

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

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く
禁じます

伝言

□ 出席パスワード : 20200703

□ 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>

□ 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

今後の予定(後ろから)

ソフトウェア概論 A/B (2020/07/03)

今後の予定(後ろから)

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

今後の予定(後ろから)

□ 今後の予定

○ 2020/07/31 (講義最終日)

▶ 試験 / Note-PC 必須 / PC のトラブル対応はしない / 課題提出最終日

○ 2020/07/24 (講義最終日前)

▶ 前期のまとめ(落穂拾い) / 模擬試験 / Note-PC 必須 / 環境を整える

○ 2020/07/17 (次々週)

▶ 次回 : 再帰 again

○ 2020/07/10 (次週)

▶ 次回 : 色々な条件分岐構文

○ 2020/07/03 (本日)

▶ 返値のある関数 (2)

前回(2020/06/26)のまとめ

ソフトウェア概論 A/B (2020/07/03)

前回(2020/06/26)のまとめ

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

前回(2020/06/26)の復習

- 前回(2020/06/26)の内容
 - 「関数」という考え方
 - ▷ 「プログラムの一部」に「名前」を付ける事
 - ▷ 関数の定義方法
 - ▷ 関数の呼出し方法
 - 文字を引数に持つ関数と型宣言
 - ▷ 文字を引数に持つ関数 : `char *` から `char`
 - ▷ 仮引数変数の「型宣言」

今回(2020/07/03)の予定と課題

ソフトウェア概論 A/B (2020/07/03)

今回(2020/07/03)の予定と課題

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

今回(2020/07/03)の予定

□ 出席パスワード : 20200703

- 出席は CST Portal で取りますが、成績には(残念ながら?)無関係です

 - ▶ 単位を取りたいならば、課題を提出しましょう

□ 本日(2020/07/03)の予定

- 前回(2020/06/26)の残り

 - ▶ 入力-処理-出力 (Input - Process - Output)

 - ▶ 返値(かえりち)のある関数 (1)

- 返値(かえりち)のある関数 (2)

- s_print.h/s_input.h

- 整数型

□ 本日(2020/07/03)の目標

- 課題の提出

先週 (2020/06/26) の課題

□ 先週 (2020/06/26) の課題

○ 課題 20200626-01

- ▶ ファイル名 : 20200626-01-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 英小文字を英大文字に、英大文字を英小文字に入れ替える関数

○ 課題 20200626-02

- ▶ ファイル名 : 20200626-02-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 指定した文字列の中の英小文字を英大文字に、英大文字を英小文字に入れ替えて出力する関数
- ▶ ヒント : 20200626-01 の関数を利用する

○ 課題 20200626-03:

- ▶ ファイル名 : 20200626-03-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 10 未満の長さの文字列の長さを表す一文字の数字を返す関数

□ 提出するファイル形式

- 全てテキストファイル(C 言語プログラムファイル)
- 提出先は CST Portal II

今週 (2020/07/03) の課題

□ 今週 (2020/07/03) の課題

○ 課題 20200703-01

- ▶ ファイル名 : 20200703-01-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 20200703-02:

- ▶ ファイル名 : 20200703-02-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)

返値のある関数 (2)

ソフトウェア概論 A/B (2020/07/03)

返値のある関数 (2)

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

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

ソフトウェア概論 A/B (2020/07/03)

s_print.h/s_input.h

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

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..)

整数型

ソフトウェア概論 A/B (2020/07/03)

整数型

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

整数型

□ 整数型

○ C 言語でも整数(の一部)が扱える

▶ 整数型: $-2^{31} \sim 2^{31} - 1$ (2147483647) まで範囲の整数 (cf. limits.h)

○ 整数の計算

▶ 四則 (和: +, 差: -, 積: *, 商: /, 余り: %) の計算ができる

○ 整数の比較 (if 文で利用する)

▶ 等価 (等しい: ==, 等しくない: !=)

▶ 大小比較 (大なり: >, 小なり: <, 以下: <=, 以上: >=)

○ 整数の入出力

▶ s_print_int()/s_input_int() を利用する

▶ ※ 他にも色々できるのだが、しばらくは触れない

○ 整数型の宣言 (引数/戻り値)

▶ 「int」を利用する (以前は char * [文字列] / char [文字] だった)

関数の合成と実行順序

ソフトウェア概論 A/B (2020/07/03)

関数の合成と実行順序

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます

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

おしまい

ソフトウェア概論 A/B (2020/07/03)

おしまい

講義内容の静止画・動画での撮影、及び SNS 等への転載を固く禁じます