ソフトウェア概論 A/B

-- 繰返し構文 (while/for) --

数学科 栗野 俊一

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
  - 代入：変数の値を変更する操作
    - 代入によって変数の値が変更される (前の値は失われる)
    - 代入文 (「変数名 = 式」) :「変数」の値は「式」の値に変化する
  - 「計算」とは？:「計算」に対する二つの見方
    - 関数的観点 : 計算は、「式の評価」をする事
    - 手続的観点 : 計算は、「値を変化」させる事
  - 局所変数宣言：引数以外の変数を利用する場合に必要
    - (局所:引数も..)変数はブロック内のみ有効
  - 変数を利用したプログラムパターン
    - プログラムは入力・処理・出力の三つの部分からなる
    - 入力 : 変数を初期化する
    - 処理 : (計算を利用しつつ..) 変数の値を変更する
    - 出力 : 変数の値を「結果」として出力する

お知らせ

- 本日の予定
  - 繰返し構文
    - while / for
  - 代入による計算
    - プログラム設計パターン：入力・処理・出力
    - ステートマシンモデル (状態計算モデル)
- 本日の目標
  - 演習
    - 課題の提出

前回 (2011/10/07) の課題

- 前回 (2011/10/07) の課題
  - 課題 1:
    - ファイル名 : 20111007-1-QQQQ.c (QQQQ は学生番号)
    - 内容 :

      二つの整数型の変数 m,n を宣言し、それにキーボードから値を入力する

      五つの整数型の変数 wa, sa, seki, sho, amari を宣言し、それに m と n の

      和、差、積、商、余りを計算した結果を代入する。

      変数 wa, sa, seki, sho, amari の結果を出力する
    - 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)

今週 (2011/10/21) の課題

□ 今週 (2011/10/21) の課題

- ○ 課題 1:
  - ▹ ファイル名 : 20111021-1-QQQQ.c (QQQQ 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 (cf. sample-007.c/sample-008.c)
  - ▹ 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)

繰返し

- 代入文による 1 ～ 5 までの総和の計算 (sample-001.c)
  - sum5 に 15 (1 + 2 + 3 + 4 + 5) を入るまで計算
    - sum5 を計算するには sum4 が必要で.. / sum4 を計算するには sum3 が..
    - sum0 は 0 だから..
- 同一変数の使い回し (sample-002.c)
  - sum[N-1] は sum[N] ができたら不要になる..
    - I-P-O パターン
  - 複数の代入文が必要
- パラメータ (i) の導入 (sample-003.c)
  - 代入文の形が同じになる
    - 「順接(表現)」の繰返しで、「(実行の)繰返し」が可能になる
- 判断 (if 文) の導入 (sample-004.c)
  - 「(実行の)繰返し」部分の「長さ」を「判断」で制御
    - n が 1 から 10 の場合は上手く行くが.. 11 だとだめ
    - 「判断」と「順接」では「有限」しか扱えない
- 再帰(蓄積型)で 1 ～ n の総和を計算 (sample-005.c)
  - 「無限」が扱えるようになる
    - 繰返しの部分が関数の本体 / 「再帰呼び出し」が、「繰り返す」事の意味

while 文

- while 構文：再帰を利用しない「繰返し」
  - while を利用して 1 ～ n の総和を計算 (sample-006.c)
- while の基本構造
  - 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
    - sample-006.c では 条件が「 i < n 」で、処理に「 i = i + 1 」があるので止る
- while と再帰
  - while 文は、再帰でエミュレートできる sample-007.c/sample-008.c)
    - 「 F() { while (C) { P } } 」
    - 「 F() { if (C) { P; F() } }」

for 文

- while 文の構造
  - 制御変数の初期化; while 文 (制御変数の更新を含む)
    - 制御変数を導入してで繰返し制御する事が多い
    - もちろん、そうでない汎用な場合もある
- for 文
  - パラメータを明示的に扱う (sample-009.c)
- for と while
  - for 文は while 文でエミュレートできる
    - 「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」;}」
    - !! 後で述べる break, continue があると同じにならない
  - すでに while 文が再帰で表現できるから..
    - while/for は「変数の変更」と関係が深い

do ～ while 構文

- □ do ～ while 構文
  - ○「do { 処理 } while (条件)」
    - ▷ 処理をしてから、条件判定を行う
    - ▷ 必ず一度はしてしまう (のでちょっと危険)
- □ do ～ while と while
  - ○ do ～ while 文は while 文でエミュレートできる
  - ○「do { 処理 } while (条件)」
  - ○「 処理 ; while (条件) { 処理 }」

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
  - この命令によって、無条件に繰返しを中断する
- continue 文
  - この命令によって、無条件に繰返しの先頭に行く
- return 文
  - これは、繰返しと無関係に関数を終了させる

色々な繰返し

- n 回数繰り返す
  - 「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
  - 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
  - 繰り返し回数が不定：本質的な繰返し