

関数プロトタイプ宣言, 乱数の仕組みとシード, 標本抽出のプログラム

樋口さぶろお

龍谷大学工学部数理情報学科

計算科学☆実習 B E02(2019-04-16 Tue)

最終更新: Time-stamp: "2019-04-15 Mon 09:16 JST hig"



<http://hig3.net>

関数プロトタイプ宣言

関数はどこで定義する?

OK

```

1  int a(int x, int y){
2      return something;
3  }
4  void b(int x){
5      return; /* nothing*/
6  }
7  main(){
8      int x=0,y=2,z;
9      z=a(x,y);
10     b(x);
11     return;
12 }
```

Error

```

1  main(){
2      int x=0,y=2,z;
3      z=a(x,y); /* エラー cc: 「こ
4      b(x);
5      return;
6  }
7  int a(int x, int y){
8      return something;
9  }
10 void b(int x){
11     return; /* nothing*/
12 }
```

困るケース

```

1  int a(int x, int y){
2      return b(y);
3  }
4  void b(int x){
5      a(x, x);
6      return; /* nothing*/
7  }
8  main(){
9      int x=0, y=2, z;
10     z=a(x, y);
11     b(x);
12     return;
13 }

```

関数プロトタイプ宣言

変数だって代入(定義)の前に宣言.

void printf(...) の関数プロトタイプ
宣言は stdio.h 内にある.

一般的なやりかた

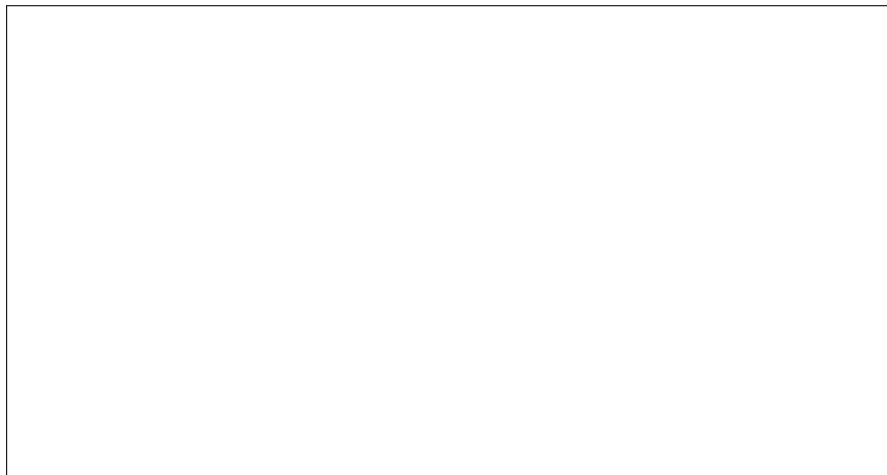
```

1  /*関数プロトタイプ宣言*/
2  int a(int v1, int v2);
3  void b(int v3);
4
5  /*関数の宣言 兼 定義*/
6  main(){
7      int x=0, y=2, z;
8      z=a(x, y);
9      b(x);
10     return;
11 }
12
13 /*関数の定義*/
14 int a(int x, int y){
15     return b(y);
16 }
17 void b(int x){
18     a(x, x);
19     return; /* nothing*/
20 }

```

擬似乱数列生成の仕組みとシード

擬似乱数列 = 'ほぼ' ランダムな数列



2 回続けて `int rand()` を呼んで得られる 2 個の数の分布は独立であるかのように考えてる (けど… そこが擬似).

(標本抽出のたびに別々の)seed を指定することが必要

- seed に応じて, 毎回異なる乱数列が得られる.
- 特定の乱数列に対する動作を再現できる. デバッグでは必須.
- seed を適切に設定すると, 複数回の実行で, 別々の (独立な) 標本抽出が行える.

E02-Q1

Quiz(rand() の振る舞い)

次のプログラムで, seed を無作為に入力するとき, A が出力される確率は?

```
1  int getrandom(double y){
2      if( y<1/3.0 ){
3          return 0;
4      }else{
5          return 1;
6      }
7  }
8
9  int main(){
10     int seed;
11     scanf("%d",&seed);
12     srand(seed);
13     if( getrandom(getuniform())==getrandom(getuniform()) ){
14         printf("A\n");
15     }
16     return 0;
17 }
```

- ① 0
- ② 1/2
- ③ 5/9 に近い
- ④ 6/9 くらい
- ⑤ 1

擬似乱数による標本抽出

ソースコード 1: 擬似乱数

```

1  /*
2  rand1.c --- -1 or +1 を確率1/4, 3/4で選ぶ乱数
3  Time-stamp: "2018-04-17 Tue 19:18 JST hig"
4  */
5  #define _CRT_SECURE_NO_WARNINGS // Visual C++用おまじない
6  #include <stdio.h>
7  #include <stdlib.h> /* srand(), rand() を使うのに必要 */
8
9  /* 関数プロトタイプ宣言 */
10 int getuniform();
11 int getrandom(double y);
12
13 int main(){
14     int seed; /* 擬似乱数のシード */
15     int n; /* カウンタ 標本内通し番号*/
16     int nmax=100; /* 擬似乱数を得る回数=サンプルサイズN */
17
18     scanf("%d",&seed);
19     srand(seed); /* シードの設定 */
20     for(n=0;n<nmax;t++){ /* 数式とnは1ずれてる*/
21         /* srand(seed); /* ここに置くと? */
22         printf("%d,%d\n",getrandom(getuniform()));
23     }
24     return 0;
25 }
26 /** [0,1) 一様擬似乱数を返す */
27 double getuniform(){
28     return rand()/(RAND_MAX+1.0);
29 }
30 /** -1 or +1 を確率1/4, 3/4 で返す乱数 */
31 int getrandom(double y){
32     if( y < 0.25 ){
33         return -1;
34     } else {
35         return +1;
36     }
37 }

```

予習復習問題のやり方+今後の予定

Learn Math Moodle

<https://learn.math.ryukoku.ac.jp/moodle>



お知らせ

教科書と (PC につながる) イヤフォン用意してください。

- 2019-04-18 木 5 講義 1-534
- Math ラウンジ 月火水木昼 1-614
- 樋口オフィスアワー火 5(1-507/1-542)

Moodle App for iOS/Android



URL をきかれたら <https://learn.math.ryukoku.ac.jp/moodle> で登録。