Programming I 0x0e

プリプロセッサ（2010.07.05）

塩澤秀和 http://vilab.org

C言語の関数（復習）

- <math.h>
  - pow(x, y)　x^y
  - sqrt(x)　$\sqrt{x}$
  - log(x)　log(x)
  - sin, cos, tan　三角関数
- <stdio.h>
  - printf, scanf　書式つき入出力
  - putchar(c)　1文字表示
  - getchar()　1文字入力
- <ctype.h>
  - isdigit(c)　文字が数字か判定
  - isalpha(c)　英文字か判定
  - isupper(c)　大文字か判定
  - tolower(c)　大文字→小文字
- <stdlib.h>
  - rand()　乱数の発生
  - srand(seed)　乱数の初期設定
  - exit(code)　プログラム終了
- <time.h>
  - time(NULL)　現在時刻の取得
- その他
  - main　メインルーチン
  - 自作の関数　サブルーチン
  - ライブラリ関数一覧（p.191）
- 関数でないもの
  - 制御文　if, while, for, return, ...
  - プリプロセッサ指令　#include, ...

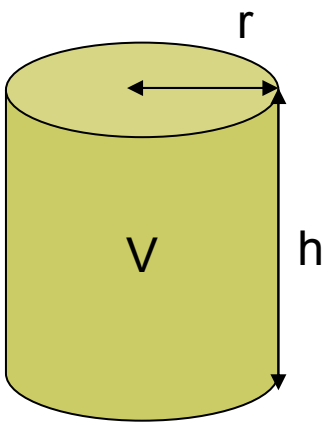
変数の種類（復習）

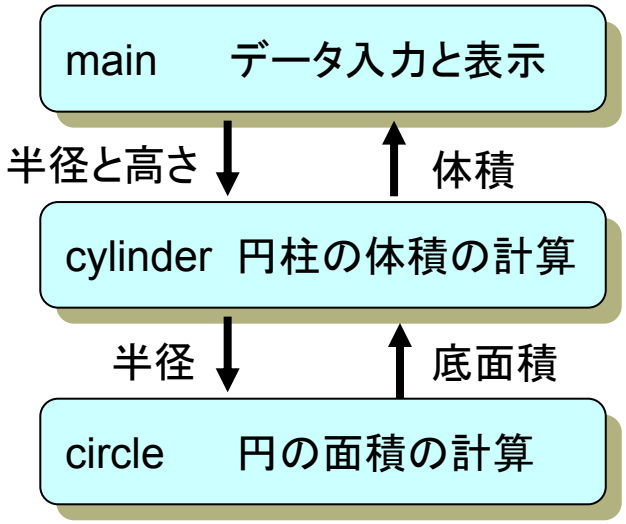
	定義場所	スコープ（通用範囲）	記憶寿命
ローカル変数 （局所変数）	関数の内部 関数の仮引数	関数の内部 ブロックで定義された場合はその内部	自動変数（auto） 関数の実行中のみ （p.87）
			静的変数（static） プログラムの実行中 （p.90）
グローバル変数 （大域変数） ※	全関数の外部 （外部変数）	プログラム全体 † 定義場所より前や別ファイルで使うには、別途、extern による宣言が必要（p.89）	プログラムの実行中 （静的記憶領域）

※ グローバル変数は、むやみに使うとプログラムが分かりにくくなる
† C言語には、このほかにグローバルな static 変数（ファイルスコープ）がある

関数の中で関数を使う（復習）

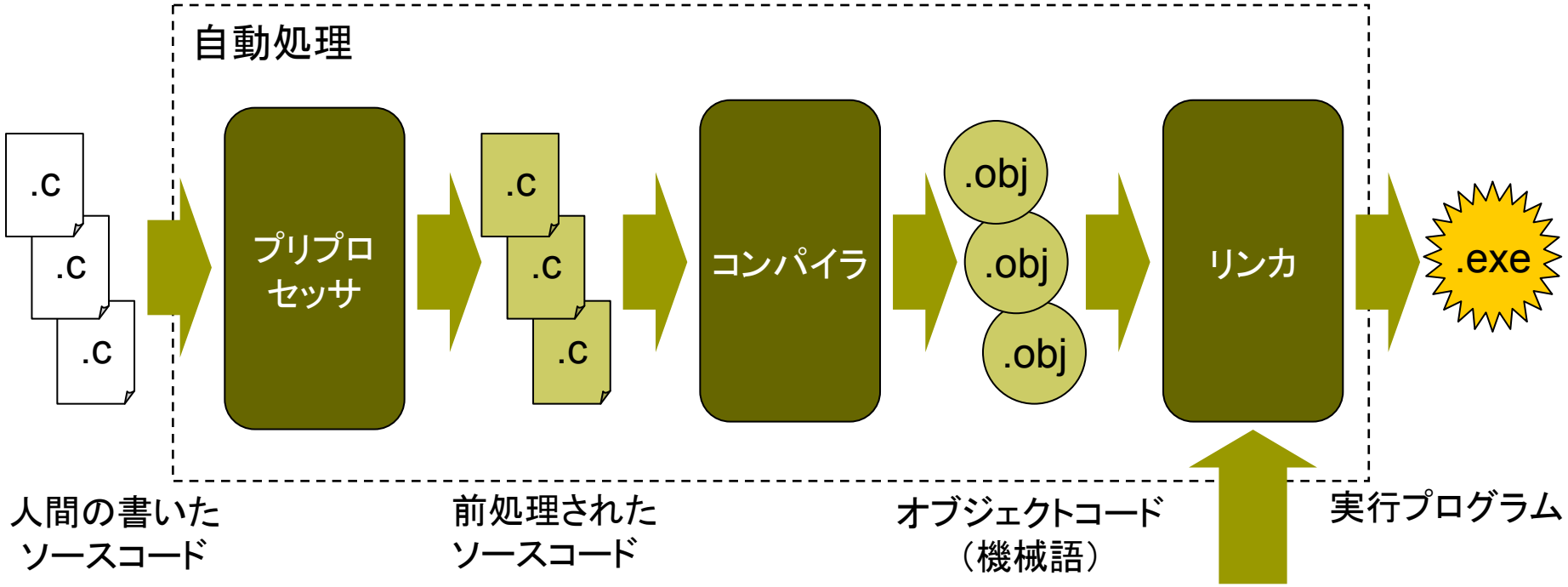
```
#include <stdio.h>

/* 円の面積 */
double circle(double r)
{
  double pi = 3.1415926536;
  return pi * r * r;
}

/* 円柱の体積＝円の面積×高さ */
double cylinder(double r,
   double h)
{
  return circle(r) * h;
}

int main(void)
{
  double r, h, V;

  printf("r = ");
  scanf("%lf", &r);
  printf("h = ");
  scanf("%lf", &h);

  V = cylinder(r, h);
  printf("V = %g¥n", V);
  return 0;
}
```

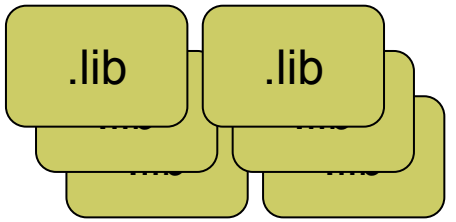




関数の呼び出し関係図

コンパイラの働き（復習）





コンパイラの役割（p.2）
人間がプログラム言語で書いたソースコードを
コンピュータが分かる機械語に翻訳する

マクロ定数

- プリプロセッサとは（p.93）
  - 前処理（コンパイルする前の処理）をするソフトウェア
  - 「#」で始まるプリプロセッサ指令に従って、ソースコードを“加工”する
- #define 指令（p.95）
  - 書式：「#define　マクロ名　文字列」
  - ソースコード内のマクロ名を、すべて文字列で置き換える
    例：#define PI 3.14　← 以後、「PI」と書くと「3.14」と同じ意味になる
  - さらに高度な #define ⇒ 引数付きマクロ（p.97）
- マクロの利点
  - 定数に名前をつけると、プログラムの意味が分かりやすくなる
  - プログラムを修正するとき、1ヶ所の定義を変更すればよい
  - 単なる文字列の置き換えなので、変数のようにメモリを使わない

マクロ定数の例

```
#include <stdio.h>

#define NUM 10
#define PROMPT "整数を10個の入力してください。¥n"
#define REPORT "合計は%dで、平均は%fです。¥n"

int main(void)
{
  int i;
  int in, sum = 0;

  printf(PROMPT);
  for (i = 0; i < NUM; i++) {
     scanf("%d", &in);
     sum += in;
  }
  printf(REPORT, sum, (double)sum / (double)NUM);
  return 0;
}
```

ヘッダファイル

- #include 指令（p.94）
  - 書式：「#include <ファイル名>」または「#include "ファイル名"」
  - 指定したファイルの内容を、その場所に丸ごと取り込む
  - <ファイル名> なら、コンパイラ設定のフォルダからファイルを探す
  - "ファイル名" なら、ソースコードと同じフォルダからファイルを探す

- ヘッダファイル（インクルードファイル）とは
  - #include で読み込むために用意されたファイル（拡張子「.h」）
  - 関数プロトタイプ宣言、マクロ定数の定義などを書いておく
  - ライブラリ関数を使うには、それぞれ指定のヘッダファイルが必要
    - C:¥Program Files¥Microsoft Visual Studio 9.0¥VC¥include¥

- その他のプリプロセッサ指令
  - #if, #ifdef, #endif, #line など ⇒ 教科書 p.98～100 を読んでおく

分割コンパイル

- 大きなソフトウェアの開発
  - プログラムを何個かのファイル(モジュール)に分けて作る

main.c

```
#include <stdio.h>
#include "menseki.h"

int main(void)
{
  double r, s;

  printf("半径: ");
  scanf("%lf", &r);
  s = menseki(r);
  printf("面積: %f¥n", s);
  return 0;
}
```

menseki.h

```
double menseki(double r);
#define PI 3.14159
```

menseki.c

```
#include "menseki.h"

double menseki(double r)
{
  double s;
  s = PI * r * r;
  return s;
}
```

問題の意味を読み取る

- 試験問題や練習問題の目的は？
  - 知識や能力を測る、知識や能力をつけさせる
    ⇒ どんな知識をテストしたいか、“出題のねらい”がある
  - 何が重要か分かっている人は、問題のねらいも分かる
    ⇒ 問題の意味（意図）をちゃんと理解して解答できる

- 例：「～のような関数を作成せよ」
  - 出題のねらい「関数を“仕様通り”正しく定義できるか？」
  - 関数の定義 ＝ ①（仮）引数　②処理内容　③戻り値
  - 問題文からこの3つの指示が読み取れるはず
    - 関数を作る問題なんだから、引数といったら関数定義の仮引数

※注意　ハンドアウト（配布プリント）は、教科書ではありません

演習問題

ドリル配布:「ヒストグラム」について勉強しておくこと

14a. マクロ定数LOWERとUPPERを適当な整数に定義し、キーボードから10個の整数を読み込んで、LOWER以上UPPER以下のものの個数を表示するプログラムを作成しなさい。論理積演算子(&&)を用いるとよい。

14b. 引数をnとすると、n個の「ミ」と1つの「★」を表示する関数を作成しなさい(例: nが3ならば「ミミミ★」)。これを利用して、プログラムで「ミ★ミミ★ミミミ★ミミミミ★」と表示させなさい。

- 関数プロトタイプ宣言 void print_comet(int n);

14c. 以下の2つの関数 f と g をC言語で定義し(引数と戻り値はdouble型)、適当なmainをつけて g の動作を確認しなさい。ただしfabsは使用禁止。

$$f(x) = |x| \quad (\text{絶対値}) \qquad g(x,y) = f(x) - f(y)$$

14d. 標準入力から人数nを読み込んだ後、n人分の年齢(整数)を読み込み、平均年齢を表示するプログラムを作成しなさい。

乱数による簡単なゲーム

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define MAXTRY 5
#define MAXNUM 100

int main(void)
{
  int number, guess;
  int i;

  srand(time(NULL));

  printf("数を当ててください¥n");
  printf("範囲1～%d¥n", MAXNUM);
  number = rand() % MAXNUM + 1;
  for (i = 1; i <= MAXTRY; i++) {
    printf("%d回目 -> ", i);
    scanf("%d", &guess);
    if (guess == number) {
      printf("あたりです！¥n");
      break;
    }
    if (guess > number)
      printf("もっと小さい数です¥n");
    else
      printf("もっと大きい数です¥n");
  }
  if (guess != number) {
    printf("残念...");
    printf("%dでした¥n", number);
  }
  return 0;
}
```