

ソフトウェア概論 A/B

-- for 構文 / switch 構文 --

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

私語は慎むように !!

- 出席パスワード : 20211015
- 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

前回の内容

ソフトウェア概論 A/B (2021/10/15)

前回の内容

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

前回の内容

□ 前回の内容

○ **while** 構文

▶ 代入を繰り返す構文

○ **while** 構文の応用

▶ 数値計算 (解析的発想: 答の近似値の精度を高めて行く)

本日(2021/10/15)の予定

ソフトウェア概論 A/B (2021/10/15)

本日(2021/10/15)の予定

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

本日(2021/10/15)の予定

□ 本日(2021/10/15)の予定

○ やりのこし

▶ while 構文の応用(数値計算:数値積分)

▶ 標準 I/O とファイル入出力

○ 新規

▶ for 構文

▶ switch 構文

□ 本日の目標

○ 演習

▶ 課題の提出

課題

ソフトウェア概論 A/B (2021/10/15)

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

今週 (2021/10/15) の課題

□ 今週 (2021/10/15) の課題

○ 課題 20211015-01:

▶ ファイル名 : 20211015-01-XXXX.c (XXXX は学生番号)

▶ 内容 : 0 ~ 99 の偶数を出力する (for 構文 版)

○ ※ 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)

先週 (2021/10/08) の課題

□ 先週 (2021/10/08) の課題

○ 課題 20211008-01:

- ▶ ファイル名 : 20211008-01-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 一つ浮動小数点数値をキーボードから入力し、その立方根を出力する

○ 課題 20211008-02:

- ▶ ファイル名 : 20211008-02-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 関数 $\sin(x)$ の区間 $[0, \pi/4]$ の定積分値

○ ※ 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)

for 構文

ソフトウェア概論 A/B (2021/10/15)

for 構文

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

for 構文

□ for 構文とは

○ 繰返しを記述する構文規則 (cf. while)

- ▶ for (<初期化式>; <継続条件式>; <再初期化式>) { <繰返し文> }
- ▶ 初期化式 : 最初に一度だけ、必ず行われる文
- ▶ 継続条件式 : 毎回、繰返し文の実行「前」に評価されこれが偽の場合は終了になる
- ▶ 繰返し文 : for 文によって繰り返される命令
- ▶ 再初期化式 : 繰返し文の実行の後に毎回実行される

□ インクリメント/デクリメント演算子

○ 変数の値を 1 だけ増やしたり減らしたりする演算子

- ▶ 「++」: 変数を 1 だけ増やす => 「i++」は、「i = i + 1」と同じと思って良い
- ▶ 「--」: 変数を 1 だけ減らす => 「i--」は、「i = i - 1」と同じと思って良い

for 構文 と while 構文

□ for 構文と while 構文の関係

○ for と while は相互に書き換えができる

▷ だから、while だけ知っていれば for は要らないのだが、for は便利なので..

○ for から while (while が解っていれば for は解る)

```
for ( <初期化式>; <継続条件式>; <再初期化式> ) { <繰返し文> }
```

→

```
<初期化式>; while ( <継続条件式> ) { <繰返し文> <再初期化式>; }
```

○ while から for (while は for の簡略形)

```
while ( <継続条件式> ) { <繰返し文> }
```

→

```
for ( ; <継続条件式>; ) { <繰返し文> }
```

▷ (while は for の <初期化式>, <再初期化式> が空のもの)

switch 構文

ソフトウェア概論 A/B (2021/10/15)

switch 構文

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

switch 構文

□ switch - case 構文

○ 機能

▶ 多分岐構文 (if 文は 2 分岐だが switch は n 分岐)

○ 構文

▶ `<switch 文> ::= switch (<式>) { <case 並び> }`

▶ `<case 並び> ::= <case> の繰り返し`

▶ `<case> ::= <label> : 文並び`

▶ `<label> := case <定数> | default`

▶ ※1 一つの `<switch 文>` 内には、同じ `<label>` を含める事はできない

▶ ※2 文並び中に `break` 文があり、それが実行されると、`switch` 文は終了する

○ 意味

▶ もし、「式」の値が「定数」の何れかであれば、そのラベル以下の文を実行する

▶ そうでなくて、もし、`default` があれば、そのラベル以下の文を実行する

▶ そのいずれでもなければ、何もしない

break 文

ソフトウェア概論 A/B (2021/10/15)

break 文

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

break 文

□ break 文とは

- 繰返し構文(`while/for`)、あるいは `switch` 構文を(途中で抜けて..)終了する
 - ▶ `break` 文のある場所から外に向けての「最も内側の 繰返し/`switch` を抜け」る
 - ▶ 複数の構文を一度に抜ける事はできない (したいなら `goto` 文か、`return` 文を使う)
- 繰返し構文の中で利用する場合は `if` 構文と一緒に使う(そうしないと意味がない)
 - ▶ `switch` の場合は、単独の場合と `if` 構文と組合せる場合の両方がある

□ ネストした構造から脱出

- 構文の深い場所から抜けるには `return` 文を使うのが簡単
 - ▶ `goto` 文の話はしたくない...(後日..)

おしまい

ソフトウェア概論 A/B (2021/10/15)

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