コンピュータとネットワーク

第11回 ソフトウェアの分類～OSとプログラム言語

石井 健太郎
1308研究室・kenta@isc.senshu-u.ac.jp

スケジュール

4月11日　第1回「イントロダクション」
4月18日　第2回「コンピュータとその利用～ビジネスと情報システム」
4月25日　第3回「コンピュータ小史～情報の表現」
5月9日　第4回「コンピュータ小史～情報の表現」
5月16日　第5回「情報の表現」
5月23日　第6回「文字コードと日本語処理～中央処理装置」
5月30日　第7回「中央処理装置～記憶装置と入出力装置」
6月6日　第8回「記憶装置と入出力装置～論理回路」

スケジュール

6月13日　第9回「論理回路」

6月20日　第10回「中間テスト」

6月27日　第11回「ソフトウェアの分類～OSとプログラム言語」

7月4日　第12回「テストのフィードバック、
グラフによる表現～通信ネットワーク」

7月11日　第13回「インターネットとTCP/IP」

7月18日　第14回「セキュリティ」

7月25日　第15回「まとめと授業内テスト」

ソフトウェアの分類

p.145～

ソフトウェアとは

- ハードウェア: コンピュータの機械そのもの
  または機械を構成する部品
- ソフトウェア：コンピュータを動かすための情報
  （プログラム・データ）

- コンピュータはハードウェアとソフトウェアがそろってはじめて機能する

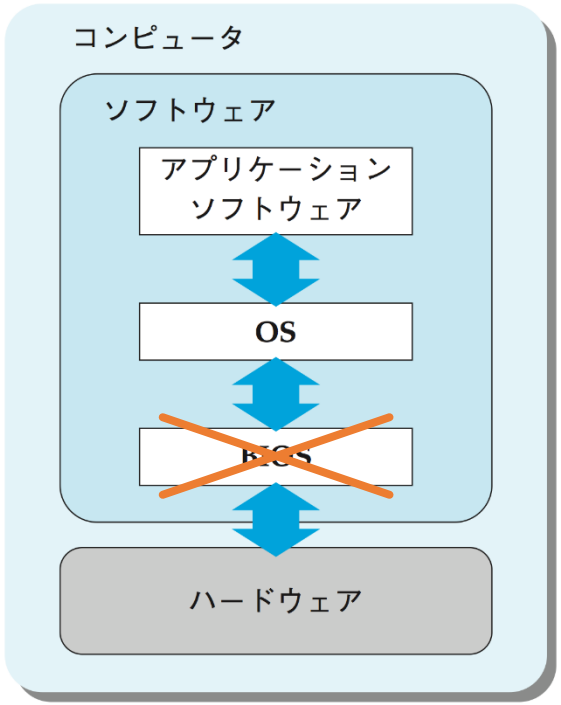
ソフトウェアの役割と重要性

- ソフトウェアがさまざまな機器や情報システムを利用する際に重要な役割を果たす
  - 情報システムにかかるコストの大半がソフトウェア関連
  - 信頼性が非常に重要

- ソフトウェアの重要性はますます高まっている
  - エンジニアでなくともソフトウェアについて学ぶことの意義は大きい

ソフトウェアの種類

- BIOS（Basic Input/Output System）
- OS（Operating System）
- アプリケーションソフトウェア



BIOS（Basic Input/Output System）

- メモリやハードディスクなどの装置に不具合がないかどうかチェックし，コンピュータに接続された周辺機器を制御するプログラム群のこと

- 電源を切っても内容が消えないROM（Read Only Memory）に組み込まれており，コンピュータの電源を入れたときに最初に動くプログラム

- 最近のOSはBIOSを使用せずに入出力装置を制御する
  - BIOSは起動時のみに使用される
    - そのBIOSもUEFIへの移行が進んでいる

OS（Operating System）

- ハードウェアの差異を隠ぺい
- コンピュータシステム全体を管理
  - キーボード入力や画面出力といった入出力機能
  - ディスクやメモリの管理
- 多くのソフトウェアで共通に利用される基本的な機能を提供
- 利用者が使いやすいインタフェースを提供

- 現在のパソコンやタブレット端末の代表的なOS
  - Windows
  - MacOS
  - Linux
  - Android
  - iOS

（クライアントの）アプリケーションソフトウェア

- 代表的なアプリケーションソフトウェア
  - メール
  - ウェブブラウザ
  - SNS
  - テキストエディタ・ワープロ
  - 表計算
  - プレゼンテーション
  - データベース
  - セキュリティ
  - 画像処理・管理
  - データ解析

サーバのアプリケーションソフトウェア

Web サーバ（HTTP サーバ）	WWW による情報送信機能をもったソフトウェア（Apache など）
メールサーバ	電子メールを配送するためのソフトウェア（Sendmail など）
FTP サーバ	FTP（File Transfer Protocol）を使用してファイルの送受信を行うソフトウェア
ファイルサーバ	LAN や WAN などのネットワーク上で，ファイルを共有するためのソフトウェア（Samba など）
プリントサーバ	ネットワーク上に配置されたあるプリンタを，複数のクライアントで利用できるように制御するソフトウェア

新たなソフトウェアの提供形態

- SaaS（Software as a Service）
  - ソフトウェアを提供者(プロバイダ)側のコンピュータで稼働させ，
    ユーザはそのソフトウェア機能をインターネット経由で
    サービスとして使用する形態
  - cf. 通常は，自分のもつコンピュータ上でソフトウェアを稼働させ利用する形態
  - Googleドキュメント，Office Web Appsなど

- クラウドコンピューティング
  - インターネット上に配置したコンピュータ資源を遠隔でも使用できるようにしたもの
  - SaaSを実現するために用いられている

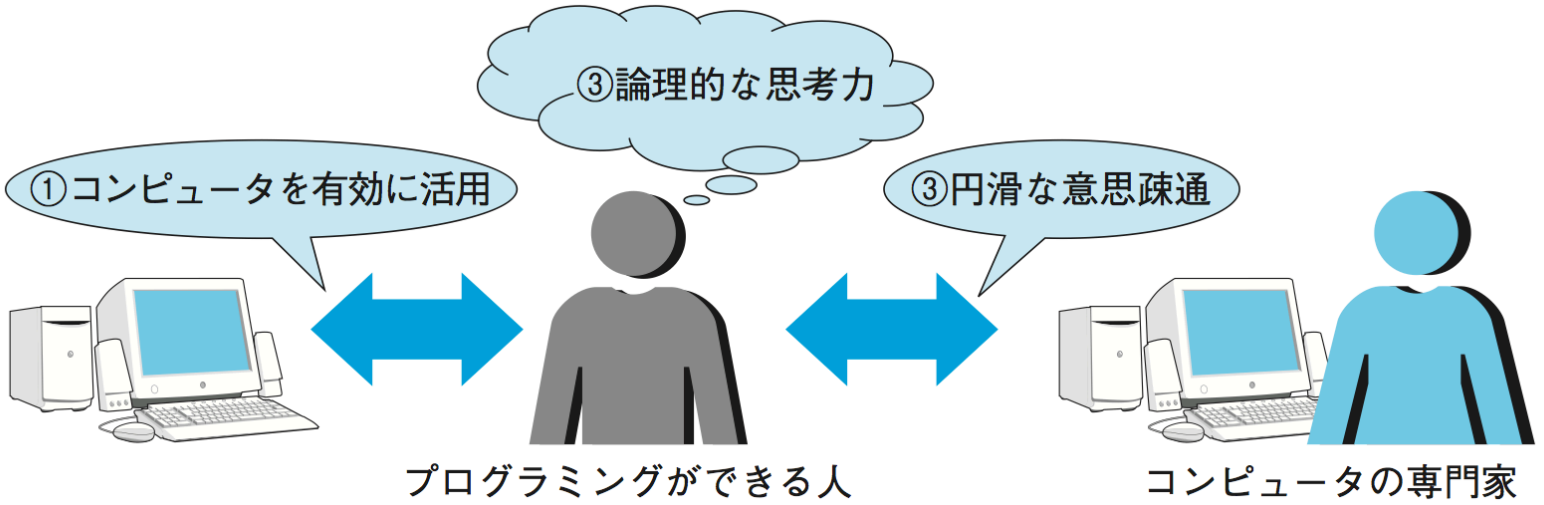
プログラミング

p.157～

コンピュータに仕事をさせるには

- コンピュータに仕事をさせるにはプログラムが必要
  - コンピュータにやらせる作業自体を理解する
  - コンピュータができる作業の限界を知る

- プログラミングができる人の利点



プログラミングをするために

- 対象とする作業の概要を理解
  - コンピュータにやらせたい作業の範囲の設定
  - 作業の手順を明確化
    - データの変化に注目
    - 処理の流れにも注目

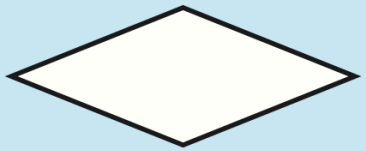
- コンピュータの持つ制約を理解
  - 指示されたことしかしない
  - 曖昧な表現は理解できない
  - 長文を読解するのが苦手

フローチャート（流れ図）

- 作業の流れを図で表したもの

- フローチャートの利点
  - 図を用いて視覚的に表すことにより，文章だけよりも理解しやすい
  - 自分の考えを明確にできるだけでなく，他人への説明が容易になる
  - 間違いが発見された場合の修正が容易である
  - 同じような処理があった場合に再利用できる

フローチャートで利用される記号

種類	意味	記号
端子	最初と最後に使用	
処理	計算や代入など	
入出力	ファイルへの書き出しや読み込み	
判断	IF などの分岐	

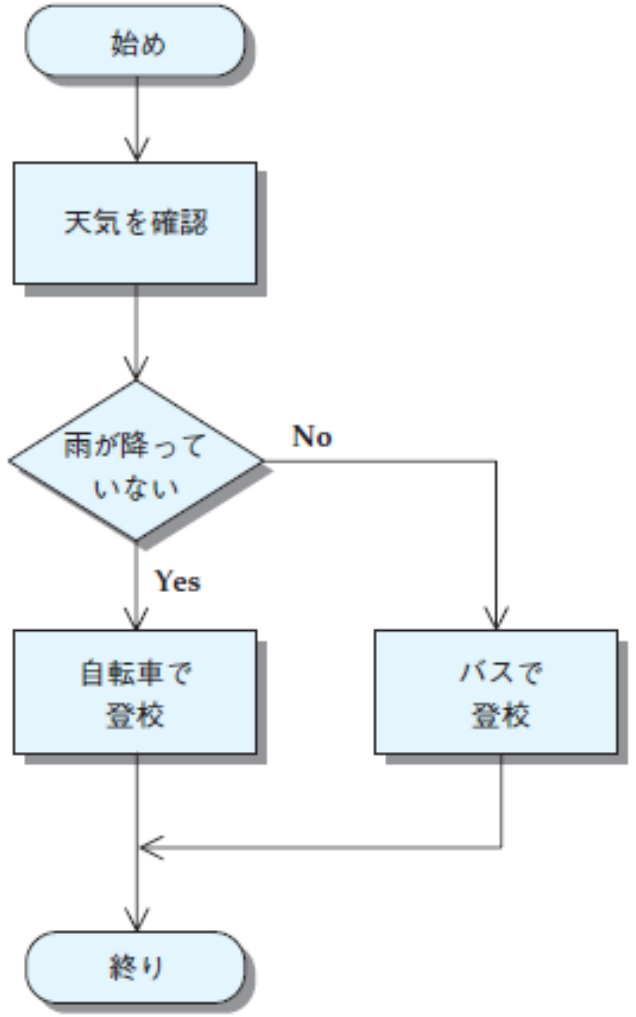
フローチャートを作成する際のポイント

- フローは，上から下へスッキリと流れるようにする
- フローチャートが見づらくなるので，
  矢印が横へ流れる（横から入る）パターンは避ける
- 「処理」および「判断」等への入力は原則としてひとつ
- 「処理」の出力も原則としてひとつ
  - ただし，「判断」での分岐は，原則として2つ
- 基本的に線は交差させない
  - どうしても交差してしまう場合は，交差する線のつながりを明確にする

フローチャートを作成してみよう

- 家を出るときに天気を確認し，雨が降っていなければ自転車で大学に行くが，雨が降っていればバスで行く

フローチャートを作成してみよう



フローチャートを作成してみよう

- 家を出るときに，天気を確認し，雨が降っていなければ自転車で大学に行くが，天気予報を確認して予報が雨の場合は歩いて行く．雨が降っていればバスで行くが，警報が出るぐらいの大雨の場合は大学へは行かない．しかし，そもそも大学のポータルサイトで休講情報を確認して，授業が休講ならば大学へは行かない．

プログラミング

- プログラムは「アルゴリズム」と「データ構造」の組合せ
  - 互いに密接に関連し合ったものであり，別扱いすることはできない

- プログラミングとは
  - アルゴリズムを考えると同時に，データ構造を考え，
    （最終的に）プログラミング言語を用いてプログラムを作成するプロセス
    - アルゴリズムとデータ構造を先に考え，日本語で表現してもよい

アルゴリズム (Algorithm)

- JIS(Japan Industrial Standard, 日本工業規格)の定義
  - 明確に定義された有限個の規則の集まりであって,
    有限回適用することにより問題を解くもの

- コンピュータを使ってある特定の目的を達成するための処理手順のこと

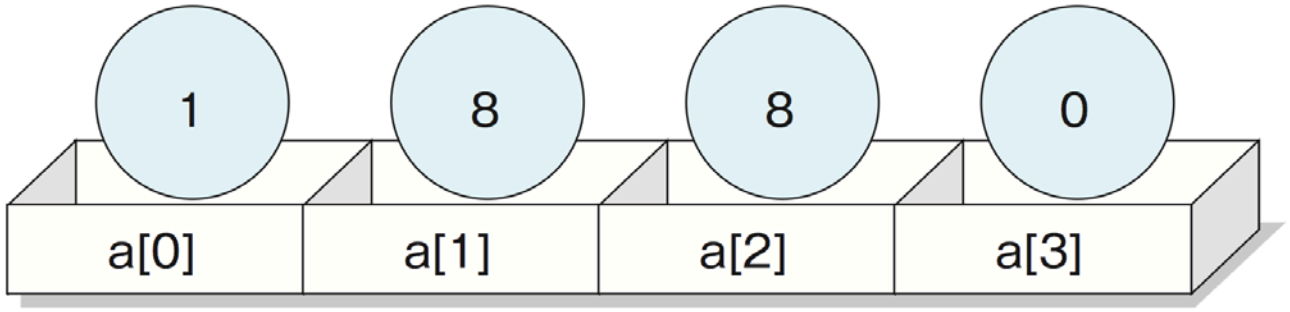
アルゴリズムの例

- 5個のデータに対するアルゴリズムの例．何を処理するアルゴリズムかわかりますか？

- 手順1:　　1番目のデータと2番目のデータを比較し，
  1番目のデータのほうが大きければ，1番目のデータと2番目のデータを入れ替える
- 手順2:　　2番目のデータと3番目のデータを比較し，
  2番目のデータのほうが大きければ，2番目のデータと3番目のデータを入れ替える
- 手順3:　　3番目のデータと4番目のデータを比較し，
  3番目のデータのほうが大きければ，3番目のデータと4番目のデータを入れ替える
- 手順4:　　4番目のデータと5番目のデータを比較し，
  4番目のデータのほうが大きければ，4番目のデータと5番目のデータを入れ替える
- 手順5:　　1度も入れ替えが発生しなかったら，終了する
  入れ替えが発生していた場合は，手順1から繰り返す

データ構造 (data structure)

- データを格納する際の構造
- アルゴリズムの実行が効果的になるデータの配置方法がある
- 例えば，配列 (array)
  - 同じ型のデータを連続的に並べたデータ形式のこと
    - 各データをその配列の要素という
    - それらは添字(インデックス)で識別される
  - 並び替え(ソート)をするのには向いている



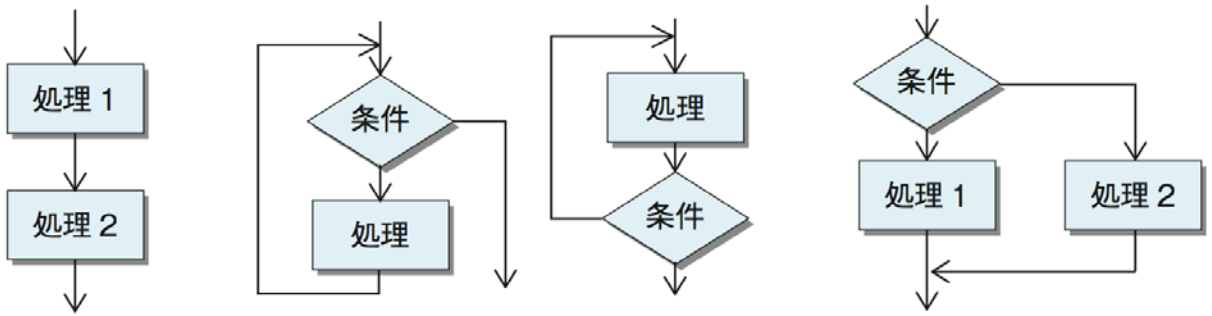
データ構造

- 詳細は「アルゴリズムとデータ構造1/2」で！

名称	特徴
リスト	データのつながり情報をもっている
木構造	木のように枝分かれしたリストである
スタック	データを山のように積み上げて一時保存する
キュー	データを行列のように並べて一時保存する

構造化プログラミング

- 基本となる考え方
  - 大きなプログラムは手に負えないので，小さく簡単なプログラムに分解して作り，それを合成して構成する（部品化）
  - 順次（連接）・繰返し（反復）・選択の3つの基本構造だけを使用する



(a) 順次　(b) 繰返し　(c) 選択

- 飛越し（go to）はプログラムの構造を複雑にするので，なるべく使用しない

プログラミング言語

- ソフトウェアの設計図に当たるソースコードを記述するための言語
  - コンパイラ型言語
    - コンパイラを用いて，ソースコードが完成時に一括して翻訳作業を行う
  - インタプリタ型言語
    - インタプリタを用いて，同時通訳のようにリアルタイムに実行することにより，
      あたかもソースコードを直接実行しているように処理する
  - スクリプト言語（明確な定義はなく多くのインタプリタ型言語がこう呼ばれる）
    - 簡易な言語仕様をもち，プログラムを簡単に記述することができる

代表的なプログラミング言語

- C言語
  - 豊富な演算子やデータ型，制御構造をもち，古くからよく使われている
- Java
  - オブジェクト指向性を備え，ネットワーク環境で利用されることを強く意識
- COBOL
  - 企業の基幹業務に使われる汎用機のプログラムに使われている（らしい）
- FORTRAN
  - 科学技術計算を目的としたプログラミング言語

代表的なプログラミング言語

- Visual Basic
  - 「フォーム (form) 」と呼ばれるウィンドウに部品を張り付け，
    その設定や部品間の関係を指定することで開発することができる
- Perl, Ruby, Python
  - インタプリタ型言語(スクリプト言語)で，PerlはCGIの開発によく使われる
- JavaScript
  - オブジェクト指向のスクリプト言語で，Webアプリケーションの開発に利用される