ソフトウェア概論 A/B

-- 引数付き関数 --

数学科 栗野 俊一

2011/05/13 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

□ 前回の内容

○ プログラムとは

- ▹ プログラムは「命令」を並べること
- ▹ cf. 命令 : printf():文字列の表示, s_midi_play ():音を鳴らす, etc..

○ 関数とは

- ▹ プログラムの列に名前をつけたもの
- ▹ (引数なし) 関数の定義 : void 関数名() { 命令列.. }
- ▹ (引数なし) 関数の利用 : 関数名(); で、命令列を呼び出せる

○ 関数の利点

- ▹ 名前が付けられるので判りやすい
- ▹ 一度定義すれば何度でも利用できるので、プログラムが単純になる

○ 音楽の演奏の仕方

- ▹ #include "s_midi.h" が必要
- ▹ s_midi_play (音の名前); で音を鳴らす
- ▹ s_midi_length (音の長さ); で音の長さを変更できる

お知らせ

- 本日の予定
  - 引数付き関数を作ってみよう
  - PC で Turtle Graphics (亀プログラム) をしてみよう
  - 条件判定をしてみよう
  - 再帰呼び出しをしてみよう
- 本日の目標
  - プログラムの基本ブロックである関数を学ぶ
  - 演習
    - 引数付き関数の使い方と作り方
    - 亀プログラム
    - 条件判定をするプログラム
    - 再帰呼び出しをするプログラム
    - 課題の提出

先週(2011/05/06)の課題

- 先週 (2011/05/06) の課題
  - 課題 1:
    - ファイル名 : 20110506-1-QQQQ.c (QQQQ は学生番号)
    - 内容 :「Hello, 自分の名前」を1000回以上出力する C 言語のプログラム
    - 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)
    - 前回 1000 回のプログラムを出した人は同じものをもう一度提出すれば良い
  - 課題 2:
    - ファイル名 : 20110506-2-QQQQ.c (QQQQ は学生番号)
    - 内容 : 童謡を演奏するプログラムを作成しなさい
    - 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 3:
    - ファイル名 : 20110506-3-QQQQ.c (QQQQ は学生番号)
    - 内容 : 童謡の歌詞を出力する
    - 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)
    - 可能な限り引数付きの関数で..

今週 (2011/05/13) の課題

□ 今週 (2011/05/13) の課題

- ○ 課題 1:
  - ▹ ファイル名 : 20110513-1-QQQQ.c (QQQQ は学生番号)
  - ▹ 内容 : 漢字の「回」という文字(にみえる..) 絵を Turtle Graphics で書なさい
  - ▹ 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 : 20110513-2-QQQQ.c (QQQQ は学生番号)
  - ▹ 内容 : 引数付き関数で、if 文で条件判断をするプログラムを作成しなさい
  - ▹ 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 3:
  - ▹ ファイル名 : 20110513-3-QQQQ.c (QQQQ は学生番号)
  - ▹ 内容 : 底辺の長さが指定した文字列の二倍の長さ - 1 の横向のピラミッドを作成するプログラム
  - ▹ 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)
  - ▹ 再帰呼び出しを利用する

ファイルの入手とインストール

- ファイルのダウンロード
  - 次の本日 (2011/05/13) のページからファイルをダウンロードする

    http://edu-gw2.math.cst.nihon-u.ac.jp/~kurino/2011/soft/20110513/20110513.html
  - ダウンロードするファイル
    - glinst.bat, glut.h, glut32.dll, glut32.lib, ogltest.c, s_turtle.h
  - いずれも「c:\usr\soft」に保存する事
- ファイルをダンロードしたら次の作業を行う
  - コマンドプロンプトを開く
  - 「cd c:\usr\soft」とする
  - 「glinst」とする
    - 後は、指示に従う
    - この作業は、今回一回だけ行えばよい
  - 「cc ogltest.c」でコンパイル
  - 「ogltest」で実行して、絵が出れよい
    - マウスでクリックすると回転するので試してみよう

Turtle Graphics (亀プログラム)

- おまじない
  - #include "s_turtle.h" を冒頭にいれる
- 「亀」の操り方
  - 「亀」は、最初の状態では
    - 画面の真中にいます
    - 上を向いています
  - 「亀」への命令は次の三つ
    - s_turtle_move(); : 現在の位置に足跡を残し、現在の方向に一歩進みます
    - s_turtle_jump(); : 現在の位置に足跡を残さず、現在の方向に一歩進みます
    - s_turtle_turn(); : 現在の方向を時計回りに 45 度変更します

引数付き関数の作り方 (再録)

- □ 引数とは
  - ○ 関数に与える事により、関数にその引数に対応した挙動をさせるもの
    - ▷ 引数付き関数の定義：引数の値によって挙動が変る
    - ▷ 引数付き関数の利用：指定したい挙動をさせるための値を指定する
    - ▷ cf. 三角関数：引数の角度によって異なる値を返す
- □ 引数付き関数の作り方
  - ○ 似ている二つ関数を一つの引数付き関数にまとめる
    - ▷ 関数の本体の部分を、同じ部分と違う部分に分ける
    - ▷ 違う部分は「変数」に置き換えて、一つの関数定義にまとめる
    - ▷ 関数の仮引数の所に、「変数」を追加する
    - ▷ 呼出す側は、実引数に、「違っていた部分の内容」を指定する

条件分岐

- 引数の内容によって振舞いを「大幅」に変更したい
  - if 文と strcmp 関数を利用して対応できる
    - strcmp 関数：二つの文字列を比較する
  - if (!strcmp (A, B)) { X } else { Y }
    - A と B が同じなら X を、そうでなければ Y を行う
  - 「else if」を使うと更に複数の命令が選べる
    - if (C1) { P1 } else if (C2) { P2 } .. else { Pn }
    - C1 の時 P1、そうでなく C2 の時は P2 .. いずれでもないと Pn
  - おまじない
    - #include <string.h>
  - strncmp (A, B, N);
    - A と B の先頭の N 文字だけを比較する
    - !strncmp ("abc", "abz", 3); : 等しくない
    - !strncmp ("abc", "abz", 2); : 等しい

再帰呼び出し

- □ 文字列を順番にみてゆく
  - ○「"abc" + 1」は「"bc"」と同じ振舞いをする
    - ▹どうゆう仕組かは今回は説明しない
  - ○次々と 1 を加えれば、どんどん短かくなる
    - ▹最も短かくなったかは、空文字(「""」)と比較すれば判定できる
- □ 再帰呼び出し
  - ○普通の関数は、別の関数を呼出す事ができた
    - ▹自分自身も呼出す事ができる !! : 再帰呼び出し
- □ 再帰呼び出しと帰納法
  - ○再帰呼び出しは、帰納法の考え方で問題を解く場合に利用できる
    - ▹再帰呼び出しが上手く行く事は、帰納法で証明できる
  - ○再帰呼び出しをする場合は次の二点が重要 (帰納法と同じ)
    - ▹最も小さい場合 (ここでは、文字列が "" の場合) には終了する
    - ▹そうでない時は、再帰呼び出しするが、その時には文字列を短くする

再帰呼び出しの考え方

- 目標
  - 「全部」をやりたい
    - でも一挙にはできない
- 対策
  - そこで問題を二つに分ける
    - 扱いやすい一部分：これは、そのまま対処してしまう
    - 残り全部：(残り)「全部」なので、再帰呼び出しする
- 注意点
  - 「全部」が空っぽの時を忘れずに処理する