

ソフトウェア概論 A/B

-- データ構造 (7) --

(動的メモリ管理とキャスト)

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

2021/12/17 ソフトウェア概

伝言

私語は慎むように !!

- 出席パスワード : 20211217
- 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

今後の予定

□ 今後の予定(後ろから)

○ 2022/01/22 ~

- ▶ 2022/01/21 に試験が受けられなかった人の追加試験(日時は本人と相談)

○ 2022/01/21 (講義最終日)

- ▶ 試験を行う
- ▶ 課題提出の最終締切り(この日までに提出した分を成績に反映)

○ 2022/01/14 (講義最終日前)

- ▶ 模擬試験を行う / 落穂拾い

○ 2021/12/31, 2022/01/07

- ▶ 冬期休暇期間中 : この講義はない

○ 2021/12/24

- ▶ 補講日 (ソフトウェア概論はない)

○ 2021/12/17 (本日)

- ▶ データ構造 (7) : 動的メモリ管理とキャスト
- ▶ 試験範囲はこの日まで

前回の内容

ソフトウェア概論 A/B (2021/12/17)

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 (6)

○ 講義

- ▶ ポインター値とメモリモデル
- ▶ アドレス値と型 (sizeof 演算子)
- ▶ 「&」演算子

本日(2021/12/17)の予定

ソフトウェア概論 A/B (2021/12/17)

本日(2021/12/17)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(2021/12/17)の予定

□ 本日(2021/12/17)の予定

○ データ構造 (7)

▶ 動的メモリ管理とキャスト

□ 本日の目標

○ 演習

▶ 課題の提出

課題

ソフトウェア概論 A/B (2021/12/17)

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 (2021/12/17) の課題

□ 今週 (2021/12/17) の課題

○ 課題 今週-01:

- ▶ ファイル名 : 今週-01-XXXX.c (XXXX は学生番号)
- ▶ 内容 : 「三角形の形」をした配列

□ ※

- 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 (2021/12/10) の課題

□ 先週 (2021/12/10) の課題

○ 課題 20211210-01: float と double の精度の差

▶ ファイル名 : 20211210-01-QQQQ.c (QQQQ は学生番号)

▶ 内容 : 自然対数の底の計算

○ 課題 20211210-02:

▶ ファイル名 : 20211210-02-QQQQ.c (QQQQ は学生番号)

▶ 内容 : 動的なメモリの利用

□ ※

○ 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)

一次元配列と多次元配列

ソフトウェア概論 A/B (2021/12/17)

一次元配列と多次元配列

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

一次元配列と多次元配列

□ 多次元配列は配列の配列

- 結局は、メモリ上の連続したセルに対応

- ▶ アドレス値が計算できれば、要素が参照できる

□ 二次元配列のアドレス計算

- `int dary[3][4]; int` 型の $3 \times 4 = 12$ 個の要素からなる

- ▶ のアドレス値 : `dary` の表すアドレス値 + `sizeof(int) * 4 * 1 + sizeof(int) * 2`

- ▶ 結局一次元の場合と同様にアドレス値の計算がされているだけ

型変換とキャスト演算

- 型情報 : 「そのデータをどのようなものとして扱うか」という情報
 - コンパイル時のみ意味がある(実行時は明示的にはわからない)
- 型変換
 - 値の型を変更する事
 - ▶ 例 : int -> double (1 -> 1.0)
 - 「型」は、「それをどう扱うか」を意味するので、型が違えば操作も違う
 - ▶ 例 : 「3/2 -> 1」(int 型) / 「3.0/2.0 -> 1.5」(double 型)
 - ▶ 型から操作を決めているのはコンパイラ / CPU には「型」はない
- キャストティング
 - 型変換を明示的に行う
 - ▶ 「(型名)」を値の前に先行する事により、型を変更できる
 - ▶ 例 : 「(double)1 -> 1.0」 / 「(int)1.5 -> 1」
 - ▶ 「型」だけでなく、「表現」が変化

ポインタ演算とキャスト

- ポインタ値の二つの情報
 - アドレス値 + 型情報
- アドレス値の変更
 - 整数値の加減算ができる
 - ▷ アドレス値は「加える整数値 x sizeof(型)」だけ変化する
- 型情報の変更
 - キャストによって、型情報を変更できる

動的メモリ領域の確保

ソフトウェア概論 A/B (2021/12/17)

動的メモリ領域の確保

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

動的メモリ領域の確保

□ 変数領域の確保

- 基本は変数宣言の時に、その変数に対応する領域が確保される
- 変数は予め宣言しておく必要がある
 - ▶ どの位のデータを記録するかを予め決めておく必要がある
 - ▶ 記憶するデータの量が予想できない場合はどうすべきか？

□ 動的メモリ領域の確保

- `malloc/calloc (alloc 関数)`を利用して、動的にメモリを確保することができる
 - ▶ 確保された領域は使用が終わったら `free` で解放する必要がある
- `alloc` 関数は、確保した領域へのポインター値を返す
 - ▶ 必要に応じて、サイズの指定とキャストが必要
 - ▶ 領域への参照は、必然的にポインター値経由となる(名前がない)

おしまい

ソフトウェア概論 A/B (2021/12/17)

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