

ソフトウェア概論 A/B

-- 浮動小数点数型の利用 --

数学科 栗野 俊一 / 渡辺 俊一

2014/06/20 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 : 20140620

- 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

前回(2014/06/13)の復習

□ 前回(2014/06/13)の内容

○ 作成

- ▶ 値と型と型宣言
- ▶ 整数型/浮動小数点数型

○ 浮動小数点

- ▶ 小数点付数を扱うデータ型
- ▶ 「実数」に対応する

□ コンピュータによる数の表現の問題

○ コンピュータは「有限の物」しか扱えない

- ▶ 整数型：大きな数を扱う事はできない
- ▶ 浮動小数点数型：誤差を伴う

お知らせ

□ 本日(2014/06/20)の予定

○ 講義

- ▶ データとコードの区別
- ▶ 浮動小数点数型の利用法

□ 本日の目標

○ 演習

- ▶ 課題の提出

前回 (2014/06/13) の課題

□ 前回 (2014/06/13) の課題

○ 課題 3:

- ▶ ファイル名 : 20140604-01-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 数当てをするプログラム
- ▶ 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 4:

- ▶ ファイル名 : 20140604-02-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 与えられた整数の素因数を表示するプログラム
- ▶ 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)

○ (注意) 前々回の課題 1, 2 を前回出す

本日 (2014/06/20) の課題

□ 本日 (2014/06/20) の課題 (CST Portal のみ)

○ 課題 1:

- ▶ ファイル名 : 20140613-01-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : キーボードから入力された二つの整数型の値の四則と余りの結果を表示する
- ▶ 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)

○ 課題 2:

- ▶ ファイル名 : 20140613-02-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : キーボードから入力された二つの浮動小数点数型の四則の結果を表示する
- ▶ 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)

値と型

□ Data(データ) と Code(コード)

- Data : C 言語のプログラムで「操作の対象」となる物
- Code : C 言語のプログラムで「操作を行う」命令部分
 - ▶ 共にメモリに記録されている (将來說明する)

□ 値(あたい)と型(かた)

○ Data は、「値」と「型」を持つ

- ▶ 型 : Data がどんな集合の要素で、どの様な演算ができるかを定める属性
- ▶ 値 : 型で指定された集合の要素の一つ (値が同じでも型が異れば異なる物)

○ Data の値と型の例

- ▶ 整数値の「1」: 型は「整数型」で、値は 1
- ▶ 文字の「1」: 型は「文字型」で、値は '1'
- ▶ 一文字からなる文字列「1」: 型は「文字列型(仮)」で、値は "1"

○ 型が異れば、できる「演算」も異なるし、「結果」も異なる

- ▶ 例1 : $9 + 1 = 10$, $'9' + 1 = ':'$, $"9" + 1 = ""$

式

□ 式：複数の Data から新しい Data を作る表現

- 例： $1 + 1 \rightarrow 2$, `*"abc"` \rightarrow `'a'` , `"abc" + 1` \rightarrow `"bc"`
- 「式」から Data を得る事ができる (つまり、値と型が得られる)
- 即値(式)：定数を表す表現
 - ▷ 1：整数型の 1 という値を持つ Data を表す「即値式」
 - ▷ '1'：文字型の '1' という値を持つ Data を表す「即値式」
 - ▷ "1"：文字列型の "1" という値(?)を持つ Data を表す「即値式」
- 演算子との組み合わせ
 - ▷ 整数の四則： $12 + 5 \rightarrow 17$, $12 - 5 \rightarrow 7$, $12 * 5 \rightarrow 60$, $12 / 5 \rightarrow 2$
 - ▷ 「+」,「-」,「/」,「*」は、二つの整数型の Data から一つの整数型の Data を作る
 - ▷ 文字列の先頭：`*"abc"` \rightarrow `'a'`：型が変化している事に注意 !!
- 関数呼出し：関数は Data を返す
 - ▷ 例 1：`s_input_int()`：キーボードから入力された整数に対応する Data
 - ▷ 例 2：`s_input_string()`：キーボードから入力された文字列
 - ▷ 例 3：`rand()`：呼び出す毎に異なる整数が返る

浮動小数点数型(再)

□ double 型

○ 小数点付きの数を表現する

▶ C 言語内での表現 : 小数点付きの数 (cf. 123.456)

▶ $2.225074 \times 10^{-308} < \text{double の絶対値} < 1.797693 \times 10^{308}$

○ 宣言 : double で行う

○ 計算 : 様々な数学的な関数ができる

▶ sin/cos/exp/log/etc.. cf #include <math.h>

○ (当然) 四則の計算ができる : $3.0/2.0 \rightarrow 1.5$ (cf. $3/2 \rightarrow 1$)

○ 比較 : 大小比較、等号、不等号が使える(整数と同じ)

▶ ただし、浮動小数点数同士の「==」は止めた方がよい

▶ その代わりに、比較する二数の差の絶対値(fabs)を取り、小さな正の浮動小数点数(ϵ)より小さいかを調べる

□ 浮動小数点数型の入出力 (当分は..)

○ 入力 : s_input.h の中の s_input_double を使う

○ 出力 : s_print.h の中の s_print_double を使う

浮動小数点数型と誤差(再)

- C 言語(一般に計算機で)の数値表現は「有限」でしかない
 - 浮動小数点数型の値もちろん「有限」な表現
 - ▶ 「実数」を正確に表現しているわけではない
- 丸め誤差
 - 実数の本当の値を浮動小数点数で表す必要がある
 - ▶ C 言語の中では、「その真値に近い浮動小数点数」に「丸めて」しまう
 - ▶ 「丸め誤差」がある
 - 計算誤差
 - ▶ 「浮動小数点数」での計算結果も「浮動小数点数」の範囲にはいかない
 - ▶ やはり、「丸めて」しまう
- 浮動小数点数型の扱い(誤差を意識して扱う)
 - 常に「誤差」を含む
 - ▶ 「誤差」は蓄積される可能性がある
 - 等号比較はしない
 - ▶ 「差」が「小さい範囲」にあれば、「等しい」と見做す

浮動小数点数の応用

□ 数値的解法(数値計算を用いた解法)

- 「問題の答え(数値)」を「近似的」に求める手法(誤差を含む)

 - ▷ <反> 「解析的解法」: 数学的に求める(誤差を含まない)

- [注意] 解析的解法があるのに、数値的に解くのは、余り望ましくない

 - ▷ 解析的解法が存在しなくても数値的な解法がある場合がある

 - ▷ 「解析的解法」と「数値的な解法」は相補的

□ 数値計算の初歩

- 方程式の解

 - ▷ 二分探索/ニュートン法

- 数値積分

 - ▷ リーマンの公式/台形公式/モンテ=カルロ法

定数の定義と define

□ シンボル定数の定義

- 定数に名前をつける事ができる
 - ▶ 「定数に名前を付ける」事を「シンボル定数の定義」と呼ぶ
- 定義方法
 - ▶ #define 定数名 定数值
- 定義例
 - ▶ #define PI 3.141592
 - ▶ #define EPSILON 0.000001

□ シンボル定数の効用

- マジックナンバーの排除
 - ▶ マジックナンバーとは：プログラムの中に散見される「意味不明(マジック)」な数値の事
 - ▶ マジックナンバーはプログラムを読み難くする
- マジックナンバーの代りにシンボル定数を利用する
 - ▶ 「定数名」に「意味のある名前」を使えば、読み易くなる
- 「共通の値」を「同時に変更する」事が可能になる
 - ▶ 定数定義を変更するだけ