

ソフトウェア概論 A/B

-- 型宣言 / Hello World again / input --

数学科 栗野 俊一 / 渡辺 俊一

2013/05/31 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

前回(2013/05/24)の復習

□ 前回(2013/05/24)の内容

○ 作成方法

- ▶ 再帰呼出：関数の定義の中で、自分自身を呼び出す (復習)
- ▶ 「文字」と「文字列」の関係

○ 表現方法

- ▶ 「文字」の表現の仕方
- ▶ 「文字列」と文字の関係 (文字列から文字を取り出す方法)

○ 操作方法

- ▶ なし (新規に追加した操作はない)

お知らせ

□ 本日(2013/05/31)の予定

○ 講義

- ▶ データ型
- ▶ 亀プログラム再
- ▶ 「Hello World」再び
- ▶ 文字の入力

□ 本日(2013/05/31)の目標

○ 講義

- ▶ プログラムの構造
- ▶ ライブラリの理解

○ 演習

- ▶ 文字の入力
- ▶ 課題の提出

前回 (2013/05/24) の課題

□ 前回 (2013/05/24) の課題

○ 課題 1:

- ▶ ファイル名 : 20130517-1-QQQQ.c (QQQQ 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
- ▶ 注意 : これは前回(2013/05/24)の課題にとするが課題名は先々週の日付(2013/05/17)のまま出す

○ 課題 1:

- ▶ ファイル名 : 20130524-1-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 自分の名前のローマ字を putchar を用いて出力する

本日 (2013/05/31) の課題

□ 本日 (2013/05/31) の課題

○ 課題 1:

- ▶ ファイル名 : 20130531-1-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : キーボードから一文字入力し、その文字によって異なる国の挨拶をする
- ▶ 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)

○ 課題 1:

- ▶ ファイル名 : 20130531-2-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : キーボードから一行(改行まで..)文字列を読み込み、それを逆順に出す
- ▶ 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)

文字列と文字(復習)

□ C 言語での「文字」の扱い

- 文字 : 文字をシングルクォート('')ではさんだもの

 - ▷ cf. 'A', 'a', '1'

 - ▷ 「一文字」と対応している「表現」

- 文字の出力 : **putchar** 関数を利用する

 - ▷ cf. putchar ('A');

- 文字列 : 文字列をダブルクォート("")ではさんだもの

 - ▷ cf. "ABC", "123", "" (空文字列)

 - ▷ 「文字の並び(0個以上)」と対応している「表現」

 - ▷ <<注意>> : 全角(日本語)は、一文字で、二文字分になる

- 文字列の出力 : **printf** 関数を利用する

 - ▷ cf. printf ("abc");

文字列の構造

- 文字列は、文字の並び + 文字の終りからなる
 - "ABC" == { 'A', 'B', 'C', '\0' }
 - ▷ 長さ 3 (n) の文字列は、4 (n+1) つの部分からなる
 - ▷ '\0' を EOS (End Of String) と呼ぶ
 - 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] == '\0', "ABC"[9] == ? (未定義)
 - 先頭の文字を取り出すには * を付けてもよい
 - ▷ cf. *"ABC" == 'A'
 - 先頭の文字を取り除いた残りを取り出すには 1 を加えればよい
 - ▷ cf. "ABC" + 1 == "BC"
- 文字列の判定 : strcmp を使うと、「二つの文字列が同じなら偽」になる
 - cf. strcmp ("ABC", "ABC") ==> 偽, strcmp ("ABC", "XYZ") ==> 真
 - 文字列が「空文字列(長さが 0 / 文字を含まない文字列)」は先頭が EOS かどうかで判定できる
 - ▷ (*"" == '\0') ==> 真

文字の計算

□ 文字は順に並んでいる (ASCII Code 表)

- 'A' の次は 'B', 'X' の次は 'Y', '5' の次は '6'

- 前の文字や次の文字が得られるか？

 - ▶ 「'A' + 1」を出力してみる ('B' が表示される..)

□ 大文字から小文字へ

- $'a' + ('B' - 'A') == 'b'$

 - ▶ 大文字 'B' を、小文字 'b' にするなら 'a' - 'A' を加える

文字を引数に持つ関数と型宣言

□ これまでの関数

- 引数がないか、文字列を引数としていた
 - ▶ 「char *」をお呪いとし、関数を呼び出す時に、文字列を指定
 - ▶ 変数には文字列が入っているとして、考える

□ 文字を引数に持つ関数の場合

- 引数宣言に「char」とする必要がある

□ 型宣言

- 「char *」/「char」は実は、「引数の型」を表現していた
 - ▶ 「char *」は「文字列」
 - ▶ 「char」は「文字」
- 変数に、その型と異なる値を入れようとすると「エラー」になる

□ 「型」と「演算」

- 「文字」に「1 を加える」と、「次の文字」
- 「文字列」に「1 を加える」と、「短くなった文字列」
 - ▶ 同じ「1 を加える」という「演算」でも、「意味」が異なる
- 「演算」と「型」は「一組」で考える必要がある

複雑な条件

□ if 文で条件をチェックできる

○ if (条件 P) { P が真の時 } else { P が偽の時 }

□ if 文を重ねると、複雑な条件がチェックできる

○ if (条件 P) { if (条件 Q) { P と Q が共に真の時 } else {} } else {}

○ if (条件 P) { P か Q が成立 } else if (条件 Q) { P か Q が成立 } else {}

▶ if 文の組み合わせだけで、任意の条件は記述できるか..

○ 論理演算子を使うともっと簡単に

▶ P && Q : P と Q の双方が真なら真

▶ P || Q : P と Q の一方が真なら真

亀プログラム

- 複雑な亀プログラムを書いてみる
 - 複雑な図形が、簡単なプログラム描画できる
- 再帰図形
 - 再帰呼び出しを利用すると再帰的な図形が記述できる
 - ▷ ペアノ曲線