

プログラミング演習 I 第 8 回

関数 (1)

今日の目標 : 関数を作ってみる。

1. 関数の基本構造

C 言語では一まとめの処理を関数にまとめておき、順に関数を呼び出して処理を実行する。

関数の長点として以下が挙げられる

(1) 同じ処理は入力データだけ変えて同じ関数を呼び出して繰り返しできる。同じ処理のプログラムを何回も書かなくて良い。

(2) まとまりのある処理を一つの関数にまとめていけば、後で見直しても理解しやすい。

main もまた関数であるが、プログラムを起動すると最初に実行される関数と思えばよい。後で理解しやすいプログラムは、main には変数の宣言と関数の呼び出ししかなく、その他の実行文が極めて少ないことが多い。関数の基本形は以下のようになる。

戻り値の型 関数名 (引数のリスト)

```
{
    変数宣言
    実行文
    ...
    return 戻り値;
}
```

引数 (ひきすう) は関数に与えるデータのことであり、全く引き数がない関数も定義できる。この場合、引き数として void を指定する。

例 : `int function(void)`

戻り値とは関数実行の結果、関数の最後で `return` 文で関数を呼び出した側に戻すデータのことであり、宣言された型のデータを返す。戻り値のない関数も定義できる。この場合は戻り値の型は `void` と宣言する。

例 : `void function2(int a, int b, ...)`

関数の具体例 :

```
int square_add(int a, int b)
{ /* 引数の 2 乗和を帰す */
    int result;
    result = a * a + b * b;

    return result;
}
```

関数の呼び出し例：

```
int main(void)
{
    int x, y, z1, z2;
    ...
    x = 2;
    y = 3;
    z1 = square_add(x, y);    ←変数を引数とする
    z2 = square_add(2, 3);    ←値を引数とする。どちらでも可能
    ...
}
```

2. 関数のプロトタイプ宣言

関数を使うには main 関数の前、include 文の後に関数の名前、戻り値の型、引数とその型を宣言しておく。

宣言文：

戻り値の型 関数名 (*引数の型のリスト*) ;

例：

```
int square_add(int, int);
int square_add(int a, int b); ←こちらでも良い。
```

関数宣言をそのままコピーし、最後に ; を付け加えるだけでもよい。

プロトタイプ宣言を使ったプログラムの例：

```
#include <stdio.h>
int square_add(int a, int b); /* プロトタイプ宣言 */

int main(void)
{
    int x, y, z1, z2;
    ...
    x = 2;
    y = 3;
    z1 = square_add(x, y);
    ...
}

int square_add(int a, int b)
{ /* 引数の 2 乗和を帰す */
    int result;
    result = a * a + b * b;

    return result;
}
```

注意：プロトタイプ宣言文、関数の定義文、関数の呼び出し全てで戻り値の型、引数の数と型が一致しないとコンパイルエラーとなる。

3. 本日の内容の副教材

本日の講義の内容の一部をカバーする動画が下記にあるので、見てみよう。

<https://paiza.jp/works/c/primer/beginner-c7>

有料部分も含まれるので、WebClass の「副教材の視聴方法」を読んで、手続きをして視聴してください。

4. 本日の演習

- (1) 引数を 2 乗する関数 `square` と 3 乗する関数 `cube` を作り、`main` 関数からキー入力した数値（実数型）の 2 乗、3 乗各々を計算し、表示せよ。計算は乗算を 2 回、または 3 回して計算すること。標準関数は使わないこと。`square`、`cube` の引数、戻り値はいずれも `float` で宣言するものとする。

表示例：

```
$ lab9_1[Enter]
Enter a number: 0.1[Enter]
a^2 = 0.010000
a^3 = 0.001000
$
```

- (2) 2 つの引数をもつ関数 `float npow(float a, int b)` を作りなさい。`npow` は a の b 乗 (a^b) を変数 a を b 回乗算して計算する (`for` などを用いよ)。標準関数は使わないこと。`npow` を用いて (1) と同じ計算、表示を行なうプログラムを作成せよ。(関数は `npow` 一つしか定義しないこと、2 乗用と 3 乗用関数を 2 つ定義するのではない)。

- (3) `int` 型引数 s から 1 までの 2 乗級数和を計算する関数 `squareSum(int s)` ($= \sum_{i=1}^s i^2$) ,

3 乗級数和を計算する関数 `cubicSum(int s)` ($= \sum_{i=1}^s i^3$) を定義し、これらを用いて `main`

関数で入力した値 s から 1 までの 2 乗級数和と 3 乗級数和を計算し、表示せよ。関数から各級数和を戻して、`main` 関数で出力のこと。各関数の戻り値は `float` 型とせよ。また、これら関数に必要な 2 乗、3 乗の計算には (2) で定義した `npow` を用いること。

表示例：

```
$ lab9_3[Enter]
Enter a number: 10[Enter]
square series sum = 385.000000
cubic series sum = 3025.000000
$
```

- (4) 半径 x を実数として入力し、その円周 (`float circum(float x)`) と面積 (`float cArea(float x)`)、また球の体積 (`float gVolume(float x)`) と表面積 (`float gSurface float x`) を計算して実数の戻り値とする関数を定義し、これらを用いて円周、円面積、球体積、球表面積表示するプログラムを作成せよ。計算は実数で行い、円周、面積は全体で 7 桁、小数点以下 2 桁として表示すること。円周率はプログラムの先頭に `#include <math.h>` を付け加えれば、プログラムの中で `M_PI` で参照できる。また、これら関数に必要な 2 乗、3 乗の計算には (2) で定義した `npow` を用いること。

表示例：

```
$ lab9_4[Enter]
```

```

Enter radius:1.2[Enter]
Circumference =    7.54
Area =    4.52
Volume =    7.24
Surface area =    18.10
$

```

- (5) 第 7 回の演習課題 (2) と同じ動作を関数を用いて実装する。身長、体重も同じデータを用いる。さらに、以下の 2 関数を定義して用いよ。

(i) `float bmiCalc(float height, float weight)`

引数としてメートル表示の `float` 型の身長とキログラム表示の `float` 型の体重を入力し、これより BMI 指数 ($\text{BMI 指数} = \text{体重 [kg]} \div \text{身長 [m]} \div \text{身長 [m]}$) を計算して `float` 型で戻り値とする。

(ii) `int healthLevel(float height, float weight)`

引数としてメートル表示の `float` 型の身長とキログラム表示の `float` 型の体重を入力し、これより関数 `bmiCalc` を呼び出し BMI 指数を計算して、健康度判定指数 (BMI 指数が 18.5~25.0 の場合が標準として "0"、18.5 未満がやせすぎ "-1"、25.0 以上が肥満 "+1") を `int` 型で戻り値とする。必ずこの関数内で関数 `bmiCalc` を呼び出すこと。

表示は第 8 回の課題 2 と同じ表示とする。

- (4) のプログラムを提出せよ。

提出期限は今週木曜 (6/18) 16 時までとします。WebClass を使って提出してください。

ファイル名は (学籍番号)_(演習の回)_(課題番号).c と名前を付けて提出してください。例えば学籍番号 191234 の人が第 2 回の課題 (4) のプログラムを提出する場合は 19123_2_4.c と名前を付けて WebClass で提出してください。

注意：提出レポートで明らかに動作しないプログラムは減点するので、**十分デバッグ、動作確認** すること。再提出はありません。期限外は一切受け付けませんので、注意。

5. 来週の予習

教科書 (明快入門 C) の第 7 章 (関数) をもう一度読んでおくこと。また、11 章 (標準ライブラリ関数) に目を通しておくこと。