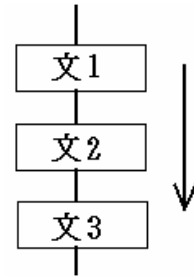


2006年10月26日(木)実施

プログラムの3基本形

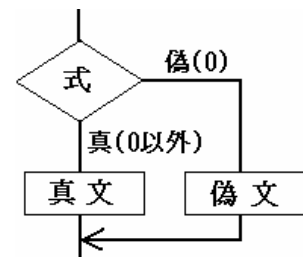
直線形

直線形のプログラムとは、プログラム中の文を処理していく順に記述したものである。これまで扱ったプログラムは全て直線形であり、最も基本的なプログラム構造と言える。プログラムの処理の流れを図示する手法の一つに**流れ図**がある。この流れ図で直線形のプログラムを表すと右のようになる。



分岐形

分岐形のプログラムとは、条件や式の値によって処理の流れを分ける構造を持つものである。分岐形の基本は**2分岐**と呼ばれる構造で、2分岐のプログラムを流れ図で表すと右のようになる。また、式の値によって、幾つもの異なる処理が必要なときには、**多分岐**というプログラム構造も利用可能である。



繰り返し形

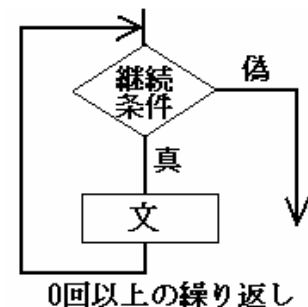
繰り返し形のプログラムとは、継続条件が満たされている間、文を繰り返して実行する構造を持つものである。(C言語以外のプログラム言語では、終了条件が満たされない間、文を繰り返して実行する構造を持つものもある) なお、繰り返し形には、右の流れ図で表される**2つの場合**がある。

1) 0回以上の繰り返し

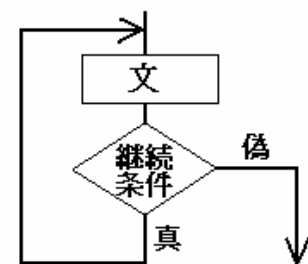
まず継続条件が判定され、真であれば文を実行する。始めから継続条件が満たされない場合には、文は全く実行されないため、**0回以上の繰り返し**と呼ばれる。

2) 1回以上の繰り返し

まず文が実行され、その後に継続条件が判定される。始めから継続条件が満たされない場合でも、最初の1回は文が実行されるため、**1回以上の繰り返し**と呼ばれる。



0回以上の繰り返し



1回以上の繰り返し

* 繰り返し形に対する流れ図に表れる閉線図形を**ループ**と呼ぶことから、繰り返し形のプログラム構造をループと称することがある。

複雑な処理は、直線形、分岐形、繰り返し形の組み合わせで実現される。プログラムの構造は、最も大掴みにした概略構造で見ると直線形となる。分岐形や繰り返し形を中間の概略構造と看做した場合、その詳細構造として、それぞれの構造の流れ図で「文」と書かれた箇所に、分岐形や繰り返し形を埋め込んだ構造も可能である。

2分岐のプログラム

if 文

C 言語で 2 分岐のプログラムを実現するための文として、if 文が用意されている。if 文の構文は次のようになる。

```
if ( 制御式 ) 真文 [ else 偽文 ]
```

ここで、[]内は省略可能であり、省略時は制御式が偽のときは何もしない場合である。また、真文または偽文を{}で囲んだブロックとして、複数の文で構成することができる。

制御式

制御式には、大小関係等を判定する**条件式**（例 $x > 0$ ）と加減乗除等の**算術式**（例 $x - 1$ ）とがある。条件式では、条件が満たされる（真となる）場合には、値が 1 となり、条件が満たされない（偽となる）場合には、値が 0 となる。算術式では、値が 0 以外の場合、真の扱いとなり、値が 0 の場合、偽の扱いとなる。

条件式で用いられる**関係演算子**及び**等価演算子**を次に挙げる。

関係演算子	書式	意味
<	$x < y$	x が y より小さければ真、それ以外は偽
>	$x > y$	x が y より大きければ真、それ以外は偽
<=	$x <= y$	x が y より小さいか、両者が等しければ真、それ以外は偽
>=	$x >= y$	x が y より大きいか、両者が等しければ真、それ以外は偽

等価演算子	書式	意味
==	$x == y$	x と y とが等しければ真、それ以外は偽
!=	$x != y$	x と y とが等しくなければ真、それ以外は偽

また、複数の条件式を組み合わせるために用いられる**論理演算子**を次に挙げる。

論理演算子	書式	意味
&&	<u>式 1</u> && <u>式 2</u>	<u>式 1</u> 及び <u>式 2</u> が共に真であれば真、それ以外は偽
	<u>式 1</u> <u>式 2</u>	<u>式 1</u> または <u>式 2</u> のどちらか一方が真であれば真（両者が真の場合も含む）、それ以外は偽
!	! <u>式</u>	<u>式</u> が真であれば偽、 <u>式</u> が偽であれば真

例題 1

次のプログラムを入力し、翻訳・編集して実行形式のファイルを作成し、実行せよ。ここで、ソースプログラム名は prog4-1.c とする。また、実行に際しては、次の 3 通りの入力を行うこと。

- 1) prog4-1 data1 data2
- 2) prog4-1 data1
- 3) prog4-1

```
/* prog4-1.c */
```

```
#include <stdio.h>

int main(int argc, char *argv[])
{
    printf("argc の値は%d です。¥n", argc);

    printf("argv[0]が指している文字列はプログラム名「%s」です。¥n", argv[0]);

    printf("argv[1]が指している文字列は第1プログラム引数「%s」です。¥n", argv[1]);

    if (argc > 1)
        printf("argv[2]が指している文字列は第2プログラム引数「%s」です。¥n", argv[2]);
    else
        printf("利用法： プログラム名 第1プログラム引数 第2プログラム引数¥n");

    return 0;
}
```

例題 2

次のプログラムを入力し、翻訳・編集して実行形式のファイルを作成し、実行せよ。ここで、ソースプログラム名は prog4-2.c とする。

```
/* prog4-2.c */
#include <stdio.h>
#include <stdlib.h>

void wa(int, int);
void sa(int, int);
void seki(int, int);
void shou(int, int);
void amari(int, int);

int main(int argc, char *argv[])
{
    int x, y;

    if (argc != 2)
    {
        printf("利用法： ¥n prog4-2 +¥n prog4-2 -¥n prog4-2 x¥n prog4-2 /¥n");
        printf(" prog4-2 %%¥n");
        exit(1);
    }

    printf("1つ目の整数を入力してください： ");
    scanf("%d", &x);
    printf("2つ目の整数を入力してください： ");
    scanf("%d", &y);

    if (argv[1][0] == '+' && argv[1][1] == '¥0')
        wa(x, y);
    else if (argv[1][0] == '-' && argv[1][1] == '¥0')
        sa(x, y);
    else if (argv[1][0] == 'x' && argv[1][1] == '¥0')
        seki(x, y);
}
```

```
    else if (argv[1][0] == '/' && argv[1][1] == '¥0')
        shou(x, y);
    else if (argv[1][0] == '%' && argv[1][1] == '¥0')
        amari(x, y);
    else
        printf("第1プログラム引数は, +, -, x, /, %%のいずれかです。¥n");

    return 0;
}

void wa(int a, int b)
{
    printf("%d + %d => %d¥n", a, b, a+b);
}

void sa(int a, int b)
{
    printf("%d - %d => %d¥n", a, b, a-b);
}

void seki(int a, int b)
{
    printf("%d * %d => %d¥n", a, b, a*b);
}

void shou(int a, int b)
{
    printf("%d / %d => %d¥n", a, b, a/b);
}

void amari(int a, int b)
{
    printf("%d %% %d => %d¥n", a, b, a%b);
}
```

【解説】

1. exit(1)は異常終了状態の形式を返して、プログラムを終了させるライブラリ関数で、利用するためには、stdlib.hを組み込んでおく必要がある。
2. if文の偽文にif文が埋め込まれた多重のif文が用いられている。
3. 等価演算子(==)は論理演算子(&&)より優先度が高い。
4. argv[1][0]は第1プログラム引数の1文字目を指す。
5. プログラム実行時に、prog4-2 * とした場合、'*'は任意の文字列を表すワイルドカードとして扱われるため、このプログラムでは、積を呼び出すための第1プログラム引数を'x'とした。

演習1

ガイドコメントに従って0点から100点までの点数を入力すると、60点未満の場合は「不可」、60点台は「可」、70点台は「良」、80点台は「優」、90点以上は「秀」と画面に表示するプログラムを作成せよ。ここで、ソースプログラム名はex4-1.cとする。