

ソフトウェア概論 A/B

-- 繰返し構文 (while/for) --

数学科 栗野 俊一

2011/11/04 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

□ 前回の復習

○ 手続的観点:変数の値の書き換えによる計算

- ▶ 「計算する事」=「変数の値を書き換える事」
- ▶ 代入文の組み合わせで計算を行う
- ▶ 特定な変数(群)の値を「答」に近付けて行く(他の変数は、答の為の材料)

○ 代入文(命令)の組み合わせ方

- ▶ 順接, 条件分岐, 繰返し(再帰) : ここまでは以前に学んでいる
- ▶ 繰返し(while 文) :

「while(「条件」) {「処理」}」

「条件」が成立している限り「処理」を繰り返す

「条件」には、「処理」によって、「値が変化する変数」を含む

「代入文」なしでは利用できない

○ while 文の眷属

- ▶ 「for (「初期化」;「繰返し条件」;「更新」) {「処理」; }」

「「初期化」;while (「繰返し条件」) {「処理」;「更新」; }」

- ▶ 「do {「処理」} while (「条件」)」

「「処理」; while (「条件」) {「処理」}」

お知らせ

□ 本日の予定

- 繰り返し構文(loop)を利用した様々なプログラム
- 正規表現
 - ▶ 有限状態オートマトンと文字列の判定

□ 本日の目標

- 演習
 - ▶ 課題の提出

前回 (2011/10/21) の課題

□ 前回 (2011/10/21) の課題

○ 課題 1:

- ▶ ファイル名 : 20111021-1-XXXX.c (XXXX は学生番号)
- ▶ 内容 :
 - ▶ 10 個の整数をキーボードから入力し、その総和を求め出力する
 - ▶ for 文を利用する事
- ▶ 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 (2011/11/04) の課題

□ 今週 (2011/11/04) の課題

○ 課題 1:

- ▶ ファイル名 : 20111104-1-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 :
 - ▶ 1 から n 個の奇数の和を求める関数を作れ
 - ▶ 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 2:

- ▶ ファイル名 : 20111104-2-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 :
 - ▶ 入力された文字列が、C 言語の 8 進整数を表すかどうかを判定する関数を作れ
 - ▶ 引数は文字列で、結果は関数の返り値とする
 - ▶ 返り値の意味 : 1 -- 8 進整数 / 0 -- それ以外
 - ▶ 8 進整数の正規表現 : `-(0[0-7]*)`
 - ▶ 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)

while/for 文 (復習)

- インクリメント/デクリメント演算子 (++ / --)
 - 「変数 ++」: ほぼ、「変数 = 変数 + 1」と同じ意味 (-- も同様)
- while の基本構造 (sample-001.c)
 - while (「繰返し条件」) { 「繰返し処理」 }
 - ▷ 「繰返し条件」が成立する限り「繰返し処理」を何度でも実行する
 - ▷ 「繰返し条件」は、基本的に「変数の値」で判断
 - ▷ 「繰返し処理」の中で「代入文」がないと、いつまでも回り続ける
- while と再帰 (sample-002.c)
 - while 文は、再帰でエミュレートできる
 - ▷ 「 F() { while (C) { P } } 」
 - ▷ 「 F() { if (C) { P; F() } } 」
- for 文の基本構造 (sample-003.c, sample-004.c)
 - 「for (「初期化」; 「繰返し条件」; 「更新」) { 「処理」; }」
 - ▷ 「 「初期化」; while (「繰返し条件」) { 「処理」; 「更新」; } 」
 - 「制御変数」を利用する while 文は for 文の方が「綺麗に」かける

繰返しの中での特殊命令

- 繰返しの中での特殊命令
 - 繰返しの中で利用する事により、特別な意味をもつ
- **break** 文 (**sample-005.c**)
 - この命令によって、無条件に繰返しを中断する
- **continue** 文 (**sample-006.c**)
 - この命令によって、無条件に繰返しの先頭に行く
 - ▶ **while** だと単に「条件判定」に行くが **for** の場合は「更新」に行く
- 繰返しの入れ子 (**sample-007.c**)
 - 繰返し部分にも繰返しが入られる..
- **return** 文 (**sample-008.c, sample-009.c**)
 - これは、繰返しと無関係に関数を終了させる
 - ▶ 入れ子の深い所から一挙に飛び出せる

色々な繰返し

- n 回数繰り返す (sample-010.c)
 - 「あり」を「十回」出力
 - ▷ 繰返し回数は制御変数で決る
 - ▷ 本体は、制御変数と無関係
- 制御変数を本体で利用する (sample-011.c)
 - 1 から n までを出力する
 - 階和の計算
- 条件が成立するまで繰り返す (sample-012.c)
 - 繰返し回数が不定：本質的な繰返し

状態機械モデル

□ 状態機械モデル

- 「ある物」が「状態」をもっている
- 何か「入る」と、「状態」が変わる
- 「状態」によって、「出る」物も変わる
 - ▷ 「出る」物が、「入る」物だけでなく、「状態」によっても異なる
 - ▷ cf. 「関数」: 「状態」がない
 - ▷ ex. 自動販売機 (sample-013.c) / 人の気分 (sample-014.c)

□ 状態機械モデルの実現

- 状態を「変数の値」で表現する
- 状態を変える事は、変数の値を変える事

文字列のパターンと正規表現

□ 正規表現

- 文字列の集合を表現する方法の一つ

- ▷ cf. 詳しくは Web 参照

- <http://ja.wikipedia.org/wiki/%E6%AD%A3%E8%A6%8F%E8%A1%A8%E7%8F%BE>

- ▷ ex.

- シンボル : `[_a-zA-Z][_a-zA-Z0-9]*`

- 整数 : `-?[0|[1-9][0-9]*`

□ 正規表現で表されたパターンかどうかの判定

- 状態機械モデルで記述できる

- ▷ cf. 状態遷移

- ▷ ex. シンボル (sample-015.c)