計算してみよう

2-03 演算

いよいよ計算しましょう。

基本例 2-3

3人分の評点から平均点を求めて表示しましょう。

▼ 実行結果

```
          評点      評価
学生1 ：  94点      S
学生2 ：   4点      D
学生3 ：  83点      A
平均点：  60.3点
学生数：  3名
```

学習

STEP 1 算術演算子による四則演算を行う

数値データに対して、四則演算を行う演算子は表13のとおりです。

▼ 表13　算術演算子

算術演算子	演算の種類	一般形	例	結果
単項の＋	そのままの値	＋オペランド	10	10
単項のー	負の値	ーオペランド	-10	-10
＋	たし算	オペランド1＋オペランド2	10 ＋ 20	30
ー	ひき算	オペランド1ーオペランド2	10 - 20	-10
*	かけ算	オペランド1*オペランド2	10 * 2	20
／	わり算	オペランド1／オペランド2	10 / 3	3
			10.0/3	3.3333…
%	割り算の余り	オペランド1%オペランド2 (オペランドは整数のみ)	10 % 3	1

▶**オペランド**とは、演算子の左または右にあって、その演算が施されるデータのことです。例えば、「x+y」では、xとyがオペランドです。

▶**単項**とは、オペランドが一つしかない演算子のことです。

演算の順は数学と同じで、加減算より乗除が先に行われます。それより「()」が優先されるのも数学と同じです。

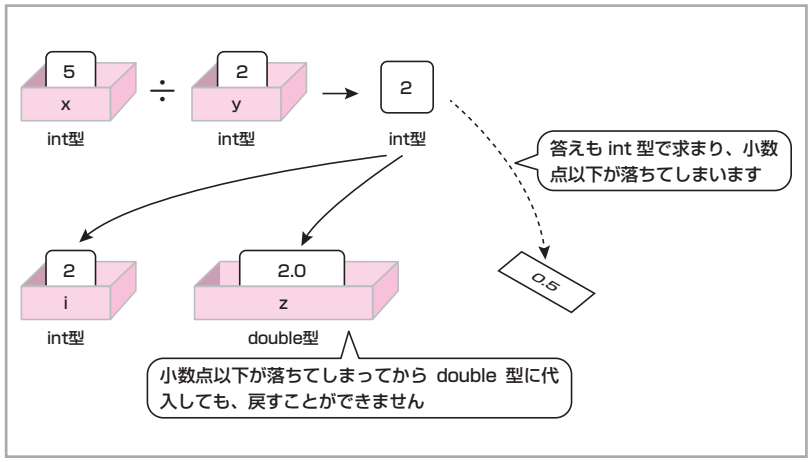
```
2  + 3 * 4 = 14
(2 + 3) * 4 = 20
```

演算は、同じ型どうしでしかできません。異なる型の間で演算を行おうとするとき、支障のない方の型が変換されて、型を揃えてから演算が行われ、結果もその型になります。これを**暗黙の型変換**といいます。

整数どうしで割り算を行うと、結果も整数となり、小数点以下が切り捨てられてしまいます。これをdouble型の変数に代入しても、切り捨てられた小数部は切り捨てられたまま、小数点以下が0の小数になります。

▶int型の数値をdouble型に変換したとき、数値に変化はありません。しかし、double型の数値をint型に変換すると、小数点以下は丸められ、数値が変化してしまいます。つまり、int型からdouble型への変換は支障ありませんが、double型からint型への変換は支障があるということです。したがって、int型とdouble型とで演算を行いたい場合には、int型をdouble型に変換してdouble型どうしで演算を行います。結果もdouble型で求まります。

▼ **図11**　int型どうしの演算



▶電卓では、整数÷整数を行っても、結果が実数で表示されます。しかし、C言語では、演算の結果はオペランドの型に依存します。

STEP 2　キャスト演算子で一瞬の変身を行う

それでは、int型どうしで割り算を行って、double型の結果を得るには、どうしたらよいでしょうか? double型の結果を得るためには、少なくともオペランドのどちらかはdouble型でなければなりません。それは、ずっとdouble型であり続ける必要はなく、演算を行う瞬間だけ、double型であれば解決します。そこで、本来int型のデータを演算が行われる瞬間だけ、暫定的に型変換する演算子が用意されました。これを**キャスト演算子**といい、次のように記述します。

```
（演算を行う間だけ変換する型）変数名
```

▼ 例

```
int x = 5;
int y = 2;
double    z1 , z2;
z1 = x / y;            //z1の値は2.0となる
z2 = (double)x / y;    //z2の値は2.5となる
```

CD-ROM

元のファイル…sample2_2o.c
完成ファイル…sample2_3k.c

プログラム例

これまでは、平均点を自分で計算して代入していましたが、いよいよコンピュータに計算してもらいましょう。式を追加してください。応用例2-2にはなかった合計点を格納する変数も追加しています。

```
/*****************************************
    演算        基本例2-3
*****************************************/
#include <stdio.h>
#define    N    3                //学生の人数を定数として定義

int main(void)
{
    //変数の宣言
    int       hyouten1 = 94;     //学生1の評点
    int       hyouten2 = 4;      //学生2の評点
    int       hyouten3 = 83;     //学生3の評点
    int       gokei;             //評点の合計点
    double    heikin;            //評点の平均点
    char      hyouka1 = 'S';     //学生1の評価
    char      hyouka2 = 'D';     //学生2の評価
    char      hyouka3 = 'A';     //学生3の評価

    //評点の合計点と平均点の計算
    gokei = hyouten1 + hyouten2 + hyouten3;
    heikin = (double)gokei / N;    //平均点を計算

    //コマンドプロンプト画面に表示
    printf("        評点    評価¥n");
    printf("学生%d ：%3d点    %c¥n", 1, hyouten1, hyouka1);
    printf("学生%d ：%3d点    %c¥n", 2, hyouten2, hyouka2);
    printf("学生%d ：%3d点    %c¥n", 3, hyouten3, hyouka3);
    printf("平均点：%5.1f点¥n", heikin);
    printf("学生数：%2d名¥n", N);

    return 0;
}
```

合計を記録する変数を追加

平均点の初期化削除

3人分の評点の合計を求め、平均点を計算
平均点を小数で求めるためにキャスト演算子が必要

プログラミングアシスタント　コンパイルエラーになった方へ

```
int main(void)
{
    //変数の宣言
    int     hyouten1 = 94;    //学生1の評点
    int     hyouten2 = 4;     //学生2の評点
    int     hyouten3 = 83;    //学生3の評点
    double  heikin;           //評点の平均点
    char    hyouka1 = 'S';    //学生1の評価
    char    hyouka2 = 'D';    //学生2の評価
    char    hyouka3 = 'A';    //学生3の評価

    //合計と平均の計算
    gokei = hyouten1 + hyouten2 + hyouten3;
    heikin = (double)gokei / N;
        :
```

int　gokei;　//評点の合計点

ここで、use of undeclared identifier 'gokei'というエラーが発生していたら、変数gokeiが宣言されているかどうかを確認してください
エラーが発生している文だけを見ていては、解決することができないケースです

プログラミングアシスタント　正しい結果が表示されなかった方へ

▼ **実行結果**

```
        評点     評価
学生1 ： 94点     S
学生2 ：  4点     D
学生3 ： 83点     A
平均点： 60.0点
学生数： 3名
```

平均点の小数点以下が0になっている

```
//合計と平均の計算
gokei = hyouten1 + hyouten2 + hyouten3;
heikin = gokei / N;
    :
    :
(double)
```

gokeiもNもどちらもint型であり、このまま演算すると、結果もint型となります。すでに小数点が落ちてしまった結果をdouble型の変数に代入しても、小数点以下の値を復活させることはできません

キャスト演算子を用いて、どちらか一方を一時的にdouble型に変換すると、暗黙の型変換により、残りもdouble型に変換され、double型の演算を行うことができます。その結果、double型の結果が得られます。演算の前に変換することが必要です