

ソフトウェア概論 A/B

-- データ構造 --

数学科 栗野 俊一 / 渡辺 俊一

2013/10/11 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>
- 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

前回 (2013/09/27) の復習

□ 前回 (2013/09/27) の復習

- 標準 I/O ライブラリの続き

- 標準 I/O : どこでも利用(標準)できる入出力(I/O)ライブラリ

 - ▶ printf : 書式付き、便利出力関数 → 是非、使い熟せるようになりたい

 - ▶ scanf : 書式付き、便利入力関数 → 超便利ではあるが、初心者には辛い

- 文字列からの情報の抽出

 - ▶ sscanf : 文字列内から情報を取り出す (sprintf もあるが今回はパス)

- ファイルへの入出力

 - ▶ fopen/fclose : ファイルを利用するための関数(ファイルポインター)

 - ▶ fprintf/fscanf : ファイルに対する I/O (printf/scanf の眷属)

お知らせ

- 本日の予定

- データ構造

- ▶ 配列と構造体

- 本日の目標

- 演習

- ▶ 課題の提出

前回 (2013/09/27) の課題

□ 前回 (2013/09/27) の課題

○ 課題 1:

- ▶ ファイル名 : 20130927-1-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : `sscanf` による文字列から情報の取出し
- ▶ 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 : 20131011-2-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : `fopen/fclose/fprintf` によるファイルの書出し
- ▶ 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 3: は、今回の課題に回す

本日の課題 (2013/10/11)

□ 本日 (2013/10/11) の課題

○ 課題 1:

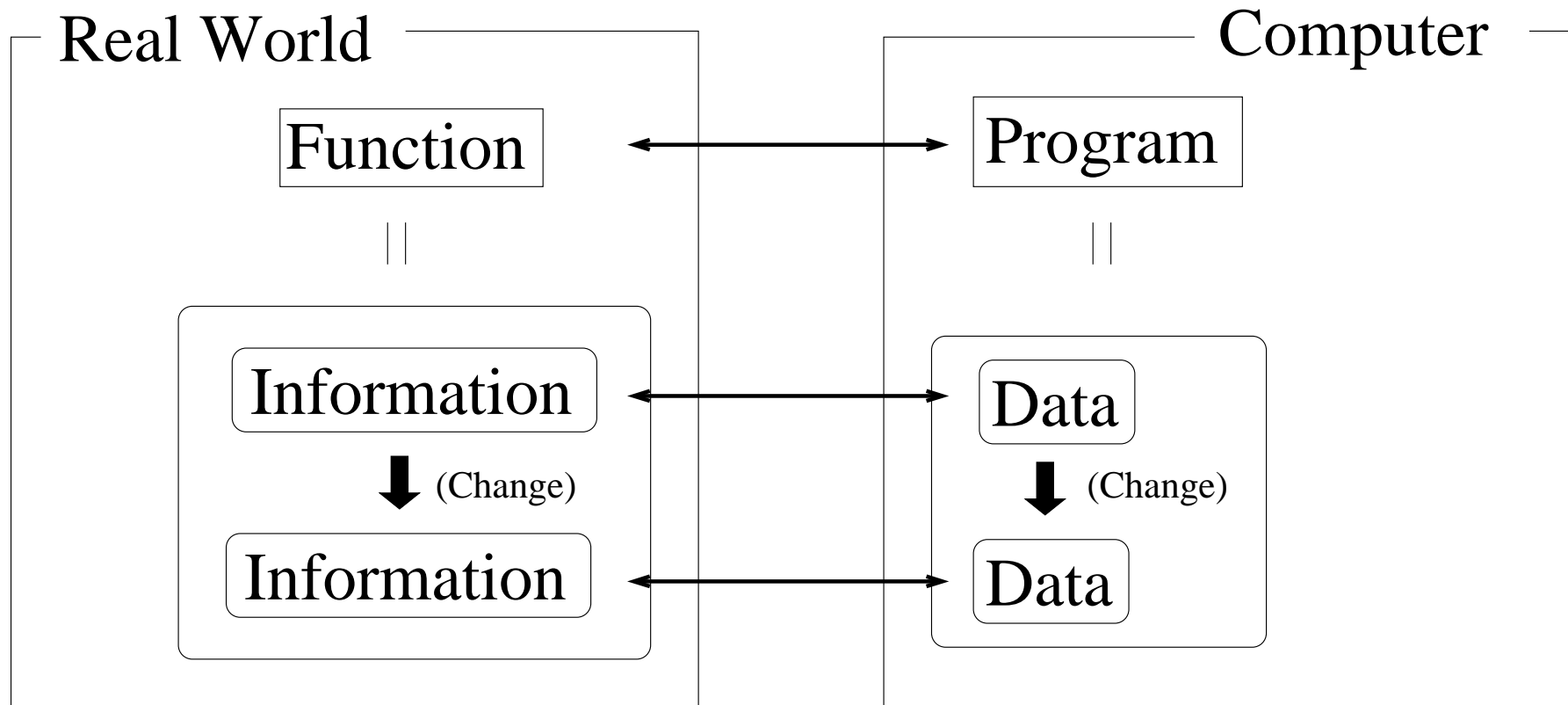
- ▶ ファイル名 : 20131011-1-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 極座標で表現されている点 Q から、それと原点に対して対称な点 R を求める
- ▶ ファイル形式 : テキストファイル(C 言語プログラムファイル)

○ 課題 2:

- ▶ ファイル名 : 20131011-2-QQQQ.c (QQQQ は学生番号)
- ▶ 内容 : 構造体を利用し、平行移動を行う関数を作成する
- ▶ 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)

「情報」を経由した「機能」の実現

- 「現実(Real World)」と「計算機(Computer)」の関係



- 機能の実現例

- 銀行口座 [001]

- ▶ データ間の対応 : Information → 残高 / Data → 整数値
- ▶ 操作の対応 : 10 万円振り込む / 100000 を加える
- ▶ 機能の対応 : 給料の振込 / 足し算

データ構造

□「プログラム」による「機能」の実現

- プログラム: 「データ(数値)」の「処理(変更)」*しか* できない
- 機能: 「情報」の「操作」によって実現される
 - ▶ どこでか「データ」と「情報」の *対応* が必要
 - ▶ 例: 文字 (ASCII Code): 文字コード(数値) と 文字(情報)の対応を行う[002]

□「情報」を「データ」の形にする

○「データ」による「情報」の「表現」を考える[003]

- ▶ 例 1: 平面上の点を「(x, y):直交座標系の座標」で表現する
- ▶ 例 2: 平面上の点を「(r, a):極座標系の距離と角度」で表現する

○表現が異れば、同じ「機能」を実現する場合でも、プログラムが異なる

- ▶ 例: 点 P と原点対象な位置にある点 Q を求める (例 1/2 で操作が異なる)

○「点」を表現するには、「二つの実数値の『対』」が必要

- ▶ 「点」には、『対』という「構造をもっている」と考えられる

□データ構造とは

○構造を持つデータとそれを作る要素の関係の事

- ▶ 既存のデータ表現から、新しいデータ表現を作る方法にもなっている

「点」のデータの構造の例

□ 平面上の点を扱う事を考える

○ x 座標と y 座標の組で「点」を表現

▶ 点 p1 の x, y 座標をそれぞれ p1x, p1y で表現してみる

○ 点の表示や、距離などは、普通に扱える [005]

○ 点そのものを操作する事を考えると..

▶ x 軸, y 軸, 原点に対象な点 .. [003,006]

▶ 特に関数にすると辛い [007,008]

□ 「点」を表すもの(データ構造)を考える

○ 構造体：複数のデータをまとめて扱うようにする仕組[009]

○ struct { 中身 };

▶ 毎回書くのは面倒なので、名前を付けてしまう typedef

○ 構造体の中身は、色々な型を並べる事ができる[010]

配列

□ 配列

- 同じデータが並んだ物を表現する仕組

 - ▶ 例: `double a0,a1,a2 -> double a[3]`

- 配列名：データの並びが入る変数の代表名

 - ▶ 添字「`[+ 整数値]`」を付けて、要素が参照できる

- 配列の宣言

 - ▶ 配列を利用する(宣言する)場合は、「`配列名[サイズ]`」の形にする

 - ▶ サイズ個数の変数がまとめて用意される

 - ▶ 参照する場合は `0 ~ サイズ-1` まで

 - ▶ 例: `int ary[10];` とすると `ary[0] ~ ary[9]` が使える

データ構造とプログラム

- データ構造とプログラム構造は対応している
 - 基本プログラム構造：順接, 繰返し, 条件分岐
 - データ構造：構造体, 配列, 共用体(後述)
- データ指向
 - データ型をきちんと考えると、プログラムが自動的にできる
 - ▶ まず、データ型をきっちりと考える
- オブジェクト指向
 - 更にデータ型とその型に対するプログラムをまとめて行くしくみ
 - ▶ 今回は言葉だけ紹介 (C++ 言語や java では中心概念)