ソフトウェア概論 A/B

-- printf/scanf --

数学科 栗野 俊一

2011/12/02 ソフトウェア概論

伝言

私語は慎むように !!

- 色々なお知らせについて
  - 栗野の Web Page に注意する事

    http://edu-gw2.math.cst.nihon-u.ac.jp/~kurino
- 廊下側の一列は遅刻者専用です(早く来た人は座らない)
- 講義開始前にすませておくこと
  - PC の電源を入れる
  - ネットワークに接続しておくこと
  - 今日の資料に目を通しておくこと
- 講義前の注意
  - 講義前は、栗野は準備で忙しいので TA を捕まえてください
- やる気のある方へ
  - 今日の資料は、すでに上っています
    - どんどん、先に進んでかまいません

前回の復習：多次元の配列と動的メモリ

- 多次元の配列
  - 一次元の配列
    - 同じ型の変数を並べた物が作れる
    - int a1d[10]; → int a1d[0], int a1d[2], .., a1d[9]
    - 個々の「a1d[k]」は int 型 / a1d は int[10] 型
    - ポイント : 同じ型であればなんでもよい
  - 二次元の配列
    - 配列を並べたらどうなるか？
    - int a2d[10][10]; → int a2d[0][10], int a2d[2][10], .., a2d[9][10];
    - 個々の「a2d[k]」は int[10] 型 (配列) / a2d は int[10][10] 型
  - n 次元の配列
    - n-1 次元の配列を並べたもの (配列の配列..)
- 動的なメモリ利用
  - 変数宣言：プログラム作成時にサイズが固定 (静的メモリの確保)
    - 配列のサイズも固定にする必要がある
  - プログラムの実行時にサイズを決めたい (動的メモリの確保)
    - malloc/calloc : 指定したサイズのメモリを動的に確保する (値はポインター)
    - free : 動的に確保したメモリを開放する

お知らせ

- 本日の予定
  - printf/scanf
  - C 言語でオセロ
- 本日の目標
  - 演習
    - 課題の提出

前回 (2011/11/25) の課題

□ 前回 (2011/11/25) の課題

- ○ 課題 1: [sample-007.c 参照]
  - ▷ ファイル名 : 20111125-1-QQQQ.c (QQQQ は学生番号)
  - ▷ 内容 : 3 次行列の積を計算するプログラム
  - ▷ ファイル形式 : テキストファイル(C 言語プログラムファイル)

今週 (2011/12/02) の課題

- 今週 (2011/12/02) の課題
  - 課題 1: [sample-006.c 参照]
    - ファイル名 : 20111202-1-QQQQ.c (QQQQ は学生番号)
    - 内容 : myprintf で Point2D 型を出力する %D が扱えるように拡張する
    - ファイル形式 : テキストファイル(C 言語プログラムファイル)

データの混在した出力

- □ データの混在した出力
  - ○ 出力の場合に文字列の中に数値を含める事が多い (sample-001.c)
    - ▹ 一行の出力に複数の関数の呼び出し : 煩雑なかんじがする
  - ○ パターンが固定ならば、ある程度は簡便化できる (sample-002.c)
    - ▹ 単に関数にしても、お余り得した気分になっていない
- □ 文字列の中に数値を埋め込む事を考える
  - ○ 文字列の中に数値を表現する特殊な文字列を含め、数値に置き換える
    - ▹ '%' を特別な意味に利用 (sample-003.c)
  - ○ 更に、複数の整数値が扱えるようにする (sample-004.c)
    - ▹ 引数が可変長になる (関数宣言の引数の "..." に注意)
  - ○ 整数値だが、8 進や 16 進で出したい場合も考える (sample-005.c)
    - ▹ '%' の後ろの一文字で判断 (d : 10 進 / o : 8 進 / x : 16 進)
    - ▹ '%' 自身を出力するために "%%" で、'%' を出力する
  - ○ 更に、色々な型の値の出力を考える (sample-006.c)
    - ▹ c : 文字 / s : 文字列 / f : 浮動小数点数
- □ printf : ライブラリ関数
  - ○ 書式付きの出力関数 (sample-07.c)

printf の書式

- printf 関数
  - 色々な数値を出力形式(format)を指定して出力する関数
    - printf (char *format, ...); // 引数の個数は可変長
    - ex. printf ("答は %d です\n", 1 + 2 + 3); →「答は 6 です[改行]」
- 書式指定の一般形式
  - <書式指定> ::= (<文字> | "%%" | <書式>)*
    - 書式指定は、文字(それ自身)か、"%%" ('%' 一文字) か、書式(数値表示)
  - <書式> ::= '%' [<精度>] <型指定>
    - 書式は '%' で始まり、省略可能な精度記述の後に型指定がある
  - <精度> ::= ['-'] <整数> ['.' <整数>]
    - 精度は '-' (省略可能) を先行した整数で、'.' 以下が追加できる
  - <型指定> := 'd' | 'c' | 's' | 'f' | ..
    - 型指定は、一文字で、そのデータの型を表現する

scanf : 書式付の入力

- scanf 関数
  - 色々な数値を出力形式(format)を指定して入力する
    - scanf (char *format, ...); // 引数の個数は可変長
    - ex. scanf ("%d", &n); // n は整数型 / 引数はポインターを指定
- scanf の書式
  - 基本は、printf と同じ

オセロのプログラム

- オセロのプログラムの使い方
  - othello.zip を展開する
    - othello という名前のフォルダができる
  - コマンドプロンプトで cd othello でそこに移動
  - cc othello.c とする
    - othello.exe ができる
  - othello とすると、オセロができる
    - 人間が交互にうつためのプログラム