

ソフトウェア概論 A/B

-- 条件判断と繰返し (2) --

数学科 栗野 俊一

2012/05/25 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 (1) 文字と文字列

□ 文字と文字列

- 'a' は文字 (一つの半角文字「a」からなる)
 - ▷ [注意] 「漢」とはできない (C 言語は「全角文字」は扱えない)
- "abc" は文字列 ('a', 'b', 'c' と EOS の四つからなる)
 - ▷ [参考] EOS は「End of String」の事で「\0」で表現される特殊文字
 - ▷ [参考] "" は、(EOS 一つからなる文字列)
 - ▷ [注意] "漢" は大丈夫 ('漢の左', '漢の右', EOS からなる)

□ 文字列の操作

- k 番目の文字の取り出し方 : [k] をつける (k は 0 から始まる事に注意)
 - ▷ cf. "ABC"[0] == 'A', "ABC"[3] == EOS
 - ▷ [注意] C 言語では「x と y が等しい」事を「x == y」で表現する
- 先頭の文字を取り出すには * を付けてもよい
 - ▷ cf. *"ABC" == 'A'
- 文字列の先頭を取り除くには 1 を加えればよい
 - ▷ cf. "ABC" + 1 == "BC"

前回の復習 (2) 文字と文字列

□ 文字列を扱う関数

○ strcmp (文字列1, 文字列2)

- ▶ 二つの引数「文字列1」と「文字列2」を辞書順に比較
- ▶ 文字列1 > 文字列2 の時 : 正の数になる
- ▶ 文字列1 == 文字列2 の時 : 零になる (C 言語の「偽」になる)
- ▶ 文字列1 < 文字列2 の時 : 負の数になる

▶ [参考 1]

```
if ( !strcmp ( a, b ) ) {  
    /* a と b が等しい時 */  
} else {  
    /* a と b が異なる時 */  
}
```

▶ [参考 2]

```
if ( !strcmp ( a, "" ) ) {  
    /* a が空文字の時 */  
} else {  
    /* a が空文字でない時 */  
}
```

- ▶ [参考 3] 「!strcmp(a, "")」の代わりに *a == EOS としてもよい

○ おまじない

- ▶ #include <string.h>

前回の復習 (3) 条件判断 (if 文)

□ 引数の値によって異なる命令を実行したい

○ if 文を使えばよい

□ if 文の利用

▷ [参考]

```
if ( 条件式 ) {  
    条件式が「真」(0 以外) の時に行う命令  
} else {  
    条件式が「偽」(0) の時に行う命令  
}
```

□ 条件式

○ 等号比較「`A == B`」: とりあえず `A` と `B` は「文字」とする

▷ `A` と `B` が同じ「文字」なら、真 (1) になる。そうでなければ偽 (0)

▷ [参考] `A != B` (等しくない), `A > B`, `A < B`, `A >= B`, `A <= B` も使える

○ 比較を行う関数を利用する

▷ `strcmp` は文字列を比較する

前回の復習 (4) 再帰呼び出し

□ 再帰呼び出し

- 関数を定義の中で、自分自身を呼び出せる

 - ▷ cf. 帰納法

- 安易に利用すると、「無限後退」に..

 - ▷ cf. 神様を作ったのは？ 神様の神様、では神様の神様は？ ..

□ 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」が空っぽの時に忘れずに処理する

お知らせ

□ 本日の予定

- 再帰呼び出しで、turtle グラフィックス

□ 本日の目標

- 文字列と文字

- ▶ 引数の扱い

- 演習

- ▶ 再帰呼び出しで turtle をするプログラム

- ▶ 課題の提出

前回の課題 (2012/05/18)

□ 前回 (2012/05/18) の課題

○ 前々回(2012/05/11) の課題の課題 2 と 3 をする

- ▶ ファイル名の日付は 20120511 となる事に注意
- ▶ 既に提出済の方はそのままよい

○ 課題 2:

- ▶ ファイル名 : 20120511-2-XXXX.c (XXXX 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 : 20120511-3-XXXX.c (XXXX 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

本日の課題 (2012/05/25)

□ 本日 (2012/05/25) の課題

○ 前回(2012/05/18) の課題をする

- ▶ ファイル名の日付は 20120518 となる事に注意
- ▶ 既に提出済の方はそのままよい

○ 課題 1:

- ▶ ファイル名 : 20120511-1-XXXX.c (XXXX は学生番号)
- ▶ 内容 : 渦巻模様 (10 辺) の絵を 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)
- ▶ 再帰呼び出しを利用する

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

 - ▶ どういう仕組みかは今回は説明しない

- 次々と 1 を加えれば、どんどん短くなる

 - ▶ 最も短くなったかは、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

