する．誤りであれば，for のループから抜け出る．
   - 再設定式である i++により，i の値を 1 増加させる．
   - 継続条件式に戻り，繰り返す．

リスト 2: for 文を使った整数の加算プログラム

```
1  #include <stdio.h>
2
3  int main(void){
4    int i, imax;
5    int sum;
6
7    printf("1からいくつまで加算しますか?\t");
8    scanf("%d", &imax);
```

```
 9
10    sum = 0;
11
12    for(i=1; i<=imax; i++){
13      sum += i;
14    }
15
16
17    printf("sum = %d\n", sum);
18
19    return 0;
20  }
```

2.2.2 for 文の書き方

while とは異なり，繰り返しの回数が予め分かっているとき，for 文がつかわれる．これも，前判定繰り返し文となっている．実際の動作は図 4 のフローチャートを見て理解してほしい．動作の順序は，初期値設定 → 継続条件式 → ループブロックの処理 → 再設定式 → 継続条件式 → ループブロックの処理となる．初期条件式は，最初の 1 回のみで，継続条件式が正しい限り，ループを繰り返す．

書式

```
for(初期値設定式; 継続条件式; 再設定式){
   文 1;
   文 2;
   文 3;
}
```

これは,「継続条件が正しい限り，文 1 と文 2，文 3 を実行する」となる．もし，制御式が誤り (偽) であれば，これら文は実行されず，ブロックの外側に出る．図 4 にこの構文のフローチャートを示す．

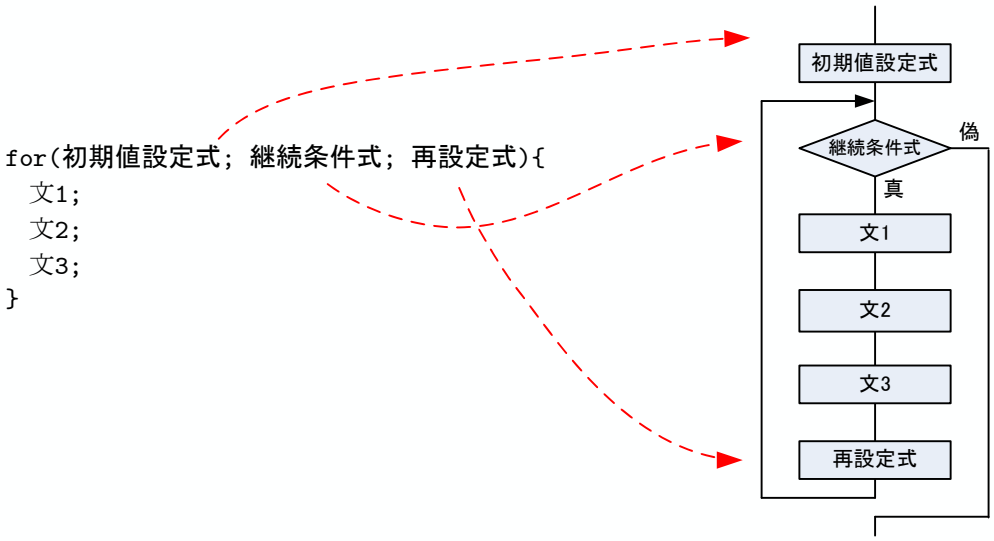

図 4: for を使ったループ処理

2.3 do–while 文

2.3.1 プログラム例

リスト 3 に do–while 文を使った例を示す．その動作は，以下のとおり．

1. 「1 からいくつまで加算しますか?」と質問を行い，ユーザーはそれに答える．入力した値は，imax に格納される．
2. 変数 sum と i に 0 と 1 を代入し，初期化を行っている．
3. do–while を使ったループ処理のブロックでは，次のように動作する．
   - sum に i の値を加える．
   - i の値をインクリメント—1 加算—する．
   - 継続条件式が正しければ，ループを繰り返す．誤りであれば，do–while のループから抜け出す．

リスト 3: do–while 文を使った整数の加算プログラム

```
1 #include <stdio.h>
2
3 int main(void){
4   int i, imax;
5   int sum;
```

```
6
7     printf("1からいくつまで加算しますか?\t");
8     scanf("%d", &imax);
9
10    sum = 0;
11    i = 1;
12
13    do{
14      sum += i;
15      i++;
16    }while(i<=imax);
17
18
19    printf("sum = %d\n", sum);
20
21    return 0;
22  }
```

2.3.2 do–while 文の書き方

これは後判定繰り返しで，予め繰り返し回数が分からないときに使われることが多い.「ループ内を実行し，継続条件式が正しければ，さらにループを繰り返す．条件式が誤りになれば，そのループから抜け出す」という構文に使われる．どのような場合でも，必ずループが 1 度は実行されるところが，前判定繰り返しと異なる．do–while 文は，次のように，書く．

書式

```
do{
   文 1;
   文 2;
   文 3;
}while(継続条件式);
```

これは,「文 1 と文 2，文 3 を実行し，継続条件が正しければ，これを繰り返す」となる．もし，制御式が誤り (偽) であれば，ブロックの外側に出る．図 5 にこの構文のフローチャートを示す.

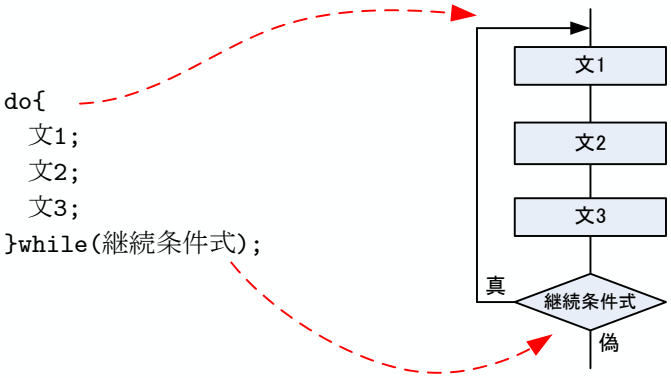

図 5: do while の後判定繰り返し文

3 プログラム作成の練習

本日の学習内容の理解を深めるために，以下の練習問題のプログラムを作成して，実行してみよ．

[**練習 1**] キーボードから奇数を読み込んで，1〜読み込んだ奇数まで合計するプログラムを作成せよ．while と for，do–while のそれぞれの構文を使った 3 種類のプログラムを作成すること．

$$sum = 1 + 3 + 5 + 7 + \ldots \tag{1}$$

[**練習 2**] 以下のような三角関数の表を作成せよ．

```
-------------------------------------------------------
deg     sin             cos             tan
=======================================================
0       0.000000        1.000000        0.000000
1       0.017452        0.999848        0.017455
2       0.034899        0.999391        0.034921
3       0.052336        0.998630        0.052408
4       0.069756        0.997564        0.069927
5       0.087156        0.996195        0.087489
6       0.104528        0.994522        0.105104

このあたりは省略

357     -0.052336       0.998630        -0.052408
358     -0.034899       0.999391        -0.034921
```

```
359      -0.017452        0.999848         -0.017455
360      -0.000000        1.000000         -0.000000
-------------------------------------------------------
```

[問 3] (復) 二次関数 $f(x) = -3x^2 + 9x - 6$ の最大値と最大になる x の値を求めよ．求める x の精度は，0.001 以内であること．これは，適当な x の初期値— 例えば $x = -100$—を決めて，x を 0.001 ずつ増加させて，$f(x)$ が最大になる値を捜す.

4　課題

次回の講義の日 (9 月 5 日) の AM8:45 までに，以下の課題をレポートとして提出すること．表紙等は，いつもの通り．表紙のタイトルは「ループ処理 (`while` と `for`，`do`–`while`)」とすること．

課題に書かれている (復) と (予) は，その問題が復習と予習のどちらに属するか—を表している．

[問 1] (復) 教科書の p.102–152 を繰り返し 3 回，読め．そして，理解できない内容があれば，クラスの分かってそうな者に聞け．さもなければ，オフィスアワーを利用して私 (山本) に質問せよ．

[問 2] (復) キーボードから，整数を入力する．0 から入力した整数までの 2 乗の和を計算するプログラムを作成せよ．プログラムは，`while` と `for`，`do`–`while` を使った 3 通りのプログラムを作成すること．

$$\mathrm{sum} = 1^2 + 2^2 + 3^2 + \cdots + N^2$$

[問 3] (復) 図 6 のように抵抗を接続した．抵抗 R の値を 0〜100[Ω] まで，0.1[Ω] きざみで変化させて，抵抗 R で発熱が最大になる値を求めるプログラムを作成せよ．繰り返し文の中で，`if` 文を上手に使って，最大発熱量になるときの抵抗値を求める．

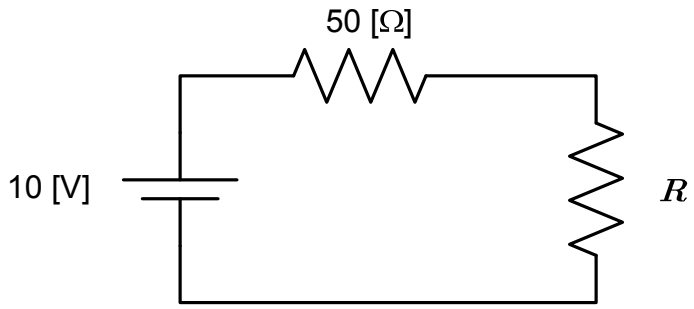

図 6: 回路

[問 4] (予) 繰り返し処理のブロックの中で，`break` や `continue` を使うとプログラムはどのように動作するか?.

参考文献

[1] 内田智史監修, (株) システム計画研究所編. C 言語によるプログラミング　基礎編　第 2 版. (株) オーム社, 2006.