

5.2 浮動小数点型

- 浮動小数点型は、小数点付きの数や、整数型に収まらない大きな数を扱うときに使う。
- 浮動小数点型には誤差が含まれるので、ループカウンタ等には使わないこと^{*16}。
- float 型

- サイズは 4 バイトで有効桁数は 6 桁
- 内部表現例^{*17} : 1.75×2^{-1}

S	E	M
0	01111110	1100000 00000000 00000000

S: 符号部 (正負を表現), E: 指数部 ($(127)_{10} = (01111111)_2$ を 0 とする二進整数), M: 仮数部 (固定小数点の二進数)

- double 型
- サイズは 8 バイトで有効桁数は 14 桁
- C における浮動小数点表現の標準的な型
- ソース内の定数 3.14 や 1. などは double 型と解釈される。

5.3 型と演算

- 例題:
クラスの平均点を求めるプログラムを作りなさい。任意の学生数に対応でき、クラス平均は小数以下第 2 位まで表示すること。-1 の入力で終了すること。

■リスト 5.3 クラス平均プログラム

```
/* Class average program */
#include <stdio.h>

int main()
{
    double average; /* double 型 (浮動小数点型) 変数の宣言 */
    int counter, grade, total;

    /* 初期化フェーズ */
    total = 0;
    counter = 0;
```

^{*16} 浮動小数点数は近似値なので、float 型などの浮動小数点型変数をループカウンタに使うと、カウンタの値が不正確になり、ループが正確に終了しないことがある

^{*17} IEEE754 規格の浮動小数点数の意味は $(-1)^S \times (1 + M) \times 2^{E-127}$

```

/* 処理フェーズ */
printf("Enter grade, -1 to end: ");
scanf("%d", &grade);

while (grade >= 0) {
    total += grade;
    counter++;
    printf("Enter grade, -1 to end: ");
    scanf("%d", &grade);
}

/* 終了フェーズ */
if (counter > 0) {
    average = (double)total / counter;
    printf("Class average is %.2f\n", average);
}
else
    printf("No grades were entered\n");

return 0;
}

```

実行結果

```

Enter grade, -1 to end: 90
Enter grade, -1 to end: 85
Enter grade, -1 to end: 63
Enter grade, -1 to end: 98
Enter grade, -1 to end: 85
Enter grade, -1 to end: -1
Class average is 84.20

```

- `average = (double) total / counter;`
 - コンピュータは同じ型のもの同士の演算だけを行うことができ、演算結果も同じ型になる
 - `total` と `counter` は `int` 型なので、`total / counter` の値も `int` 型（小数点以下切り捨て）
 - キャスト演算子 (*type*)：単項演算子
 - オペランドのデータ型を一時的に *type* に変える（明示的型変換）。優先順位は四則演算より高い。
 - `(double) total / counter` は「`double` 型」/「`int` 型」
 - * 二項演算子のオペランドのデータ型が異なる場合、それを一時的に同一にする型変換が自動的に行われる（暗黙的型変換）。
 - * 演算結果の型は（暗黙的型変換後の）オペランドと同じ
 - * `int` 型が `double` 型に一時変換されて除算が行われるので、結果は `double` 型

– printf("クラス平均は %.2f\n", average);

* 浮動小数点数の表示には %f を使う。表示桁数は小数点以下 6 桁

* %.2f は四捨五入して小数点以下二桁まで表示。

- これまでに登場した演算子の優先順位（上ほど高い）と結合性

演算子	結合性	グループ
()	左から右	カッコ
++ -- + - (type) sizeof	右から左	単項
* / %	左から右	乗法
+ -	左から右	加法
< <= > >=	左から右	関係
== !=	左から右	等値
&&	左から右	論理積
	左から右	論理和
?:	右から左	条件
= += -= *= /= %=	右から左	代入
,	左から右	カンマ

- 暗黙的型変換時の昇格順位（上ほど高い）と各型でよく用いられる printf, scanf の変換仕様

データ型	printf の変換仕様	scanf の変換仕様
long double	%Lf	%Lf
double	%f	%lf
float	%f	%f
unsigned long int	%lu	%lu
long int	%ld	%ld
unsigned int	%u	%u
int	%d	%d
short	%hd	%hd
char	%c	%c

- 問 i と j が、各々 3 と 2 が格納された int 型変数であるとき、次の各式の値と型は何か。

1 + i / j * 10

1 + 10 * i / j

1 + i / j * 10.0

1 + 10.0 * i / j

1 + (double) i / j * 10

1 + (double) (i / j) * 10

i = j = 1.0