

ソフトウェア概論 A/B

-- データ構造 (3) --

(様々な配列)

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

2020/11/13 ソフトウェア概

伝言

□ 出席パスワード : 20201113

□ 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

前回の内容

ソフトウェア概論 A/B (2020/11/13)

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：データ構造 (2)

- 配列：複数の同じ型の変数(配列の要素)をまとめたもの
 - ▶ 同じ型の値が並んだもの(ベクトル)を表現するデータ型
- 配列の宣言：配列を構成する要素数を指定して宣言
 - ▶ 例：int a[3] -> 3 変数 (a[0], a[1], a[2]) を宣言
- 配列の要素(配列の一部となる変数)は、[] (添字) で参照可能
 - ▶ 例：a[1] -> 配列 a の 2 つ目 (1 目 a[0] なので..) を表す
 - ▶ 「添字」には、「整数値を持つ式」が入れられる
- 配列による「集合」の表現
 - ▶ サイズ N の配列で、サイズ N の(有限)部分集合が表現できる
- 配列と for 構文の関係
 - ▶ 配列の要素を処理する場合は for 構文が馴染む

本日(2020/11/13)の予定

ソフトウェア概論 A/B (2020/11/13)

本日(2020/11/13)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(2020/11/13)の予定

□ 本日(2020/11/13)の予定

○ データ構造 (3)

▶ 様々な配列

□ 本日の目標

○ 演習

▶ 課題の提出

課題

ソフトウェア概論 A/B (2020/11/13)

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 (2020/11/13) の課題

□ 今週 (2020/11/13) の課題

○ 課題 20201113-01:

- ▶ ファイル名 : 20201113-01-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

○ 課題 20201113-02:

- ▶ ファイル名 : 20201113-02-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

○ 課題 20201113-03:

- ▶ ファイル名 : 20201113-03-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 一行分の文字列を入力して、その中の文字列を全て大文字に変換する

○ ※ 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 (2020/11/06) の課題

□ 先週 (2020/11/06) の課題

○ 課題 20201106-01:

- ▶ ファイル名 : 20201106-01-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 複素数型の四則

○ 課題 20201106-02:

- ▶ ファイル名 : 20201106-02-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 二次元行列の和、差、積

○ 課題 20201106-03:

- ▶ ファイル名 : 20201106-03-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)

配列要素の参照

ソフトウェア概論 A/B (2020/11/13)

配列要素の参照

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

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

□ 「配列名」の意味(詳細は後日)

- 「間接参照「`*`」や添字参照「`[]`」をする」事ができる「値」を表す

- ▶ 「値」なので、関数に渡したり、関数の値にできる

配列と文字列

ソフトウェア概論 A/B (2020/11/13)

配列と文字列

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

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(変更はできないが)

▶ 「構造」(参照の仕方)は同じだが、「属性」(代入できるかどうか)は違う

○No : 逆は真ではない(char の一次元配列が、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

配列と関数引数

ソフトウェア概論 A/B (2020/11/13)

配列と関数引数

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

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みえる
 - ▶ 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
 - ▶ 配列の最初のサイズは省略可能 ([] でよい) / 一次元配列は「*」でも良い
 - 比較：通常の変数(単純変数/構造体)の場合
 - ▶ 「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」の概念が必要
 - ▶ (近い内に話をするが..)今回は、これには触れない
- 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)

おしまい

ソフトウェア概論 A/B (2020/11/13)

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