システム科学科 情報処理演習テキスト 1 – 変数 –

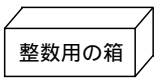
原案:佐藤宏介 修正: 升谷 保博, 庄野 逸, 鳩野逸生, 橋本守, 才脇直樹，田中宏喜，日浦慎作

ポイント 1: C 言語によるプログラミングの基本構造

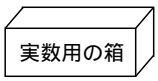
テンプレート 1_1

```
#include <stdio.h>     /* 標準入出力 (standard I/O) を利用する */
int main( void )
{
  /* まず , 変数宣言を行う */
  /* 次に実行文を書いていく */
  return 0;
}
```

ポイント 2: '変数' とは数値を扱うための入れ物 (箱)



- 表現できる範囲:
  $-2147483648 \sim 2147483647$
- 整数定数: 小数点なし
  (例: 123456, -987654)
- 厳密な計算ができるが表現できる範囲は広くない



- 表現できる範囲:
  $\pm 2.22507 \times 10^{-308} \sim \pm 1.79769 \times 10^{+308}$
- 実数定数: 小数点をつけるか、e や E を使って仮数部と指数部で表現
  (例: 10.0、-12.3、9.8765e4、-5.432E-10)
- 表現できる範囲は広いが、厳密な計算はできない。有効数字は約 13 桁

ポイント 3: 変数を使用するときには、あらかじめ "変数の名前 (変数名)" と変数が取り扱うデータの種類 (整数型、実数型など) を "宣言" する必要がある

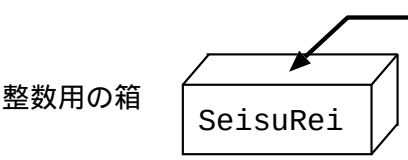
ポイント 4: 変数の名前 (変数名) は英文字から始める。途中に数字やアンダースコア "_" を用いても良い。変数名は長くても意味のある名前にした方が良い場合が多い。

整数の宣言: `int` 整数変数の名前 1, 整数変数の名前 2, ···, 整数変数の名前 N;
例: `int SeisuRei, age_women, i, j;`

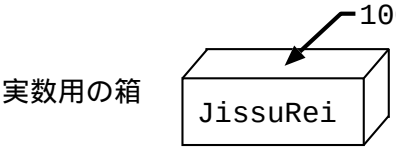
実数の宣言: `double` 実数変数の名前 1, 実数変数の名前 2, ···, 実数変数の名前 N;
例: `double JissuRei, MinValue, MaxValue, x, y;`

ポイント 5: '変数' に数値を入れる (代入する) には、記号 "=" を用いる

ポイント 6: 演算子は "+" (足し算)、"-" (引き算)、"*" (掛け算)、"/" (割り算)、"%" (剰余) などが使用可能である。計算式のなかで括弧 "(" と ")" で囲われた部分は先に実行される。詳しくは参考書を参照。



↔ `SeisuRei = 200-77;`



↔ `JissuRei = 100.0 + 23.0;`

ポイント 7: 変数宣言や計算式の後のセミコロン ";" を忘れないこと

ポイント 8: プログラム中でキーボードからの数値の入力および文字列を出力するには以下の文を用いる

- 整数の入力: `scanf( "%d", &整数変数名 );`

例:
```
int n;
scanf( "%d", &n );
```

- 実数の入力: `scanf( "%lf", &実数変数名 );`

例:
```
double r;
scanf( "%lf", &r );
```

- 文字列の出力: `printf( "文字列\n" );`

例:
```
printf("This is a pen.\n");
```

- 整数変数値の出力:
  `printf( "%d", 変数名 );`

例:
```
int a = 3;
printf( "a = %d\n", a );
```

- 実数変数値の出力:
  `printf( "%f", 変数名 );`

例:
```
double r = 3.1415;
printf( "r = %f\n", r );
```

ポイント 9: 整数同士の計算は整数、実数同士の計算は実数、整数と実数の計算は実数となる (2/3 は 0) . また、実数を整数に代入すると小数点以下が切り捨てられる.

example1_1.c

```
#include <stdio.h>

 int main(void)
{
     int a,b,c,d,e,f;

     a = 3+2*4;
     b = 6/4;
     c = 1/2*3;
     d = 12/(2*3);
     e = a%d;
     f = 1.2;

     printf("a l %d \n", a);
     printf("b l %d \n", b);
     printf("c l %d \n", c);
     printf("d l %d \n", d);
     printf("e l %d \n", e);
     printf("f l %d \n", f);

     return 0;
}
```

ポイント 10: 定数は、"#define 定数の名前 数値" という文を使ってわかりやすい名前に置き換えると良い (代入はできない)

例:
```
#define PI 3.141592  /* 円周率 */
double r, S;
r = 2.0;
S = PI * PI * r; /* 面積 */
```

ポイント 11: sin(正弦)、cos(余弦) などの数学関数を用いるときは math.h を #include する必要がある。(example1_1.c 2 行目参照)

ポイント 12: 数学関数を用いるときは、オプション "-lm" が必要 (テキスト 0 ポイント 4 参照)

例題: キーボードから入力された数値に基づいて、計算を行うプログラム。　ここでは「辺の長さ length と両端の角の角度 (度単位)angleDeg を入力すると、その面積を求める」という問題が与えられたとする

example1_2.c

```
/*一辺とその両端角の角度から面積を計算する*/
#include <stdio.h>
#include <math.h>                       /*sin, cos 等　数学関数を使用するために必要*/
#define PI 3.14159265358979323846       /*π をマクロ定義*/

int main(void)
{
    double len, angleDeg1, angleDeg2, area;
    double angle1,angle2;

    printf("一辺とその両端角の角度から面積を計算します\n");
    printf("一辺の長さを入力して下さい\n");
    scanf("%lf", &len);
    printf("一方の角の角度 ( 度 ) を入力して下さい\n");
    scanf("%lf", &angleDeg1);
    printf("他方の角の角度 ( 度 ) を入力して下さい\n");
    scanf("%lf", &angleDeg2);

    angle1=angleDeg1/180.0*PI;        /*angleDeg1 をラジアンに変換*/
    angle2=angleDeg2/180.0*PI;        /*angleDeg2 をラジアンに変換*/

    area=len*len*sin(angle1)*sin(angle2)/2.0/sin(angle1+angle2);
    printf("面積は%f です\n",area);

    return 0;
}
```