

# ソフトウェア概論 A/B

## -- データ構造 (3) --

(データ構造の利用と多次元配列)

数学科 栗野 俊一 / 渡辺 俊一

2015/10/30 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>

### □ 講義開始前に済ませておく事

- PC の電源を入れる
- ネットワークに接続しておく事
- 今日の資料に目を通しておく事

### □ 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

### □ 本日の CST Portal の出席パスワード : 20151030

- 出席は成績に影響しませんが、折角の機能なので、使いましょう

# 風邪をひいちゃいました

---

## □ 風邪をひいちゃいました

○ 先週(2015/10/23)の、丁度この時間から喉の調子が悪くなった(風邪をひいた様子)

- ▶ 特に、熱が出たり頭痛はしないのだが..
- ▶ 喉が痛い(咳が出る)、声が擦れてしまう(所謂、喉風邪らしい..)
- ▶ ※睡眠中、咳で目が覚める事があり、ちょっと睡眠不足気味..
- ▶ 医者にはまだ、行っていない..

○ 本来に休講にすべきなのかもしれないのだけど..

- ▶ 体調が万全でない状態で、講義をするって.. ?
- ▶ 学生に風邪をうつしたらどうするの.. ?

○ でも..(言い訳だが..) まあ、普通に生活はできるので、休みにするも抵抗が..

- ▶ 週末は学祭で、少し休めるし...

## □ まあ、とりあえず、講義はやります

○ 声が擦れて聞き取り難いかもしれませんが、許してください

- ▶ ということで、今日は珍しく、マイクを使います

# 前回(2015/10/23)の内容

---

## □ 前回 (2015/10/23) の復習

### ○ コーディングと I/O (入出力)

- ▶ 計算機(プログラム)の「内部表現(=データ)」は全て「数値」
- ▶ 現実(計算機の外)では「情報(様々な形式)」を扱いたい
- ▶ 情報とデータの対応規則がコーディング(規則)
- ▶ コーディング規則に従って、内部と外部の間の変換を行うのが I/O

### ○ 配列と多次元配列

- ▶ 配列 : 同じ型の変数をまとめたもの (cf. 構造体は、異なる型も可)
- ▶ 配列の特徴 : 添字(「式」が使える)で、要素(配列の一部となる変数)が参照できる
- ▶ 配列の要素を配列にできる : 「配列の配列」を、特に「多次元配列」と呼ぶ

# お知らせ

---

- 本日の予定
  - データ構造 (3)
    - ▶ 配列とその応用(2)
- 本日の目標
  - 演習
    - ▶ 課題の提出

# 前回 (2015/10/23) の課題

---

## □ 前回 (2015/10/23) の課題

### ○ 課題 20151023-01:

- ▶ ファイル名 : 20151023-01-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 複素数型の四則
- ▶ 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 20151023-02: (これは、今週[2015/10/30]に回す)

- ▶ ファイル名 : 20151023-02-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 二次元行列の和、差、積
- ▶ 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 20151023-03:

- ▶ ファイル名 : 20151023-03-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 配列を使って 5 個の数を入力し、その 5 倍と 1/2 を出す
- ▶ 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)

# 本日の課題 (2015/10/30)

---

## □ 本日 (2015/10/30) の課題

### ○ 課題 20151023-02: (先週の課題の積み残し)

- ▶ ファイル名 : 20151023-02-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 二次元行列の和、差、積
- ▶ 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 20151030-01:

- ▶ ファイル名 : 20151030-01-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 配列内の浮動小数点数の合計を求める Sum 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 20151030-02:

- ▶ ファイル名 : 20151030-02-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 文字配列に入った文字列の途中に文字を挿入する
- ▶ 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)

# 多次元配列(再)

---

□ 配列：同じ物が、一列に並んだ物（一次元配列）

○ 配列の配列は？（配列が二次元に並んでいる）：二次元配列となる

▶ `int ary[3][3];` int 型変数 `ary[i][j]` が二次元 (3x3) に並んでいる

	0	1	2
0	<code>ary[0][0]</code>	<code>ary[0][1]</code>	<code>ary[0][2]</code>
1	<code>ary[1][0]</code>	<code>ary[1][1]</code>	<code>ary[1][2]</code>
2	<code>ary[2][0]</code>	<code>ary[2][1]</code>	<code>ary[2][2]</code>

○ k 次元の配列の配列は？

▶ k+1 次元の配列になる



# 配列と文字列

---

## □ 文字列とは？

○ 文字の並びだった.. という事は..

▶ char の一次元配列が「文字列」？

## □ 答は Yes でもあり No でもある

○ Yes : 「文字列」は「char の一次元配列」で実現されている

▶ char の一次元配列にできることは「文字列」にもできる

▶ 「文字列」を「char の一次元配列」のように扱って良い

○ No : 逆は真ではない(char の一次元配列は、文字列とは限らない)

▶ char の一次元の配列の要素を文字にし最後に EOS(0)を入れれば文字列と同様に振る舞う

▶ C 言語のコンパイラが、「文字列」を特別扱いしてくれる

# 配列要素の参照

---

## □ 配列要素の参照方法

- 基本は添字参照「`[]`」を利用する

  - ▶ 例 : `int ary[5];` の時 `ary[2]` は 3 番目の要素の参照

- 間接参照「`*`」で先頭の要素が参照できる (文字列の時と同じ)

  - ▶ 例 : `*ary` は `ary[0]` と同じ

- 配列名に整数値を加えると、配列の先頭の要素が「ない」ように振る舞う

  - ▶ 例 : `*(ary+3)` は `ary[3]` と同じ (文字列の時と同じ)

## □ 配列要素参照「`[]`」と「`*`」の関係 (配列一般)

- 「`ARY[INDEX]`」と「`*(ARY+INDEX)`」は何時も同じ振舞いをする

# 配列と関数引数(注意!!)

---

## □ 関数の引数にも配列を渡す事ができる

- 結果：あたかも「配列自身」が渡されているように見える

  - ▶ 関数内で、配列の要素の内容を書き換えると、呼出し元の配列の内容も変る

- 比較：通常の変数(単純変数/構造体)は、「変数そのもの」ではなく「変数の値」が「コピーされて」渡される

  - ▶ 関数内で引数変数の値を変更しても、呼出し元には影響しない

## □ 関数の引数における配列の挙動

- (まだ学んでいない..) ポインターストックの概念が必要

  - ▶ (近い内に話をするが..)今回は、これにはふれない

## □ 関数における配列の扱い

- しばらくは、「そんなもの」と捉える (後に、正しい理解をする)