

# ソフトウェア概論 A/B

-- 型変換とキャスト --

数学科 栗野 俊一 / 渡辺 俊一

2016/12/02 ソフトウェア概

# 伝言

---

## 私語は慎むように !!

- 出席パスワード : 20161202
- 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>
- 廊下側の一列は遅刻者専用です(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)

○ 2017/01/20 (講議最終日)

▶ 試験を行う

○ 2017/01/13 (講議最終日前)

▶ 模擬試験を行う

○ 2016/12/30 / 2017/01/06

▶ 冬期休暇期間中：この講議はない

○ 2016/12/23

▶ 天皇誕生日(祝日)

○ 2016/12/16

▶ 落穂拾い

○ 2016/12/09 (次週)

▶ データ構造 (8) / 動的データ構造 / ファイル I/O

○ 2016/12/02 (本日)

▶ データ構造 (7) : 動的メモリ管理とキャスト

# 前回(2016/11/25)の内容

---

## □ 前回(2016/11/25)の内容

### ○ ポインタ型

- ▶ 「型」に対して、「型 \*」が、「型」への「ポインタ型」
- ▶ 例「char」に対して、「char \*」は、「char へのポインタ型」
- ▶ 「配列名」は、「ポインタ定数」になっている

### ○ ポインタ型の実体

- ▶ アドレス値(メモリの何処か)と型(サイズ,どんな操作が可能か)情報の対
- ▶ 型情報は、コンパイル時にのみ意識され、実行時はアドレス値しか存在しない
- ▶ 関数の引数や値として受け渡されるのは、「アドレス値」

### ○ ポインタ型の操作 : 「型」の変数 V と「型 \*」のポインタ値 P に対して

- ▶ \*P は「型」になる / &V は「型 \*」になる
- ▶ \* と & は逆操作 :  $\&*P == P$  /  $*\&V == V$

### ○ ポインタとは

- ▶ ポインタ型の値を持つ変数の事
- ▶ ポインタは、変数(メモリ)を経由して間接的にデータを処理する
- ▶ ポインタは、他の変数と関連付けて初めて意味を持つ

# 前回 (2016/11/25) の課題

---

- 前回 (2016/11/25) の課題
  - 課題はなしになった

# 本日の課題 (2016/12/02)

---

## □ 本日 (2016/12/02) の課題

### ○ 課題 20161118-02: (これは前回 2016/11/18 の残り)

- ▶ ファイル名 : 20161118-02-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : アドレスを利用した間接参照

### ○ 課題 20161125-01:

- ▶ ファイル名 : 20161125-01-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)

# 動的メモリ管理

---

- 動的メモリ管理：前回の資料を参照
  - 課題も前回の資料を参照

# 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