

ソフトウェア概論 A/B

-- 返値のある関数(2) --

数学科 栗野 俊一 / 渡辺 俊一 (TA: 石川 溪 / 山脇 直樹 [M1])

2019/06/28 ソフトウェア概

伝言

私語は慎むように !!

- 出席パスワード : 20190628
- 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>
- VNC Server Address : 10.9.154.227
 - Password : vnc-2018
- 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 の電源を入れておく
 - ネットワークに接続しておく
 - 今日の資料に目を通しておく
- 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

前回(2019/06/21)の復習

- 前回(2019/06/21)の内容
 - プロトタイプ宣言とヘッダーファイル (分割コンパイル)

お知らせ

- 本日(2019/06/28)の予定
 - s_print.h/s_intput.h
 - 返り値のある関数

今週 (2019/06/28) の課題

□ 今週 (2019/06/28) の課題

○ 課題 20190628-01

- ▶ ファイル名 : 20190628-01-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 大文字を小文字に変換する関数
- ▶ 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 20190628-02:

- ▶ ファイル名 : 20190628-02-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 二つの整数値を入力し余りを計算する関数
- ▶ 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)

先週 (2019/06/21) の課題

□ 先週 (2019/06/21) の課題

○ 課題 20190621-01

- ▶ ファイル名 : 20190621-01-YYYY.c (YYYY は学生番号)
- ▶ 内容 : 2019/05/31 の資料の「砂漠の旅行の問題」の問題を解くプログラムを考えなさい
- ▶ 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)

関数の値

□ 関数の返値(かえりち)[戻り値]

- 関数は `return` 命令を利用する事により、値を返す事ができる

- ▶ 値を返す場合は、関数名の前の `void` の代わりに返す値の型を記述

- `return` 命令

- ▶ `return` 命令の後ろに「式」を書くと、その「式の値」が関数の返値になる

- ▶ `return` 命令が実行されると、その関数の実行は終了する(以下の命令は実行されない)

- ▶ `void` 関数(値を返さない関数)でも、式を省略した `return` が書ける

- ▶ 値を返す関数は、関数の最後に必ず `return` 命令を書く習慣を付ける

□ `main` 関数の返値

- 整数の値を「終了ステータス」として返す習慣がある

- ▶ 「0」は、「正常終了」を意味する

- ▶ 「0 以外の値」は、「異常終了(値はエラーコード)」を意味する

- ▶ `exit` 関数を利用して、`main` 以外でも、終了(終了ステータスの指定)ができる

- `main` 関数の返値の利用例

- ▶ shell 変数「\$?»で、参照可能

- ▶ `make` の実行時にエラーが起きると、`make` を中断する

- ▶ 例 : `cc` (C コンパイラ) コマンドは、コンパイルエラーを見付けるとエラーコードを返すので `make` が中断する

s_print.h/s_input.h

□ s_print.h/s_input.h の使い方

○ インストール

▶ c:/usr/c/include 下にダウンロード

□ 利用法

○ ソースコード内で `#include "s_print.h"`, `#include "s_input.h"` とする

○ コンパイル時に `-I~/c/include` を付ける

□ 利用可能な関数

○ s_print.h

▶ `s_print_char`, `s_print_string`, `s_print_int`, `s_print_newline` (, 実は他にも..)

○ s_input.h

▶ `s_input_char`, `s_input_string`, `s_input_int` (, 実は他にも..)

関数の合成と実行順序

□ 関数の評価と副作用

○ 関数：式[計算順序を意味する]の一部を為し、値を持つ (数学的な対象)

▷ $f(x)+g(x) \Rightarrow f(x), g(x)$ の結果(返り値)を求めて、その和を計算する式

▷ 「関数の値を求める」事を「関数評価」と呼ぶ

○ 副作用：値の計算とは独立に、関数内で実行される操作内容

▷ 例：入出力 (副作用の実行順を知るには、出力を試みればよい:デバッグプリント)

▷ 関数評価の順番が、副作用の実行順序になる

▷ $f(x)+g(x) \Rightarrow f(x)$ の副作用がおきてから $g(x)$ の副作用が起きる

□ 関数合成

○ $f(g())$ とすると、 g が実行され、その返り値を利用して f が実行される

▷ $f(g()) \Leftrightarrow g;f$ (副作用の順番に注意)

○ $f(x)=x^2, g(x)=x+1$ の時、

▷ 正: $f(g(3)) \Rightarrow f(3+1)$ [g の実行] $\Rightarrow f(4) \Rightarrow 4^2$ [f の実行] $\Rightarrow 16$

▷ 誤: $f(g(3)) \Rightarrow g(3)^2$ [f の実行] $\Rightarrow (3+1)^2$ [g の実行] $\Rightarrow 4^2 \Rightarrow 16$

□ 関数の引数の評価は後ろから行う

○ $\text{func}(\text{式1}, \text{式2})$ とすると、「式2」、「式1」の順に処理される

○ 関数の引数に副作用(入出力を行う)がある場合は注意