ソフトゼミ A 2016 第 6 回

関数

今回は関数の作り方と使い方を解説します。関数をつくることによって、プログラムのソースをよりわかりやすいものにすることができます。実は、これまでに説明した printf や scanf は関数なのです。このようなあらかじめ用意されている関数はライブラリ関数と呼ばれます。

➢ 関数とは

関数とは、ある特定の計算や出力・入力などのプログラムをまとめて、名前を付けたもののことです。例えば printf は「二重引用符の中の文字列を%d や%f などの書式に従って画面に出力する」という風に定義されています。

➢ 関数の定義

関数を使うためにはまずは定義をしなければいけません。定義の仕方は

```
返り値の型　関数名(引数の型　引数名 1 , 引数の型　引数名 2 , …){

    (関数の内容)

    return 返り値 ;     /*←返り値がない場合は書きません*/
}
```

となります。

ところで、この形は main 文に似ています。

```
int main(void){
    …
return 0 ;
}
```

実は main 文も関数の一種なのです。

➢ 関数の使用

関数を使用するときには、使用したいところでその関数の名前と引数を書きます(関数の呼び出し)。ただしある関数を呼び出すよりも前に、その関数の定義あるいはプロトタイプ宣言をする必要があります。プロトタイプ宣言とは、関数の

定義を書く前に、その関数の「仕様」を宣言することです。宣言の仕方は

```
返り値の型　関数名(引数の型　引数１, 引数の型　引数２, …) ;
```

です。

➢ 引数と返り値

関数間では値のやり取りがされます。そのやり取りされる値が引数と返り値です。

■　引数

呼び出した側の関数から、呼び出された関数に渡される値のことです。複数設定できます。

■　返り値

呼び出された関数から、呼び出した側の関数に返される値のことです。返り値は引数と違って、1 つだけです。

実際にプログラムを見てみましょう。関数 function は「受け取った値を 2 倍にして返す関数」です。

```
#include<stdio.h>
int function(int x);      //関数プロトタイプ宣言
int main(void){
    int a＝10;                  //a の値を 10 に初期化
    printf(“before a=%d¥n”,a);
    a=function(a);              //a に、関数 function からの返り値を代入
    printf(“after a=%d¥n”,a);

    return 0;
}
int function(int x){       //関数の定義　返り値は int 型　引数は int 型 x
    x=x*2;                      //x に、x を 2 倍したものを代入
    return x;                   //x の値を返す
}
```

このプログラムを実行すると、

```
before a=10
after a=20
```

となります。関数 function が呼び出されてからの流れは以下のとおり。

6 行目で関数 function が呼び出され、この時に a の値である「10」が引数として渡される。

↓

渡された「10」という値は function 内の変数 x に代入される。その後 x の値は 2 倍されて、「20」の値が x には代入される。そして x の値である「20」が、返り値として main 関数へと返される。

↓

function から返された「20」という値が main 内の変数 a に代入される。

➢ 返り値を持たない関数(void)

関数には返り値を返さないものがあります。その場合、返り値の型は「void」となります。void は「空の」という意味です。

ではプログラムの例を見てみましょう。関数 function2 は「受け取った値を 3 倍して画面に出力する関数」です。

```
#include<stdio.h>
void function2(int a);      //関数プロトタイプ宣言
int main(void){
    int a;
    printf("a=");
    scanf("%d",&a);

    function2(a);              //a の値を関数 function2 に渡す
    printf("main a=%d¥n",a);   //main 関数における a の値を表示
    return 0;
}
void function2(int a){      //関数の定義 返り値なし→返り値の型は void.
    a=a*3;
    printf("function a=%d¥n",a);   //function2 関数での a の値を表示
}
```

main と function2 のそれぞれで宣言されている変数 a は、名前は同じですが別の変数です。そのため、「function a」はキーボードから入力した a の数値を 3 倍したものになりますが、「main a」は入力した数値そのままです。

(出力例)
a=21　　(←キーボードから入力)
function a=63
main a=21

➢ グローバル変数

関数内で宣言された変数を**ローカル変数**といいます。ローカル変数は宣言された関数内でだけで使うことができます。そのため、ローカル変数の値を外部の関数が書き換えることは基本的にはできません。(ポインタを使えば可能になりますが。) これに対して、関数の外で宣言された変数を**グローバル変数**といいます。グローバル変数はプログラム全体で共用されます。

ではプログラムの例を見てみましょう。

```
#include<stdio.h>
void function3(void);   //プロトタイプ宣言
int a;                  //グローバル変数の宣言(初期値は 0 になります)
int main(void){
    printf("a=%d¥n",a);      //a=0
    a=4;
    printf("a=%d¥n",a);      //a=4
    function3();
    printf("a=%d¥n",a);      //a=9
    return 0;
}
void function3(void){
    a=9;
}
```

グローバル変数は初期値として 0 が代入されます。ローカル変数の初期値は不定です。ある関数内にグローバル変数と同じ名前のローカル変数があった場合は、ローカル変数が優先して使われます。

グローバル変数は一見便利そうですが、どこからでも値がいじれるということは、逆に言えばどこで値が変わったのかがわかりづらくなるということです。そのため、バグが発生した場合のプログラムの修正(デバッグ)がやりにくくなります。

グローバル変数の多用はなるべく控えましょう。

➢ 練習問題

次の関数を作成してください。ただし main 文も用意して動作確認をするようにしてください。

1. 円の直径の長さを引数として受け取って、その面積を返す関数
   ただし円周率は 3.14 としてよい。
   double Circle(double d)

2. 整数の値を 3 つ、引数として受け取ってそのうち最大の値を返す関数
   int max(int a,int b,int c)

3. 整数の値を 2 つ(x,y)、引数として受け取って x の y 乗を返す関数。ただし y は 0 以上の整数とします。
   int power(int x,int y)

4. 整数の値を 4 つ引数として受け取って大きい順に出力する関数
   void order(int a,int b,int c,int d)