# ソフトウェア概論 A/B

## -- 条件分岐、再帰、返り値、整数型 --

数学科 栗野 俊一

2012/06/15 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 (1)

- 関数(再び)
  - 関数とは => 命令列に名前を付けたもの
    - 関数定義時：自分で、命令列に名前を付けて、新しい関数が作れる
    - 関数実行時：名前を指定する(関数呼び出し)だけで、その命令列が実行できる
  - 引数付き関数とは => 命令の一部を仮引数で表現した関数
    - 関数定義時：命令の一部を仮引数(変数)で表現し、そこでは内容を指定しない
    - 関数実行時：仮引数に入る内容は関数呼び出し時に「実引数」で指定する
  - ライブラリ => 他の人が作った関数定義の集まり
    - 中身を知らなくても、機能が解れば、関数を呼び出す事で利用可能
    - cf. printf, s_midi_play, ..

# 前回の復習 (2)

- 文字の C 言語での表現
  - シングルクォーテーション(「’」)で挟まれた一つの「文字」
    - cf. ’a’ という表現は「a」という文字、’0’ は「0」という文字
  - 文字の画面(標準出力)への出力
    - 「putchar ( 文字表現 )」で、「文字」を出力できる
- 文字列の表現
  - ダブルクォーテーション(「"」)で挟まれた「文字」の並び
    - cf. "abc" という表現は ’a’, ’b’, ’c’, ’\0’ の 4 つの「文字」が並んだもの
    - 「printf ( 文字列表現 )」で、「文字列」が出力される(\0 は出ない)
- 文字列での「計算」
  - 「+1」: 文字列に 1 を加える：先頭の文字が取り除かれる
    - "abc" + 1 は "bc" と同じ(ように振る舞う)
  - 「頭に * を付ける」文字列の先頭の「文字」を取り出す
    - *"abc" は ’a’ と同じ(ように振る舞う)

# 前回の復習 (3)

- 関数宣言
  - 関数の宣言の頭部の形式：「void 関数名 ( 仮引数宣言 )」
  - 仮引数の宣言：「型名 仮引数名」
    - cf. void s_print_string ( char *string )
  - 「型宣言」とは
    - その変数に入るデータの種類を指定する
  - 「char *string」の意味
    - 「string」という仮引数(変数)には「char *」型の物(文字列)が入る
  - 引数に入る情報の種類によって、「型の宣言」も変更する必要がある
    - 他の種類のデータを処理するなら対応して変更する ( cf. char )
- 関数の引数の型の一致
  - 仮引数(の変数)の型と、実引数(の値)の型は一致している必要がある

# お知らせ

- 本日の予定
  - 返り値と return
  - 条件判断と再帰(再度)
  - 整数型の扱い
- 本日の目標
  - 演習
    - 課題の提出

# 前回の課題 (2012/06/08)

- 前回 (2012/06/08) の課題
  - 課題 1:
    - ファイル名 : 20110608-1-QQQQ.c (QQQQ は学生番号)
    - 内容 : putchar で自分の名前を表示
    - 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)

# 本日の課題 (2012/06/15)

- 本日 (2012/06/15) の課題
  - 課題 1:
    - ファイル名 : 20110615-1-QQQQ.c (QQQQ は学生番号)
    - 内容 : 先頭の数字の回数だけ文字列を繰り返し、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)
  - 課題 2:
    - ファイル名 : 20110615-2-QQQQ.c (QQQQ は学生番号)
    - 内容 : 余りを計算するプログラム考えよ
    - 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)

# 関数の値と return 命令 (新)

- □関数宣言
  - ○関数の宣言の頭部の形式：「関数の値の型 関数名 ( 仮引数宣言 )」
    - ▹「関数」は(実は..)値を返す事ができる
    - ▹関数が返す値(関数値)を「返り値(かえりち)」と呼ぶ
  - ○関数の値の型：返り値の型
    - ▹関数の値の型の「void 宣言」:「値を返さない」という特別な意味
    - ▹cf 仮引数宣言の時の「void 宣言」:「引数はない」という特別な意味
    - ▹[注意]「void：ボイド」:「空虚な、空っぽの」という意味
- □return 命令
  - ○「return 式;」で、「式」の値を関数の「返り値」として関数を終了する
  - ○[注意] return には二つの役目がある ( というか、二つ同時に働いてしまう.. )
    - ▹関数の終了：return 命令を実行すると、そこで関数の実行は終了する
    - ▹返り値の指定：その場合、return の後の式の値を「返す値」にする
  - ○[蘊蓄]
    - ▹return 命令がないと関数本体の最後(「}」の所)で、終了する
    - ▹void 型の関数でも「return ;」で、「値を指定せず終了」する事ができる

# 条件分岐 (再)

- 条件分岐 ( if ～ else 文 )
  - 構文「if ( 条件式 ) { X } else { Y }」
    - 条件式が真 ( 0 以外 ) なら「X」を行う
    - そうでない(偽すなわち 0)なら「Y」を行う
    - X と Y のどちらか一方しか行わない事がポイント
- 引数の内容によって振舞いを「大幅」に変更したい
  - cf. X と Y は全く異なる命令で構わない !!
  - if 文と 論理式を利用して対応できる
- 論理式
  - 条件判定式
    - 比較演算子 :「==」:等しい,「!=」:等しくない,「>」:大きい、..
    - [注意] 文字列(str)が空文字かどうかは「*str == '\0'」で判定する
    - (「 str == "" 」は上手く行かない : 理由は後日... )
- 複数の条件のチェック ( else if )
  - 「if ～ else」は一つの条件を判定する
  - 複数の条件を判定する場合は、「if ～ else if ～ else」を使う
    - if ( 条件1 ) { 命令1 } else if ( 条件2 ) { 命令 2 } ..
      else if ( 条件n ) { 命令n } else { 命令n+1 }

# 再帰呼び出し (再)

- 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 (再)

- 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

# 整数型

- 整数型
  - C 言語での整数
    - 表現できる範囲は限られている
    - 32bit の場合は −2147483648 から 2147483647
  - 宣言：int で行う
  - 計算：四則が可能 +, -, *, /
    - / は整数割り算なので、小数点以下は切捨てになる
  - 比較：大小比較、等号、不等号が使える
    - a > b : a が b より大きい
    - a >= b : a が b 以上
    - a == b : a と b が等しい ( = でないことに注意 !! )
    - a != b : a と b が等しくない
- 整数型の出力 (当分は..)
  - s_print.h の中の s_print_int を使う ( sample-010.c )
  - s_print_string で文字列が出力できる
  - s_print_newline で、改行