

# ソフトウェア概論 A/B

-- switch 文/文字型/関数値/入力 --

数学科 栗野 俊一

2011/06/10 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

---

- 3D モデリング
  - ninja モデル
- 型概念
  - 整数型
  - 引数の型宣言(再)

# お知らせ

---

## □ 本日の予定

- switch 文
- 文字型 (コーディング)
- 関数値
- 入力

## □ 本日の目標

- 演習
  - ▷ 課題の提出

# 先週 (2011/06/10) の課題

---

## □ 先週 (2011/06/10) の課題

### ○ 課題 1:

- ▷ ファイル名 : 20110610-1-XXXX.c (XXXX 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)

### ○ 課題 2:

- ▷ ファイル名 : 20110610-2-XXXX.c (XXXX は学生番号)
- ▷ 内容 : ニンジャを歩かせて、八角形に二周目回るようにしなさい
- ▷ 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)

# 今週 (2011/06/10) の課題

---

## □ 今週 (2011/06/10) の課題

### ○ 課題 1:

- ▷ ファイル名 : 20110610-1-QQQQ.c (QQQQ は学生番号)
- ▷ 内容 : Fizz Buzz 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)

### ○ 課題 2:

- ▷ ファイル名 : 20110610-2-QQQQ.c (QQQQ は学生番号)
- ▷ 内容 : 文字コード(十六進整数値)を利用して "Hello, World 自分の名前" を表示するプログラム
- ▷ 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)

### ○ 課題 3:

- ▷ ファイル名 : 20110610-3-QQQQ.c (QQQQ は学生番号)
- ▷ 内容 : 二つの整数の余りを値として返す関数を作成しなさい
- ▷ 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)

# ファイルの入手とインストール

---

## □ ファイルのダウンロード

- 次の本日 (2011/06/10) のページからファイルをダウンロードする

<http://edu-gw2.math.cst.nihon-u.ac.jp/~kurino/2011/soft/20110610/20110610.html>

- ダウンロードするファイル

▷ s\_print.h(更新), s\_input.h

- いずれも「c:\usr\soft」に保存する事

# switch 文

---

## □ 条件分岐 (再)

- 条件の成立不成立によって、振舞いを変える仕組 (sample-001.c)

  - ▶ if 文で行う : if ( 「条件」 ) { 「成立時」 } else { 「不成立時」 }

  - ▶ if 文は「成立」と「不成立」の二つ場合にわけ

- 複数の条件の確認 ( sample-00{2,3,4,5}.c )

  - ▶ 「else if」を使う ( sample-002.c )

  - ▶ if ( 条件-1 ) {} else if ( 条件-2 ) {} .. else if ( 条件-n ) {} else {}

## □ switch 文

- 数値の値によって、複数の分岐が行える構文 ( sample-006.c )

  - ▶ switch ( 式 ){ case 数値-1: 文; break; case 数値-2: 文; break; ... default: 文; }

- switch は、排他的条件の時に使える

## □ 整数値の表現 ( sample-007.c )

- 十進数 : 1 から始まる数字列 ( 負の数は '-' を先行する )

- 十六進数 : 0x の後に続く 0 ~ 9, A ~ F ( 小文字も可 ) の列

  - ▶ A ~ F は、十進数の 10 ~ 15 に対応する

- 八進数 : 0 から始まる 0 ~ 7 の数字列



# 文字型

---

## □ 文字型

### ○ C 言語での「文字」の表現

▶ ASCII Code (半角英数記号) : 一つ ( 1 byte ) で 1 文字を表現

▶ Shift JIS Code (全角) : 二つ ( 2 byte ) で 1 文字を表現

### ○ 宣言 : `char` で行う

### ○ 計算 : 小さな範囲の整数と考えてよい ( 整数と互換性がある )

▶ 整数型 (BCC32 の場合..) : 32 bit = 4 byte

▶ 文字型 : 8 bit = 1 byte

### ○ 表現 : 0 ~ 255 の (小さな..) 整数

▶ 「半角英数記号」をシングルクォートで挟むと、それに対応する ASCII Code になる

## □ 文字の出力 (当分は..)

### ○ `s_print.h` の中の `s_print_char` を使う ( `sample-001.c` )

### ○ `putchar` (標準) も使える

# 文字のコーディング

---

## □ コーディングとは ( 再 : 2011/05/20 )

- 「表現」と「意味」の対応付けを行う作業

## □ C 言語での文字のコーディング

- 文字の集合と(小さな)整数値の対応付けを行う

- ▶ 表現 : 小さな整数値 ( 半角は 1 byte, 全角は 2 byte )

- ▶ 意味 : 半角英数記号 ( ASCII/JIS\_X\_0201 ), 全角文字 ( SJIS/JIS\_X\_0208 )

- コーディングの規則

- ▶ 文字と数値の対応表 (コード表) があり、それに対応付けされる

## □ 「文字」操作

- 文字コードは C 言語的には、単なる整数値

- ▶ 整数値として、操作できる : cf. 'A' + 1 -> 'B'

- ▶ 表示の場合は、特別扱い ( putchar : 整数値(文字コード) -> 文字(グリフ/表示) )

# C 言語での文字処理

---

## □ 文字の出力

- putchar/s\_print\_char を利用する

- ▶ 整数値(文字コード)を利用して、「文字」が表示できる (sample-008.c)

- シングルクォートで、半角文字が扱える (sample-009.c)

## □ 文字コードの計算

- C 言語の中では、「文字」は「文字コード(整数)」で表現される

- ▶ 普通に「計算」ができる ( sample-{010,011,012}.c )

## □ 文字の処理

- 「文字の世界」を ASCII Code 表で「整数の世界」へ

- ▶ 'A' -> 65

- 「整数の世界」で計算

- ▶ 65 -> 65 + 32 -> 97

- 「整数の世界」をASCII Code 表で「文字の世界」へ

- ▶ 97 -> 'a'

- 処理の意味

- ▶ 整数の世界 : 32 を加える

- ▶ 文字の世界 : 大文字を小文字に変換する

# 関数値

---

## □ 関数値

- 関数は関数の中で計算した結果を「値」として返す事ができる
  - ▶ 数学の「関数」がそうだった
  - ▶ 関数の値を示すために型宣言が必要
  - ▶ 値を返す場合は「`return 式`」を使う ( `sample-013.c` )
- 関数の値は、「式」の中で利用できる ( `sample-{014,015}.c` )

## □ 乱数関数

- `rand` : 乱数値を関数値として返す関数
  - ▶ 呼出す毎に異なる整数値が返る
- 使用方法 ( `sample-016.c` )
  - ▶ `stdlib.h` を `include`
- 一定の範囲の値を出したければ、余りを利用する ( `sample-017.c` )

# 入力

---

## □ 入力と出力

- `s_print_QQQQ` : 何かを画面に表示させる

  - ▷ プログラムからの出力

- `s_input_QQQQ` : 何かをキーボードから入力する

  - ▷ プログラムへの入力

## □ プログラムの基本構造

- 入力-処理-出力

# 今日のまとめ

---

## □ switch 文

### ○ 条件分岐のもうひとつの形

- ▶ 一つの式の複数の値に対応した分岐をしたい場合に利用する

### ○ 文字型 (コーディング)

- ▶ 「文字」と「数値」の対応関係を利用して、「文字操作」を「計算」できる

### ○ 関数値

- ▶ 計算した結果を関数を呼出した所に返す事ができる

### ○ 入力

- ▶ プログラムにデータを与える事ができる