ソフトウェア概論 A/B

-- データ構造 1 : 構造体/配列 --

数学科 栗野 俊一 / 渡辺 俊一

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 : 20151016
  - 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(2015/10/09)の内容

- if 文の本当の姿
  - else 節は省略可能 / ブロック({}) も不要 : ぶら下がり構文
- 条件の正体 : 条件式 → 整数値を取る式
  - 条件の判断 : 0 の時は偽 / それ以外の値は真
  - 条件式の値 : 偽を表す時は 0 になり、それ以外は 1 の値を取る
  - 論理演算子
    - && : 論理積 (両方真[0以外] の時のみ真[1])、それ以外は偽[0]
    - || : 論理和 (どちらか一方でも真[0以外]なら真[1])、それ以外は偽[0]
    - ! : 否定 (真の時は偽、偽の時は真 / 条件を反転する)
- スカッシュゲームの構造
  - スカッシュゲームの世界をシミュレート
    - 時間が刻々と進み、それに応じて、世界(の状態)が変化
    - 「世界の変化」は、「変数の値の変化」で表現

お知らせ

- 本日の予定
  - データ構造 (1)
    - 構造体と配列
  - スカッシュゲーム (3)
- 本日の目標
  - 演習
    - 課題の提出

前回 (2015/10/09) の課題

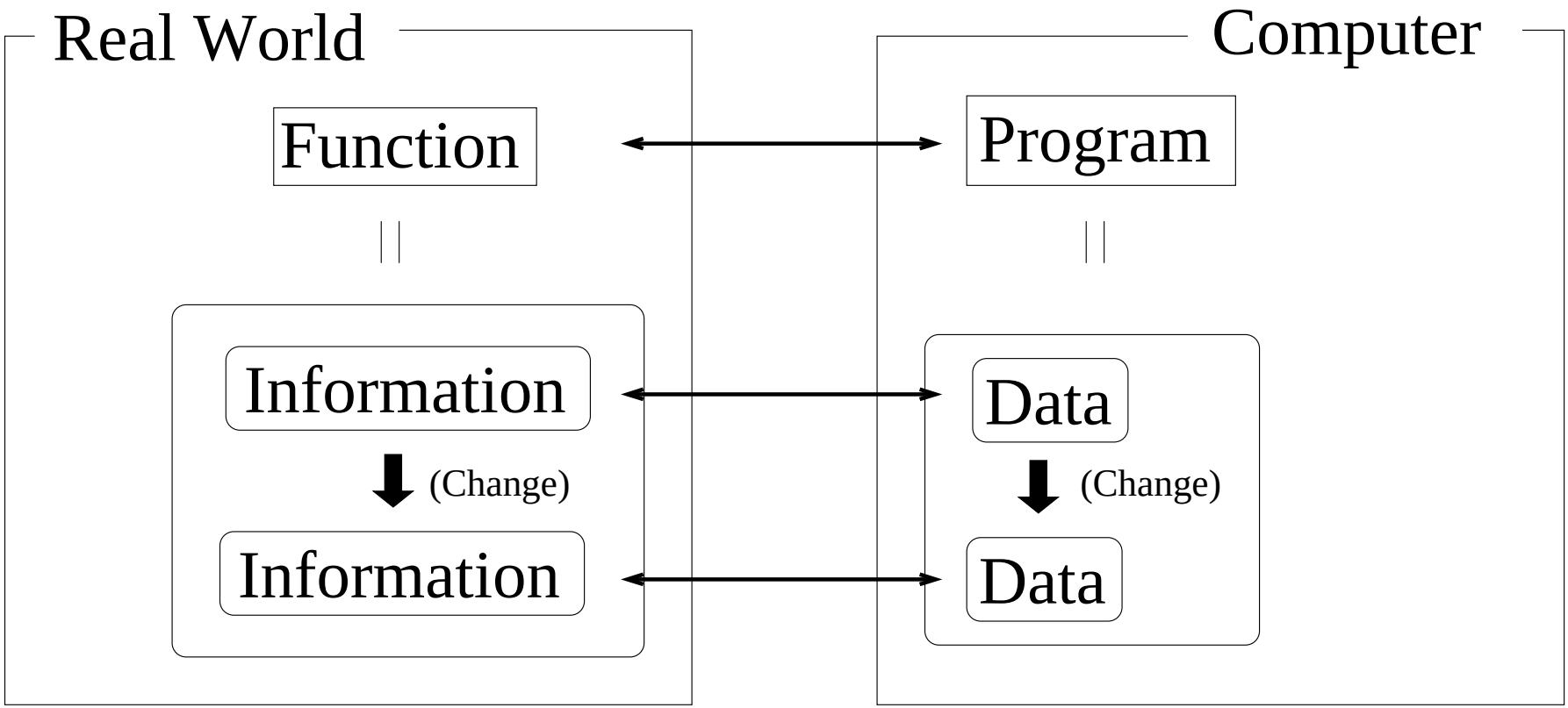
- □ 前回 (2015/10/09) の課題
  - ○ 課題 20151009-01:
    - ▹ ファイル名 : 20151009-01-QQQQ.c (QQQQ は学生番号)
    - ▹ 内容 : 三つの整数の比較(if 構文版)
  - ○ 課題 20151009-02
    - ▹ ファイル名 : 20151009-02-QQQQ.c (QQQQ は学生番号)
    - ▹ 内容 : 三つの整数の比較(論理積版)
  - ○ 課題 20151009-03:
    - ▹ ファイル名 : 20151009-03-QQQQ.c (QQQQ は学生番号)
    - ▹ 内容 : 真偽表
  - ○ 課題 20151009-04
    - ▹ ファイル名 : 20151009-04-QQQQ.c (QQQQ は学生番号)
    - ▹ 内容 : ド･モルガン
- □ ※
  - ○ 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)

本日の課題 (2015/10/16)

- 本日 (2015/10/16) の課題
  - 課題 20151016-01:
    - ファイル名 : 20151016-01-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
  - 課題 20151016-02:
    - ファイル名 : 20151016-02-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
  - 課題 20151016-03:
    - ファイル名 : 20151016-03-QQQQ.c (QQQQ は学生番号)
    - 内容 : 3 次元ベクトルの差の計算
- ※
  - 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)

「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 : Infomation → 残高額 / 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

○これでわかる : 「情報(集合)」を「データ(数値)」に対応付ける事

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
    - ▷例 2: 平面上の点を「(r, a):極座標系の距離と角度」で表現
  - ○「表現」が異れば、同じ「機能(操作)」を実現する場合でも、「プログラム(処理)」が異なる
    - ▷例：点 P と原点対象な位置にある点 Q を求める「機能」の実現 (例 1 と 例 2 で「処理」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
    - ▷注意：ただ「『対』という『形』」だけでは意味がない、「操作」まで含めて考える必要がある
- □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えるようにする仕組[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扱う仕組(class 概念)を持つ
    - 今回は言葉だけ紹介 (C++ 言語や java では中心概念)