

# ソフトウェア概論 A/B

-- if 構文/条件式/スカッシュゲーム --

数学科 栗野 俊一 / 渡辺 俊一 ( TA: 栗原 望 / 小嶋 仁子 [M2] )

2018/11/16 ソフトウェア概

# 伝言

---

## 私語は慎むように !!

- 出席パスワード : 20181116
- 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>
- 廊下側の一行は遅刻者専用です(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
  - 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(2018/11/09)の内容

---

## □ switch - case 構文

- 多分岐構文 ( if 文は 2 分岐だが switch は n 分岐 )
  - ▶ (基本は) 整数値を取る式の値に応じて、複数の選択肢の一つを選択する
  - ▶ 表現が直感的で分かり易い ( 実は、「効率も良い」のだが、まあ、それはどうでも .. )
- if -- else if -- .. -- else で(機能的には..) 代用はできる
  - ▶ むしろ else if の方が高機能

## □ for 構文

- 変数の初期化、条件チェック、更新を頭にまとめて表現できる while 構文
  - ▶ for 構文でできる事は (ほぼ) while 構文でも表現できる(逆も真)
  - ▶ 繰返しに関係ある「変数操作」が「一箇所にまとまっている」事が分かり易い
  - ▶ 特に、「N 回繰返す」という頻出パターンの表現には for が適切

## □ 新しい構文 ( switch/for ) の位置付け

- 無くても良いが、利用すると、「判り易く」なる : 今後は多用する
- 自分は使わなくても他人が使うかもしれない ( reading/writing 問題 )

# お知らせ

---

- 本日の予定

- if 文

- スカッシュゲーム (curses)

- 本日の目標

- 演習

- ▶ 課題の提出

# 先週 (2018/11/09) の課題

---

## □ 先週 (2018/11/09) の課題

### ○ 課題 20181109-01:

▶ ファイル名 : 20181116-01-QQQQ.c (QQQQ は学生番号)

▶ 内容 : `curses` を利用して画面に、「回」の文字を書く

○※ ファイル形式は、いずれもテキストファイル(C 言語プログラムファイル)

# 今週 (2018/11/16) の課題

---

## □ 今週 (2018/11/16) の課題

### ○ 課題 20181116-01:

- ▶ ファイル名 : 20181116-01-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 三つの整数の比較(if 構文版)

### ○ 課題 20181116-02:

- ▶ ファイル名 : 20181116-02-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 三つの整数の比較(論理積版)

### ○ 課題 20181116-03:

- ▶ ファイル名 : 20181116-03-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 真偽表

### ○ 課題 20181116-04

- ▶ ファイル名 : 20181116-04-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : ド・モルガン

## □ ※

- ファイル形式は、いずれもテキストファイル(C 言語プログラムファイル)

# C 言語：条件分岐

---

## □ 条件分岐 (条件によって二つの命令の一方を実行)

### ○ if 文の構文

▶ if (「条件」) {「条件成立時の命令」} else {「条件不成立時の命令」}

### ○ if 文の意味

▶ まず「条件」をチェック (0 なら偽、それ以外なら真)

▶ 「条件」が真なら「条件成立時の命令」を実行、そうでなければ「条件不成立時の命令」を実行

## □ if 文の色々なパターン

### ○ else 節の省略：「条件不成立時の命令」がない時

▶ 「if (「条件」) {「条件成立時の命令」}」だけで「else 以下」は省略

### ○ 「命令が一つ」の時：

▶ 「if (「条件」)「条件成立時の命令」」だけで「{」、「}」は不要

▶ 注意：「{」、「}」の省略はお勧めできない !!

# 条件式

---

## □ 条件式

### ○ 機能

▶ 条件分岐 ( if ) や、繰り返し ( while, for ) で条件の判定に利用される式

### ○ 構文

▶ 整数値(真偽値ではない..)を取る式

### ○ 意味

▶ 結果が 0 ならば「偽」、そうでなければ(どんな値でも 0 でない限り)「真」

▶ 論理(真偽)値 : 0(偽) か 1(真) を取る値 (条件式は論理値になる)

## □ 条件式で利用される演算子

### ○ 論理演算子 : 以下の説明では「真(1)」にならない時は「偽(0)」になる

▶ 等号 (==) : 両辺の値が等しい時のみ 真(1)

▶ 等号否定 (!=) : 両辺の値が異なる時のみ 真(1)

▶ 不等号 (>, <, >=, <=) : 両辺の値を比較し、不等式が成立した時のみ 真(1)

▶ 論理積 (&&) : 両辺が共に真(0でない)の時のみ真(1)

▶ 論理和 (||) : 両辺の共に偽(0)の時のみ偽(0)

▶ 否定 (!) : 条件式の前に前置し、それが偽(0)の時のみ真(1)