

物理学情報処理演習

5. C++言語② 変数・演算・制御文

ver20170509_3

2017年5月9日

5.1 変数

5.2 演算

5.3 制御文

参考文献

- やさしいC++ 第4版 高橋 麻奈 (著)
ソフトバンククリエイティブ
- プログラミング言語C++第4版
ビャーネ・ストラウストラップ, Bjarne Stroustrup, 柴田 望洋
- Numerical Recipes: The Art of Scientific Computing, Third Edition in C++

身内賢太郎

レポート提出: fsci-phys-jouhou@edu.kobe-u.ac.jp

5.1 変数

• 5.1.1 変数

- 変数:”値”を入れておく箱
- 変数は、
 - 名前と型を持ち(変数定義)
 - 指し示した値を見たり(参照)
 - 値を変えたり(代入)できる。

- 使用される変数は全て宣言する。

参考file:triangle_1.cxx

演習5.1.1

triangle_1.cxxをダウンロード、実行してみよう。

実行時にパラメータ50[†]、プログラム中で20を渡す。

```
$/triangle_1 50
```

次ページからの説明を読みながら、このプログラムの内容を理解しよう。

```
#include <iostream>
#include <stdlib.h>
#include <string.h>
#include <math.h>
using namespace std;
int main(int argc, char *argv[]){
    //calculate the area of a triangle.
    //command line parameter is angle C(degree).
    int i=1;
    double a,b,c;
    double A,B,C,S;
    char message[128];

    a=10.0;

    strcpy(message,"input length b >");
    cerr <<"STEP"<< i <<": ";
    cout <<message;
    cin>>b;
    C=atof(argv[1]);
    cout << "a="<<a << " b="<<b<<" C="<<C<<"(degree)"<<endl;
    S=a*b*sin(M_PI*C/180)/2.;
    cout << "area="<<S<<endl;
    return 0;
}
```

triangle_1.cxx :2辺とその間の角度から三角形の面積を求めるプログラム。

[†] コマンドライン引数:4.3.3参照

5.1.2 定数と変数、代入

- 定数
 - 整数: 12, 0xFF
 - 浮動小数点整数: 123.4, 1.5e-10
 - 文字定数: 'a', '\0', '\x
 - 文字列: "Hello"
 - 文字定数と文字列は異なる
 - 'x': 文字定数
 - "x": 文字列(「x」と「\0」から構成) \0:ヌル文字
- 定数と変数
 - 例


```
int i, j;           変数定義
i = 3; 変数に定数を代入
j = i * i;         変数を参照、演算をし、結果を変数に代入
9 = j; × 定数には代入できない
```
- 変数に関するマクロ
 - #define MAX 1000000
 - #define g 9.80665
 - #define BELL '\x7'

```
#include <iostream>
#include <stdlib.h>
#include <string.h>
#include <math.h>
using namespace std;
int main(int argc, char *argv[]){
  //calculate the area of a triangle.
  //command line parameter is angle C(degree).
  int i=1;
  double a,b,c;
  double A,B,C,S;
  char message[128];

  a=10.0;

  strcpy(message,"input length b >");
  cerr <<"STEP"<< i <<": ";
  cout <<message;
  cin>>b;
  C=atof(argv[1]);
  cout << "a="<<a << " b="<<b<<" C="<<C<<"(degree)"<<endl;
  S=a*b*sin(M_PI*C/180)/2.;
  cout << "area="<<S<<endl;
  return 0;
}
```

- 宣言時に代入することも、宣言と独立に代入することも可能。
 - 一般には = を用いて代入するがstrcpyを用いた文字列への代入、cinによる標準入力からの代入なども可能。
- triangle_1.cxx :2辺とその間の角度から三角形の面積を求めるプログラム。

5.1.3 変数の型

整数型

- short 16bit $-2^{15} \sim 2^{15}-1$ 先頭bitは符号
- int 32 bit $-2^{31} \sim 2^{31}-1$ 先頭bitは符号
- long 32(or 64) bit 先頭bitは符号
- unsigned short 16bit $0 \sim 2^{16}-1$
- unsigned int 32bit : $0 \sim 2^{32}-1$
- unsigned long 32(or 64)bit

浮動小数点型

- float 32bit
- double 64bit
- long double 32bit

文字型

- char 8bit 1文字
- 8bitのデータ 整数と見られることもできる。
- signed char 8bit $-2^7 \sim 2^8-1$ 先頭bitは符号
- unsigned char 8bit $0 \sim 2^8-1$

```
#include <iostream>
#include <stdlib.h>
#include <string.h>
#include <math.h>
using namespace std;
int main(int argc, char *argv[]){
    //calculate the area of a triangle.
    //command line parameter is angle C(degree).
    int i=1;
    double a,b,c;
    double A,B,C,S;
    char message[128];

    a=10.0;

    strcpy(message,"input length b >");
    cerr <<"STEP"<< i <<": ";
    cout <<message;
    cin>>b;
    C=atof(argv[1]);
    cout << "a="<<a << " b="<<b<<" C="<<C<<"(degree)"<<endl;
    S=a*b*sin(M_PI*C/180)/2.;
    cout << "area="<<S<<endl;
    return 0;
}
```

triangle_1.cxx : 2辺とその間の角度から三角形の面積を求めるプログラム。

5.2 演算

• 5.2.1 算術演算

- +(和) -(差) *(積) /(除)
%(整数同士の除算で余り)
- *, /, % が+, -よりも優先される。

• 5.2.2 数学演算

- 使用時には<math.h>をincludeする
- sin(x) 正弦 (radで与える)
- cos(x) 余弦(radで与える)
- tan(x) 正接(rad)
- asin(x) arc sin
- acos(x) arc cos
- atan(x) arc tan
- exp(x) 指数関数
- sqrt(x) 平方根
- log(x) 自然対数(底e)
- log10(x) 常用対数(底10)
- pow(x,y) xのy乗
- fabs(x) 絶対値
- M_PI 円周率

```

#include <iostream>
#include <stdlib.h>
#include <string.h>
#include <math.h>
using namespace std;
int main(int argc, char *argv[]){
//calculate the area of a triangle.
//command line parameter is angle C(degree).
int i=1;
double a,b,c;
double A,B,C,S;
char message[128];

a=10.0;

strcpy(message,"input length b >");
cerr <<"STEP"<< i <<": ";
cout <<message;
cin>>b;
C=atof(argv[1]);
cout << "a="<<a << " b="<<b << " C="<<C<<"(degree)"<<endl;
S=a*b*sin(M_PI*C/180)/2.;
cout << "area="<<S<<endl;
return 0;
}

```

triangle_1.cxx :2辺とその間の角度から三角形の面積を求めるプログラム。

5.2.3 演算 (算術以外)

- 代入演算子 =
 - 関係演算子 ==(等しい), >(より大きい), <(より小さい), >=(以上), <=(以下)
 - 論理演算子 &&(且つ(AND)), ||(または(OR)), !=(等しくない)
 - ビット演算子 &(ビットごとのAND), |(ビットごとのOR), ^(ビットごとのXOR)
-
- 関係演算子、論理演算子、ビット演算子は真の場合に1を返す。

5.3 制御文

5.3.1 繰り返し文

○for文: 決まった回数繰り返す

```
for (初期化; 条件; 終端処理){  
    実行文;  
}
```

○while文 条件が満たされる間繰り返す

```
while (条件) {  
    実行文;  
}
```

$i=i+1$; と $i++$;

$i=i-1$; と $i--$;

$y+=a$; と $y=y+a$;

は同じ意味

$\#t$ はtabで列区切りの記号

$\#tvy$ は $\#t$ で列区切り、その後に変数 vy の値を出力する。

```
#include <iostream>  
#include <stdlib.h>  
#include <string.h>  
#include <math.h>  
#define g 9.80665 //m/s^2  
#define T_MAX 10 //sec  
using namespace std;  
int main(int argc, char *argv[]){  
    //calculate the freefall  
    int t;  
    double y,vy;  
    y=0;  
    vy=0;  
    for(t=0;t<T_MAX+1;t++){  
        cout << "t="<<t<<"\#tvy="<<vy<<"\#ty="<<y<<endl;  
        y+=vy; // same as y=y+vy  
        vy-=g; // same as vy=vy-g  
    }  
    return 0;  
}
```

freefall_1.cxx

```
#include <iostream>  
#include <stdlib.h>  
#include <string.h>  
#include <math.h>  
#define g 9.80665 //m/s^2  
#define T_MAX 10 //sec  
#define T_STEP 1 //sec  
using namespace std;  
int main(int argc, char *argv[]){  
    //calculate the freefall  
    double t;  
    double y,vy;  
    t=0;  
    y=0;  
    vy=0;  
    // for(t=0;t<T_MAX+1;t++){  
    while(t<T_MAX+1){  
        cout <<  
        "t="<<t<<"\#tvy="<<vy<<"\#ty="<<y<<endl;  
        vy-=g*T_STEP; // same as vy=vy-g  
        y+=vy*T_STEP; // same as y=y-vy  
        t+=T_STEP;  
    }  
    return 0;  
}
```

freefall_2.cxx

freefall_1.cxx freefall_2.cxx :
自由落下を計算するプログラム。

5.3.2 if 文

○条件の真偽で判定

```
if(条件){
    実行文(条件が真のとき);
}
else{
    実行文(条件が偽のとき);
}
```

○条件分岐が3つ以上のとき、else if を使って

```
if (条件1){
    実行文1(条件1が真のとき);
}
else if (条件2){
    実行文2(条件2が真のとき);
}
else {
    実行文3(条件1,2 が共に偽のとき);
}
```

**$a < b$ かつ $b < c$ という条件を課す際
 $a < b < c$ としてはいけない。**

```
#include <iostream>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#define g 9.80665 //m/s^2
#define T_MAX 10 //sec
#define Y_MIN -100
using namespace std;
int main(int argc, char *argv[]){
    //calculate the freefall
    int t;
    double y,vy;
    y=0;
    vy=0;
    for(t=0;t<T_MAX+1;t++){
        cout << "t="<<t<<"\tvy="<<vy<<"\tvy="<<y<<endl;
        if(y<Y_MIN){
            break;
        }
        else{
            y+=vy; // same as y=y+vy
            vy-=g; // same as vy=vy-g
        }
    }
    return 0;
}
```

freefall_3.cxx 自由落下を計算するプログラム。
freefall_1.cxxにY_MIN以下に達したら計算を終了する機能を加えた。

5.3.3 switch 文

○switchによる分岐も可能

```
switch (変数) {
    case 値1:
        実行文1;
        break;
    case 値2:
        実行文2;
        break;
    default:
        実行文3
        break;
}
```

caseの値として取れるのは、整数もしくは文字

```
#include <iostream>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#define g 9.80665 //m/s^2
#define T_MAX 10 //sec
#define Y_MIN -100
using namespace std;
int main(int argc, char *argv[]){
    //calculate the freefall
    int t;
    double y,vy;
    y=0;
    vy=0;
    for(t=0;t<T_MAX+1;t++){
        switch(t){
            case 0:
                cout << "initial paarameters:¥t";
                break;
            default:
                break;
        }
        cout << " t= <<t<< " ¥tvy= <<vy<< " ¥ty=" <<y<<endl;
        if(y<Y_MIN){
            break;
        }
        else{
            y+=vy; // same as y=y-vy
            vy-=g; // same as vy=vy-g
        }
    }
    return 0;
}
```

freefall_4.cxx

freefall_4.cxx :freefall_3.cxxに初期値を明示する機能をswitch分を使って加えたもの。

- 一般的注意

- θ や π 、日本語は機種依存があるので、プログラム中で使用しない。
- バックスラッシュが入力できないときには、「システム環境設定」→「キーボード」→「入カソース」→”¥”キーで入力する文字 を「バックスラッシュ」にする。

課題5:C言語②

ソースコードファイル名:2017_jouhou_05_学籍番号の下4桁.cxx
出力結果ファイル名:2017_jouhou_05_学籍番号の下4桁.txt

質点の運動を計算する、以下の仕様を持つものプログラムを製作し、ソースコード及び出力結果を提出せよ。

- ①初速度 v [m/s]と仰角[degree]をコマンドライン引数として入力する。
- ②初速度を30m/s、仰角30度として原点から照射する。照射時刻 $t=0$ から $y(t+dt)=y(t)+v(t)dt$ に従って逐次計算を行い、水平面に達するまで計算を行う。(最大時刻は50とする。)
初速度、仰角(degree)、円周率、仰角(rad)、水平方向初速、垂直方向初速を出力fileに出力する。
その後、各時刻での t,x,y,vx,vy をtab区切りでこの順番で出力fileに出力する。
- ③手計算で水平面に達するまでの時間を計算、得られた結果と比較し、メールの本文に考察せよ。

課題提出

- 宛先 fsci-phys-jouhou@edu.kobe-u.ac.jp
- 件名 2017-report05_学籍番号の下4桁
- 本文 学籍番号と名前
- 添付ファイル:
 - 2017_jouhou_05_学籍番号の下4桁.cxx
 - 2017_jouhou_05_学籍番号の下4桁.txt
- 締め切り 2017年5月16日(火)13:00