

ソフトウェア概論 A/B

-- 代入と制御構造 --

数学科 栗野 俊一 / 渡辺 俊一

2013/07/05 ソフトウェア概

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)

□ 講義開始前に済ませておく事

- PC の電源を入れる
- ネットワークに接続しておく事
- 今日の資料に目を通しておく事

□ 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

今後の予定

□ 今後の予定 (後ろから)

○ 2013/07/26 前期講義最終日

▶ 試験 / Note-PC 必須 / PC のトラブル対応はしない

○ 2013/07/19 前期講義最終日前

▶ 前期のまとめ / 模擬試験 / Note-PC 必須 / 環境を整える

○ 2013/07/12

▶ 前期の内容の終わり / データ構造 (二周目の終了)

○ 2013/07/05

▶ 本日 / 代入と制御構造

前回 (2013/06/28) の復習

□ 前回 (2013/06/28) の復習

○ 関数：幾つか(0 以上)の値を与えると値(ない場合も含む)を返す

▶ 与える値：仮引数変数に値が入っている (変数には値に対応する型宣言が必要)

▶ 返す値：`return` 命令で値を返す(関数の返す値に対応する型宣言が必要)

○ 三つの制御構造

▶ 順接：命令を組合せる (命令を並べる / 関数合成)

▶ 条件分岐：`if (条件) { 成立時 } else { 不成立時 }`

▶ !! 「条件」は整数値で、0 だと偽、0 以外だと真になる

▶ 繰返し：同じ命令を繰り返す (再帰呼出しで実現)

○ 浮動小数点数の性質

▶ 小数(実数)を表すための型：小数点を利用できる

▶ 計算に誤差を含む場合(殆どの場合)があるので 0 と比較してはいけない

○ 二分探査による方程式の数値解法

▶ 5 以上の代数方程式は解析的には解けない / 数値的には解ける

二分法

□ 二分法 (<数学> 計算機の中でも実生活[辞書]でも使える)

○ 基本的なアイデア

- ▶ 解(候補)集合を二つに分け、現時点の情報から解がどちらにあるか判定し選択する
- ▶ 解(候補)集合が十分に小さく(要素が一つになるなど..)なれば、それを解とする

○ 数値計算での二分法の適用

- ▶ 解の入っている区間 $[i, a]$ を二つに分ける $[i, m]$, $[m, a]$ ($m = (a+i)/2$)
- ▶ 「解の性質から、どちらに解があるかが解る」のであれば、二分法が適用できる
- ▶ 区間の幅 $(a-i)$ が十分に小さく(先に定めた ε 以下に)なれば、解が(誤差 ε 以下で)得られる

二分探査による方程式の数値解法

□ 二分法による方程式($f(x)=0$)の解法

○ 目的 : 連続関数 $f(x)$ の 0 点 ($f(x')=0$ となる x') を一つ求める

○ 条件

▷ $f(x)$ は連続 (<大前提> 中間値の定理を使うので)

▷ $f(i) < 0 < f(a)$ となる区間 $[i,a]$ が与えられている

◇ 中間値の定理からその中に根が一つある事が保証される

○ 手段 : 二分法を使う

▷ $[i,a]$ を $[i,m]$, $[m,a]$ に分割

◇ $f(m) \geq 0$ なら $[i,m]$ に解がある

◇ そうでないなら $[m,a]$ に解がある

お知らせ

- 本日の予定
 - 代入
 - **while** 文 : 代入を利用する制御構造
- 本日の目標
 - 演習
 - ▶ 課題の提出

前回 (2013/06/28) の課題

□ 前回 (2013/06/28) の課題

○ 課題 1:

- ▶ ファイル名 : 20130705-1-XXXX.c (XXXX は学生番号)
- ▶ 内容 : キーボードから入力された一行の文字列を全て大文字にして出力する
- ▶ 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)

本日の課題 (2013/07/05)

□ 本日 (2013/07/05) の課題

○ 課題 1: (前回の課題の残り)

- ▶ ファイル名 : 20130628-2-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 入力された数の 3 乗根(の近似値)を求める
- ▶ 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)

○ 課題 1:

- ▶ ファイル名 : 20130705-2-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 入力された三つの整数を小さい順に出す
- ▶ 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)

代入

□ 代入とは

○ 概念：「変数」に「値」を「割り当て」る「操作」

- ▶ 代入「後」は、その変数の値は、代入さ(割当ら)れた値に「変化」する
- ▶ 代入「前」の値は、「失われ」る
- ▶ 代入の「前」と「後」という「時間」の概念の把握が必要となる

○ 表現：代入の構文

- ▶ 「変数名」=「式」 [例] `a=1+2;` (変数 `a` に `3 (= 1+2)` を代入)
- ▶ 「=」は、「代入」を表現する(等号[等しい]ではない!! / 等号は「==」)

□ 局所変数宣言

○ 概念：局所変数を宣言する

- ▶ 関数(ブロック)内のみ(局所的)で有効(利用可能)な変数を宣言する
- ▶ 「仮引数変数(実は局所変数の一種)」以外にも、変数が増やせる

○ 表現：局所変数の宣言

- ▶ 「変数の型名」「変数名」 [例] `int a;` (整数型の変数 `a` を宣言)
- ▶ cf. 仮引数変数は、実引数の値で、「代入済」の変数
- ▶ 未代入の変数の値は「未定(プログラムミスの代表例 !!)」
- ▶ 変数は宣言と同時に「初期化(最初の代入)」できる(すべき) [例] `int a=1;`

while 文

□ while 文

○ 概念：繰返しのため構文

▶ 同じ命令を繰り返す事ができる (cf. 再帰呼出し)

○ 表現：while 文

▶ while (「条件」) {「繰り返す命令」}

▶ 「条件」の部分は、if と同じ

▶ 「繰り返す命令」の中には、「代入」が必須 (でないと「条件」が変化しない)

□ while 文 vs 再帰

○ while 文は常に再帰に変換できる (実は原理的に逆も可能だが自明ではない)

▶ `func() { while (条件) { 文 } }` → `func() { if (条件) { 文; func() } else { } }`

○ その意味で、再帰の方が表現力がある(優秀)といえる

▶ 逆に(工学のトレードオフの典型例)、while 文の方が「効率」がよい

printf

□ printf : 超高機能出力関数

○ print with format (書式付き出力)

▶ 単なる文字列出力関数ではなかった (cf. `s_print_string` : 単機能)

○ 「書式('%' + 書式指示)」を指定する事により何(基本型+文字列)が出力できる

▶ `printf ("%d", 123);` / `printf ("%f", 1.23);` / `printf ("%c", 'a');` / `printf ("%s", "abc");`

○ 文字列の中に出力を埋め込む事ができる

▶ `int a=123; printf ("int a=%d\n", a);`

○ 複数のデータを一度に出力する事ができる

▶ `int a=123; double b=1.23; printf ("int a=%d, double b=%f\n", a, b);`

○ print の動作原理

▶ 後期にちゃんと話すので、今回は我慢 !!

scanf

□ scanf : 超高機能入力関数

○ scan with format (書式付き入力)

▶ 色々な型のデータを読み込む事ができる (cf. s_input_int : 単機能)

○ 「書式('%' + 書式指示)」を指定する事により何(基本型+文字列)が入力できる

▶ `int a; scanf ("%d", &a);`

▶ !! a の前の「&」は「お呪い」(後期にちゃんと話す)

○ 書式や機能などについても printf と同様に考えてよい

▶ 文字列の中から値を取り出す事もできるのだが.. (結構難しいのでさけるのが無難 ..)