ループ処理 (break, continue, goto)

山本昌志*

2006 年 10 月 12 日

概 要

繰り返しのループから抜け出すための break と，ループをスキップする continue の学習を行う．さらに，無条件分岐の goto 文の学習を行う．

1 本日の授業内容

これまで学習した 3 つのループ文—while と for，do–while—のブロックから，抜けだす方法とスキップする方法を学習する．さらに，無条件に分岐する goto 文を学ぶ．本日の学習のゴールは，以下のとおりである．

- break を使ってループから抜け出す構文が書ける．
- continue を使ってループをスキップする構文が書ける．
- goto とラベルを使って，無条件に分岐する構文が書ける．

2 前回の復習

前回の授業では，関数の最大値と，連立不等式の解を求めるプログラムの作成練習を行った．これらの問題は授業で練習を重ねたので，何も見ないで即座にプログラムが書けなくてはならない．

復習に代わって，以下の関数の最大値を求めるプログラムを何も見ないで書いてもらう．最大値を求める関数は，

$$f(x) = -x^2 + 10x + 8 + 10\sin^2 x \qquad -10 \leqq x \leqq 10 \tag{1}$$

である．計算のステップ幅 (きざみ幅)—計算精度をあらわす—は，0.00001 とする．

さあ，プログラムを作成せよ．制限時間は，15 分とする．

*独立行政法人秋田工業高等専門学校電気工学科

3　その他の制御構造

3.1　ループからの脱出

場合によっては，継続条件式が正しくても，構文から抜け出たい場合がある．このとき，break 文を使う．通常は if 文を伴って，次のように書く．

書式

```
while(継続条件式){
   文 1;
    if(制御式) break;
   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．ただし，制御式が正しければ，この構文から抜ける」となる．当然，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 1 にこの構文のフローチャートを示す．ここでも，while 文に，break 文を用いているが，for や do while 文にも使える．いずれの構文でも，break 文に出会うと，構文から抜け出る..

以下のようなプログラムが，この構文の使用例である．

```
#include <stdio.h>

int main(void){
  int i;

  for(i=0; i<10; i++){
    printf("i = %d\n",i);
    if(5<i)break;
    printf("------\n");
  }

  return 0;
}
```

実行結果は，次のようになる．

```
i = 0
------
i = 1
------
i = 2
------
i = 3
------
i = 4
```

```
------
i = 5
------
i = 6
```

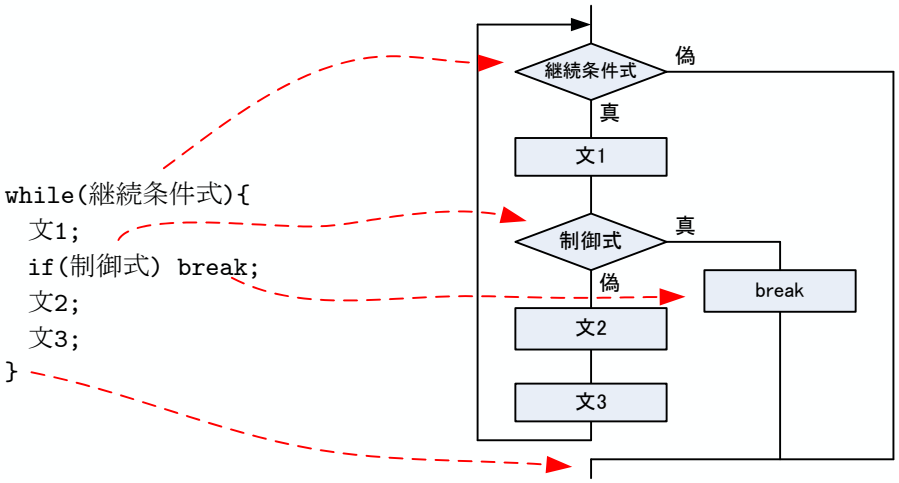

図 1: break 文を用いた構文からの脱出

3.2 ループのスキップ

場合によっては，ループブロックの文を実行させたくない場合がある．このとき，continue 文を使う．通常は if 文を伴って，次のように書く．

書式

```
while(継続条件式){
    文 1;
     if(制御式) continue;
    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．ただし，制御式が正しければ文 2 と文 3 はスキップする.」となる．当然，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 2 にこの構文のフローチャートを示す．ここでは，while 文に，continue を用いているが，for や

do while 文にも使える．いずれの構文でも，contine 文に出会うと，それ以降のループブロックが実行されない．

以下のようなプログラムが，この構文の使用例である．

```
#include <stdio.h>

int main(void){
  int i;

  for(i=0; i<10; i++){
    printf("i = %d\n",i);
    if(5<i)continue;
    printf("------\n");
  }

  return 0;
}
```

実行結果は，次のようになる．

```
i = 0
------
i = 1
------
i = 2
------
i = 3
------
i = 4
------
i = 5
------
i = 6
i = 7
i = 8
i = 9
```

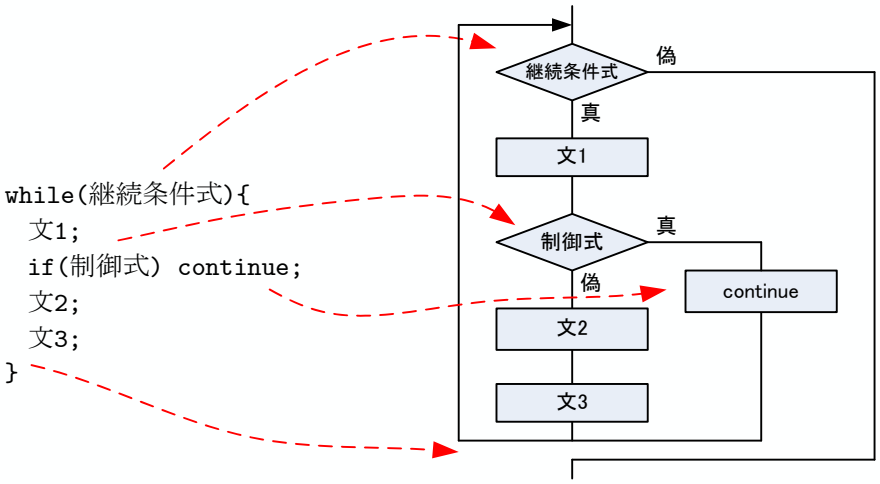

図 2: contine を使って，ループブロックをスキップする構文

3.3 無条件分岐

強制的にプログラムの制御を移す．goto 文が示すラベルに実行が移る．if 文と共に用いられることが多い．もちろん，単独で使用することも可能である．

ただし，goto 文はプログラムの流れがわかりにくくなりますので，使わないほうが良いとされている．行儀の良いプログラムを書くためには goto 文は使わないことになっている．ただし，初心者が書くような短いプログラムであれば使っても良いでろう．簡単だし，行儀が悪くてもプログラムを書くことに慣れる方が重要である．上達したら，goto 文を書かないようにすればよい．

ラベル名の後ろには，セミコロンではなく文の前に書いてコロンをつける．

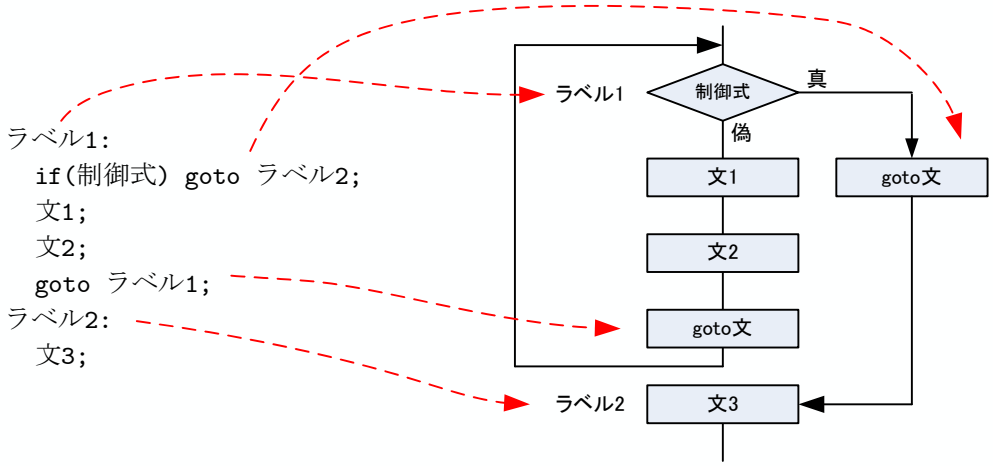

図 3: goto 文とラベルによる制御

4 プログラム作成の練習

[練習 1] 関数

$$f(x) = -2x^2 + 4x + 6\cos x \qquad -100 \leqq x \leqq 100 \tag{2}$$

が 0 を越える左端の x の値を表示せよ．x の精度は 0.0001 とすること．break 文を使って，プログラムを作成すること．

[練習 2] 1〜1000 のそれぞれの整数の約数の和を示せ．例えば，6 の場合，その約数は $\{1, 2, 3, 6\}$ なので，その和は 12 となる．

5 課題

次回の講義の日 (10 月 20 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 (break, continue, goto))」とすること．

[問 1] (復) 教科書の p.102–159 を読め．

[問 2] (復) 整数の 1〜10000 の全ての素数を書き出すプログラムを作成せよ．素数とは，1 と自分自身でしか割り切れない数．例えば，11 は 1 と 11 でしか割り切れないので素数となる．

[問 3] (復)goto 文を使って，1〜1000 まで合計するプログラムを作成せよ．

[問 4] (予) 教科書の p.162–179 を 3 回読め．そして，以下の語句の意味を 3 行程度にまとめよ．

- 関数
- 実引数
- 仮引数
- メイン関数
- プロトタイプ宣言
- 関数の呼出
- 局所変数