

プログラミング概論

<http://bit.ly/kosen01>

Week10@後期
2016/12/1

お知らせ

12/8(木)は米国出張のため休講にします。

- 先週の演習解答アップ済み
 - 講義資料配布ページ→演習問題解答→
後期第9週

本日の内容

- 講義
 - 復習: 関数の引数と戻り値
 - 関数から別の関数を呼び出す
 - ~~関数の再帰呼び出し~~(→次回)
- 演習
 - 別の関数を呼び出す練習問題

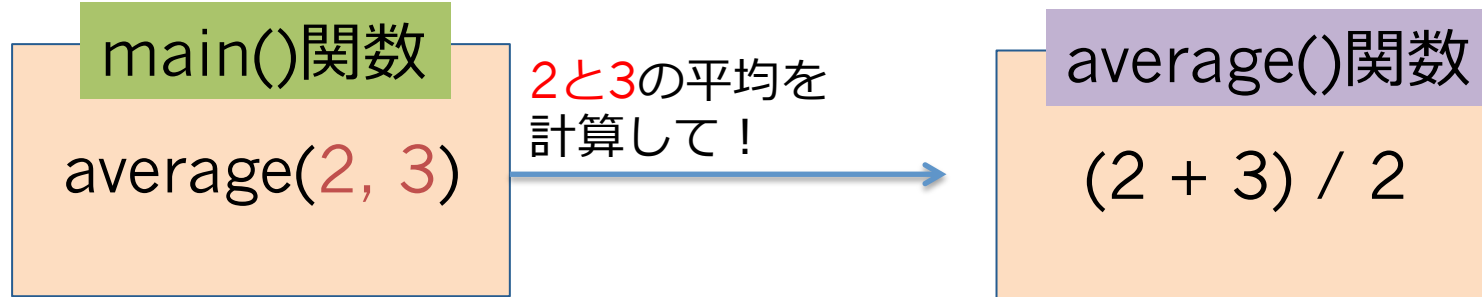
本日の内容

- 講義
 - 復習: 関数の引数と戻り値
 - 関数から別の関数を呼び出す
 - ~~関数の再帰呼び出し~~(→次回)
- 演習
 - 別の関数を呼び出す練習問題

引数と戻り値

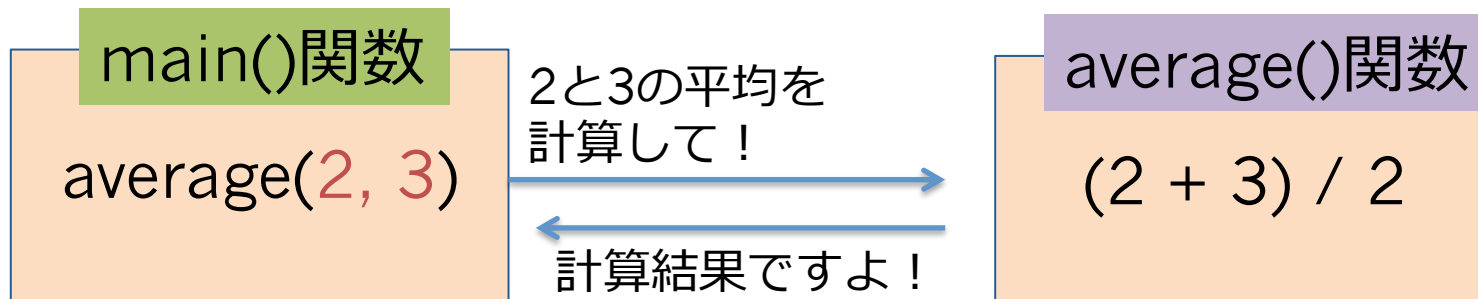
- 引数

- 関数に値を送る仕組み



- 戻り値

- 関数から値を受け取る仕組み



例3) 引数/戻り値の両方を使う

```
1 #include <stdio.h>
2 // 1. 戻り値/引数を使う場合の形式宣言
3 int average(int a, int b);
4 // 2. 戻り値/引数を使う関数の本体
5 int average(int a, int b) {
6     int result;
7     result = (a + b) / 2;
8     return result;
9 }
10 // 3. 戻り値/引数を使う関数をmain()関数から呼び出す
11 int main() {
12     printf("%d", average(2, 3));
13     return 0;
14 }
```

例3) 引数/戻り値の両方を使う

```
1 #include <stdio.h>
2 // 1. 戻り値/引数を使う場合の形式宣言
3 int average(int a, int b);
4 // 2. 戻り値/引数を使う関数の本体
5 int average(int a, int b) {
6     int result;
7     result = (a + b) / 2;
8     return result;
9 }
10 // 3. 戻り値/引数を使う関数をmain()関数から呼び出す
11 int main() {
12     printf("%d", average(2, 3));
13     return 0;
14 }
```

引数を使うには関数宣言と関数の本体の () 内に型と変数名を書く。

例3) 引数/戻り値の両方を使う

```
1 #include <stdio.h>
2 // 1. 戻り値/引数を使う場合の形式宣言
3 int average(int a, int b);
4 // 2. 戻り値/引数を使う関数の本体
5 int average(int a, int b) {
6     int result;
7     result = (a + b) / 2;
8     return result;
9 }
10 // 3. 戻り値/引数を使う関数をmain()関数から呼び出す
11 int main() {
12     printf("%d", average(2, 3));
13     return 0;
14 }
```

戻り値を使うには関数名の前にreturnする値の型名を書く。

補足：戻り値の型

```
int average(int a, int b) {  
    int result;  
    result = (a + b) / 2;  
    return result;  
}
```



returnする値が整数型
(int)ならば
int 関数名(引数) {
 return (intの値)
}

```
float average(int a, int b) {  
    float result;  
    result = (a + b) / 2;  
    return result;  
}
```



returnする値が小数型
(float)ならば
float 関数名(引数) {
 return (floatの値)
}

```
void average(int a, int b) {  
    float result;  
    result = (a + b) / 2;  
    printf("平均 : %d", result);  
}
```



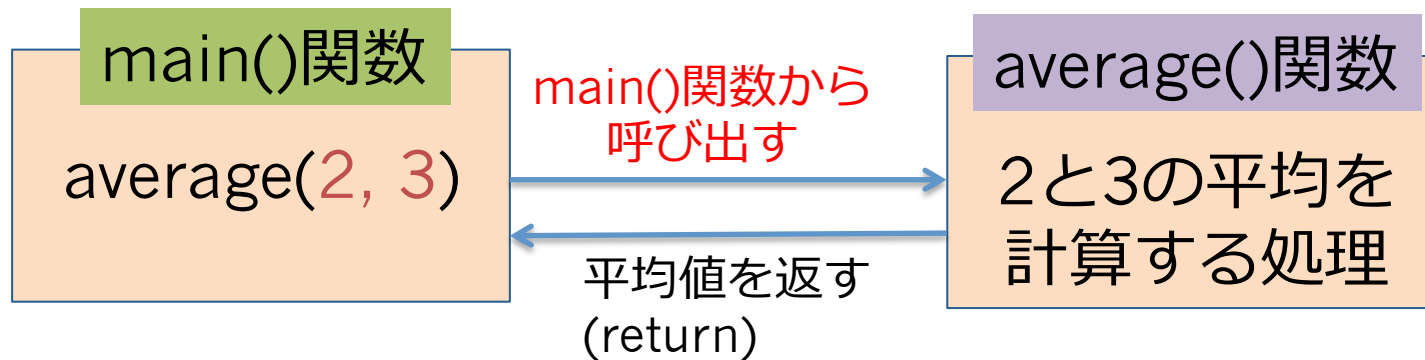
returnする値がない場合
は
void 関数名(引数) {
}

本日の内容

- 講義
 - 復習: 関数の引数と戻り値
 - 関数から別の関数を呼び出す
 - ~~関数の再帰呼び出し~~(→次回)
- 演習
 - 別の関数を呼び出す練習問題

関数同士の連携

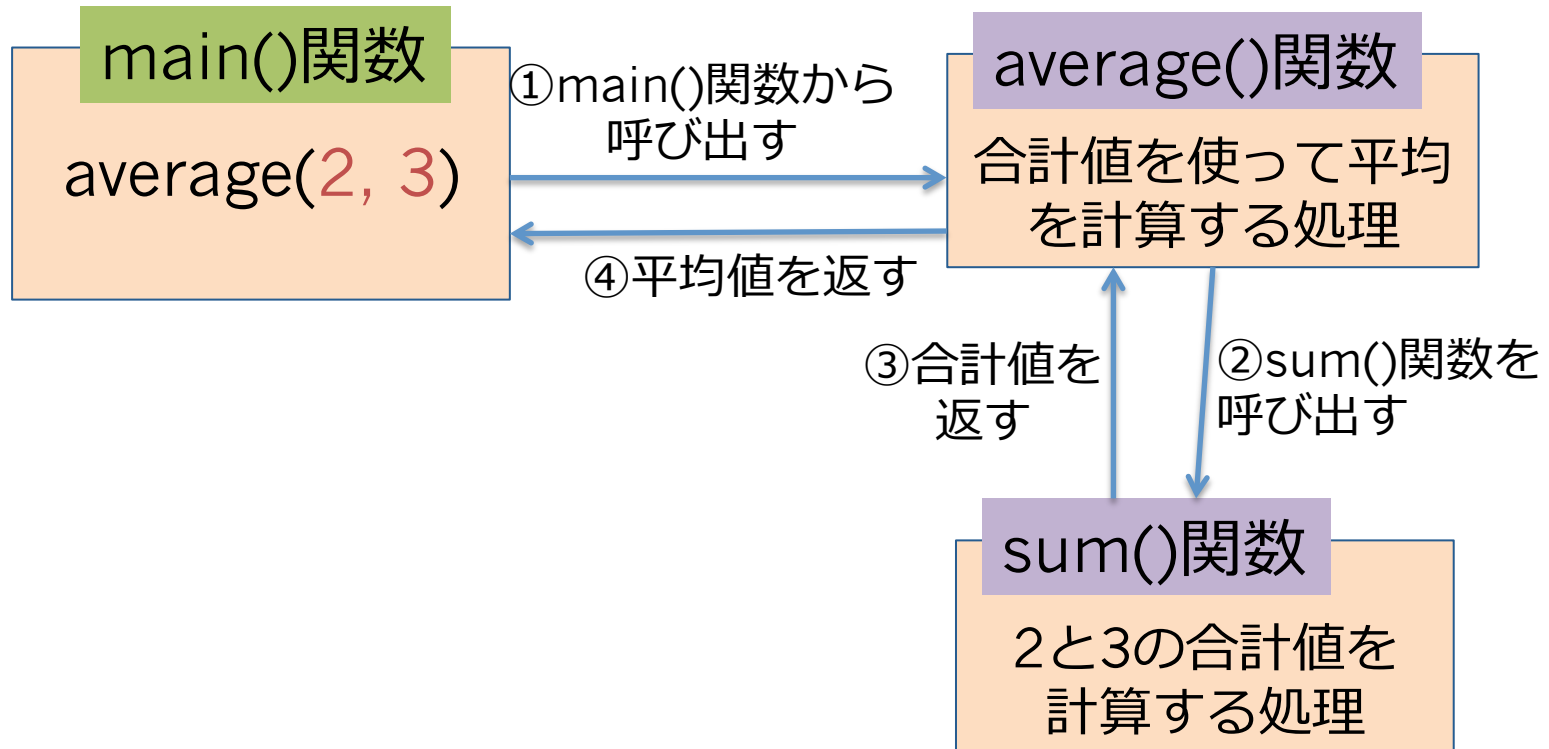
- これまで
 - main()関数からaverage()関数を呼び出す



関数同士の連携

- 関数が連携する例

1. main()関数からaverage()関数を呼び出す
2. average()関数はsum()関数を呼び出す



プログラム例1

```
1 #include <stdio.h>
2
3 int average(int a, int b);
4 int sum(int a, int b);
5
6 int average(int a, int b) {
7     int result;
8     result = sum(a, b)/2 ;
9     return result;
10 }
11
12
13 int sum(int a, int b) {
14     int result;
15     result = a + b;
16     return result;
17 }
18
19 int main() {
20
21     printf("平均値:%d\n", average(2, 4));
22     return 0;
23 }
```

プログラム例1

```
1 #include <stdio.h>
2
3 int average(int a, int b);
4 int sum(int a, int b);
5
6 int average(int a, int b) {
7     int result;
8     result = sum(a, b)/2 ;
9     return result;
10 }
11
12
13 int sum(int a, int b) {
14     int result;
15     result = a + b;
16     return result;
17 }
18
19 int main() {
20
21     printf("平均値:%d\n", average(2, 4));
22     return 0;
23 }
```

average()関数

合計値を使って平均
を計算する処理

③合計値を
返す

②sum()関数を
呼び出す

sum()関数

2と3の合計値を
計算する処理

プログラム例1

```
1 #include <stdio.h>
2
3 int average(int a, int b);
4 int sum(int a, int b);
5
6 int average(int a, int b) {
7     int result;
8     result = sum(a, b)/2 ;
9     return result;
10 }
11
12
13 int sum(int a, int b) {
14     int result;
15     result = a + b;
16     return result;
17 }
18
19 int main() {
20
21     printf("平均値:%d\n", average(2, 4));
22     return 0;
23 }
```

average()関数

合計値を使って平均
を計算する処理

③合計値を
返す

②sum()関数を
呼び出す

sum()関数

2と3の合計値を
計算する処理

1. average()関数が呼び出される。

プログラム例1

```
1 #include <stdio.h>
2
3 int average(int a, int b);
4
5 2. average()関数を上から逐次実行
6 int average(int a, int b) {
7     int result;
8     result = sum(a, b)/2 ;
9     return result;
10 }
11
12
13 int sum(int a, int b) {
14     int result;
15     result = a + b;
16     return result;
17 }
18
19 int main() {
20
21     printf("平均値:%d\n", average(2, 4));
22     return 0;
23 }
```

average()関数

合計値を使って平均
を計算する処理

③合計値を
返す

②sum()関数を
呼び出す

sum()関数

2と3の合計値を
計算する処理

1. average()関数が呼び出される。

プログラム例1

```
1 #include <stdio.h>
2
3 int average(int a, int b);
4
5 2. average()関数を上から逐次実行
6 int average(int a, int b) {
7     int result;
8     result = sum(a, b)/2 ;
9     return result;
10 }
11
12 3. sum()関数を呼び出し
13 int sum(int a, int b) {
14     int result;
15     result = a + b;
16     return result;
17 }
18
19 int main() {
20
21     printf("平均値:%d\n", average(2, 4));
22     return 0;
23 }
```

average()関数

合計値を使って平均
を計算する処理

③合計値を
返す

②sum()関数を
呼び出す

sum()関数

2と3の合計値を
計算する処理

1. average()関数が呼び出される。

プログラム例1

```
1 #include <stdio.h>
2
3 int average(int a, int b);
4
5 2. average()関数を上から逐次実行
6 int average(int a, int b) {
7     int result;
8     result = sum(a, b)/2 ;
9     return result;
10 }
11
12 3. sum()関数を呼び出し
13 int sum(int a, int b) {
14     int result;
15     result = a + b;
16     return result;
17 }
18
19 4. sum()関数を逐次実行
20
21 int main() {
22     printf("平均値:%d\n", average(2, 4));
23     return 0;
24 }
```

average()関数

合計値を使って平均を計算する処理

③合計値を返す

②sum()関数を呼び出す

sum()関数

2と3の合計値を計算する処理

1. average()関数が呼び出される。

プログラム例1

```
1 #include <stdio.h>
2
3 int average(int a, int b);
4
5 2. average()関数を上から逐次実行
6 int average(int a, int b) {
7     int result;
8     result = sum(a, b)/2 ;
9     return result;
10 }
11
12 3. sum()関数を呼び出し
13 int sum(int a, int b) {
14     int result;
15     result = a + b;
16     return result;
17 }
18
19 4. sum()関数を逐次実行
20
21 5. average()関数に戻る
22 int main()
23 {
24     printf("平均値:%d\n", average(2, 4));
25     return 0;
26 }
```

average()関数

合計値を使って平均を計算する処理

③合計値を返す

②sum()関数を呼び出す

sum()関数

2と3の合計値を計算する処理

1. average()関数が呼び出される。

プログラム例1

```
1 #include <stdio.h>
2
3 int average(int a, int b);
4 int sum(int a, int b);
5
6 int average(int a, int b) {
7     int result;
8     result = sum(a, b)/2 ;
9     return result;
10 }
11
12
13 int sum(int a, int b) {
14     int result;
15     result = a + b;
16     return result;
17 }
18
19 int main() {
20
21     printf("平均値:%d\n", average(2, 4));
22     return 0;
23 }
```

average()関数

合計値を使って平均

6. sum()関数がreturnした値
を使って計算し、resultに代入

③合計値を
返す

②sum()関数を
呼び出す

sum()関数

2と3の合計値を
計算する処理

プログラム例1

```
1 #include <stdio.h>
2
3 int average(int a, int b);
4 int sum(int a, int b);
5
6 int average(int a, int b) {
7     int result;
8     result = sum(a, b)/2 ;
9     return result;
10 }
```

7. main()関数に戻る

```
11
12
13 int sum(int a, int b) {
14     int result;
15     result = a + b;
16     return result;
17 }
18
19 int main() {
20
21     printf("平均値:%d\n", average(2, 4));
22     return 0;
23 }
```

average()関数

合計値を使って平均
を計算する処理

③合計値を
返す

②sum()関数を
呼び出す

sum()関数

2と3の合計値を
計算する処理

プログラム例1

```
1 #include <stdio.h>
2
3 int average(int a, int b);
4 int sum(int a, int b);
5
6 int average(int a, int b) {
7     int result;
8     result = sum(a, b)/2 ;
9     return result;
10 }
11
12
13 int sum(int a, int b) {
14     int result;
15     result = a + b;
16     return result;
17 }
18
19 int main() {
20
21     printf("平均値:%d\n", average(2, 4));
22     return 0;
23 }
```

average()関数

合計値を使って平均
を計算する処理

③合計値を
返す

②sum()関数を
呼び出す

sum()関数

2と3の合計値を
計算する処理

7. printf()で平均値を表示して終了！

関数の連携

- これまではmain()関数から自作関数 (average()など)を呼び出し
- 関数から別の関数を呼び出すことも可能

演習0

- プログラム例1を記述し動作を確認しなさい

演習1-1

- この演習ではタクシーの走行距離や収益などを管理する「タクシー」アプリを作成します。
- このタクシーは1km走ると100円の収益を獲得し、乗せた人数×走った距離分のガソリンを消費します。
- このアプリが完成したときの動作は以下のコマンドで確認できます。
\$ ~ishigaki/taxi
- はじめに「taxi.c」というファイル名で空のC言語プログラムを作成しなさい

演習1-2

- 下のように収益合計、ガソリン残量、距離を管理する整数型のグローバル変数、money, gasoline, distanceを宣言しなさい。

```
#include <stdio.h>

int money; // 収益合計
int gasoline; // ガソリン残量
int distance; // 距離

int main() {
    /*
     * ここにグローバル変数money, gasoline
     * distanceをそれぞれ0, 100, 0で
     * 初期化する処理を書く */

    return 0;
}
```

演習1-3

- main()関数内にmoney, gasoline, distanceにそれぞれ0, 100, 0を代入しなさい

```
#include <stdio.h>

int money; // 収益合計
int gasoline; // ガソリン残量
int distance; // 距離

int main() {
    /*
     * ここにグローバル変数money, gasoline
     * distanceをそれぞれ0, 100, 0で
     * 初期化する処理を書く */

    return 0;
}
```

演習1-4(show()関数の作成)

- 現在のタクシーの状況を表示するshow()関数を作成し、main()関数から呼び出さない
 - 1. 関数の形式は以下のようにする。
void show();
 - 2. 関数の本体は以下のように記述し、main()関数からshow()関数を呼び出さない。

```
// 演習1-4: show()関数の本体
void show() {
    printf("現在のタクシーの状況を表示します。 \n");
    printf("獲得した収益合計: %d円 \n", money);
    printf("ガソリン使用量合計: %dリットル \n", gasoline);
    printf("走行距離合計: %dkm \n", distance);
}
```

演習1-5(run())関数の作成)

- 以下の動作をするrun()関数を記述しなさい。
 1. 関数の形式は以下(aは走行距離)
// aは走った距離、bは乗客の人数
void run(int a, int b);
 2. printf()関数を用いて「(aの値)kmを(bの値)人乗せて走りました。」と表示する
 3. グローバル変数moneyに引数a * 100の計算結果を+=演算子を用いて加算する
 4. グローバル変数gasolineから引数aの値を-=演算子を用いて減算する
 5. グローバル変数distanceに引数aの値を+=演算子を用いて加算する

演習1－6(run()関数の呼び出し)

- main()関数からrun()関数を以下のように呼び出さない。

run(10, 2); // 10kmを2人乗せて走る

演習1－7(run()とshow()の連携)

- run()関数の中にshow()関数を呼び出す処理を追加し、タクシーが走る度にタクシーの状況が表示されるようにしなさい。

次回

- 12/8(木)は休講です。
- 関数から自分自身の関数を呼び出す「再帰処理」を学習します。