

K<sup>o</sup>GEI コンピュータ応用学科

## 「情報処理概論」 CS1年・後期・必修(3)

---

担当: 浦谷・荒井・行谷  
{uratani, arai, yukiya}@cs.t-kougei.ac.jp  
<http://gakunai.t-kougei.ac.jp/lecture/cs/>

K<sup>o</sup>GEI コンピュータ応用学科

## 第3回: プログラム開発

---

- プログラム開発言語
  
- プログラム開発の手順
  
- 小テスト

---

K<sup>o</sup>GEI コンピュータ応用学科

## プログラムとは？

---

- ソフトウェア: コンピュータシステムの機器などをハードウェア(金物)というのに対し、コンピュータを利用・活用する技術のこと
- プログラム: 問題や仕事の内容と手順をコンピュータ開発言語で記述したもの
- プログラミング: プログラムを作成すること
- プログラム指示書(仕様書): システム開発プロセスで作成される内部設計仕様書(の一部)問題や仕事の内容と手順を記述したもの

---

K<sup>o</sup>GEI コンピュータ応用学科

## フリーウェアって？

---

- パッケージソフト: 箱に入った形態で販売店などで売られている通常のソフトウェア
- オンラインソフト: インターネット上で公開されているソフトウェア
  - シェアウェア: いわゆる「お試し版」。気に入れば(長く使うつもりなら)使用料を払うもの。
  - フリーウェア: 作者が営利を目的としないで公開しているソフトウェア。無料で使用することを認めているが、改変や再配布は原則禁止されている。

---

K<sup>o</sup>GEI コンピュータ応用学科

## プログラム内蔵方式

---

初期のコンピュータはデータとプログラムは明確に区別され、プログラムは配線によって行われていた(ワイヤードロジック)。フォン・ノイマンはプログラムをコンピュータの記憶装置に入れ、自動的に仕事をさせる方法を考案した。

- **プログラム内蔵方式**: ハードウェアの改変をすることなくコンピュータにさまざまな機能をもたせることが可能となった。つまり、**ソフトウェア**(という概念)の誕生である。
- **データフロープロセッサ**など、反対にハードウェアによるプログラムを残すことで高速化を図るものもある。現在の**非ノイマン型**コンピュータ(プロセッサ)である。

---

K<sup>o</sup>GEI コンピュータ応用学科

## プログラム開発言語

---

- **低水準言語**: (低級言語)
  - 機械語: 0と1で表される命令語そのもの
  - アセンブリ言語: 機械語と1対1に対応しているニューモニックコード(表意記号)によって記述するプログラム言語; CPUの機種に依存する
- **高水準言語**: (高級言語)
  - 人間が日頃使っている言葉(自然言語)に近い
  - FORTRAN, COBOL, BASIC, Pascal・・・

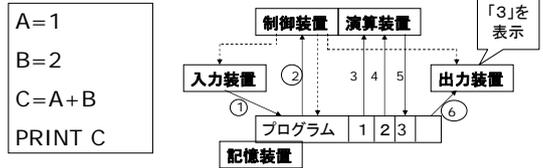
---

## アセンブリ言語

- 機械語と1対1に対応
- 命令コード、ラベル、オペランド(番地)からなる
- 命令コードは通常、英語の単語の短縮形
  - 加算はADD、減算はSUB、LD, ST, GRO など
- CPUやメモリに対する細かな使い方の指示が可能
- スピードアップやメモリ節約などの細かいテクニックが使える
  - OSやゲームソフトの開発に今でも利用

## 高水準言語

- 日頃使っている言葉(自然言語)に近いプログラム言語。高級言語ともいう。
- 例:「1+2の計算」をBASICで書くと



## FORTRAN

- 1957年にIBMが開発;最初の高水準言語
- 科学技術計算向き
  - 豊富な数値計算用ライブラリと組み込み関数
  - 実行速度が速い
  - ファイル処理などの機能が弱い

## FORTRANによるプログラム例

```

c 1から10の足し込み合計
i=0
n=10
isum=0
10 i=i+1
isum=isum+i
if(i.ne.n)goto 10
20 format(*1から10の足し込み合計=*,i5)
stop
end
i=カウンタ
isum=1~10の合計欄
i≠nの時、10にとぶ
合計を出力
出力の書式(整数5桁)
    
```

## COBOL

- 1959年にCODASYL(米国:データシステムズ言語協会)が開発
- 現在でも、稼動している商用プログラムの大半
- 事務処理向き;
  - 英語表示のため記述がしやすい
  - ファイル操作、データベース操作などの機能が豊富
  - 最初に、見出し部、環境部などを定義する必要があるため、プログラムが冗長になる

## COBOLによるプログラム例

```

IDENTIFICATION DIVISION.
PROGRAM-ID. SAMPLE.
*1から10までの足し込みの合計
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
77 I PIC 9(5) VALUE 0.
77 N PIC 9(5) VALUE 10.
77 ISUM PIC 9(5) VALUE 0.
PROCEDURE DIVISION.
HAJIME.
COMPUTE I=I+1.
COMPUTE ISUM=ISUM+I.
IF I NOT = N GO TO HAJIME.
DISPLAY *1から10までの足し込みの合計は* ISUM.
STOP RUN.
見出し部
プログラム題名
環境部
データ部
記憶場所の定義と
初期値の設定
手続き部
    
```

K●GEI コンピュータ応用学科

## BASIC

- 1964年に米国ダートマス大学で開発
- FORTRANをより簡易に表現することを目的
- インタープリタ型言語
- プログラムの初心者用言語
  - インタープリタ型なのでデバッグが容易
  - 文字操作、グラフィック操作が容易
  - パソコンで広く採用
  - Windows環境の開発言語(Visual BASIC)

K●GEI コンピュータ応用学科

## BASICによるプログラム例

```

10 I=0
20 ISUM=0
30 I=I+1
40 ISUM=ISUM+I
50 IF I<10 GOTO 30
60 PRINT "1から10を足した合計は...", ISUM
70 END
    
```

K●GEI コンピュータ応用学科

## Pascal

- 1971年、スイスチューリッヒ工科大学のN.Wirth教授によって公開
- 教育目的に開発された**構造化プログラミング**言語

```

program factorial(input, output);
  var i, f, n:integer;
  begin
    f:=0;
    n:=10;
    for i:=1 to n do f:=f+i;           Pascalによる
    writeln(output, f);               プログラム例
  end.
    
```

K●GEI コンピュータ応用学科

## C

- 1972年に米国AT&T社のベル研究所で開発
- 低水準の命令まで記述可能
  - OSの開発も可能 → UNIX
  - ハードウェアに近い処理ができる
  - 移植性が極めて高い
  - 構造化プログラミングに適している
  - 豊富な演算機能
  - UNIXの広まりとともに広く使用されている

K●GEI コンピュータ応用学科

## C によるプログラム例

```

#include<stdio.h>                /* 標準関数の組み込み */
int main()                       /* 実行単位 */
{                                 /* プログラムの始まり */
  int i, f, n;                   /* 変数の宣言 */
  f = 0;
  for(i = 1; i <= 10; i++)
  {
    f = f + i;                   /* 1から10までの加算 */
  }
  printf("1から10の足し込み合計 = %d", f); /* fの値の出力 */
}                                 /* プログラムの終わり */
    
```

K●GEI コンピュータ応用学科

## オブジェクト指向型言語

- 手続きとデータを1つにまとめて部品化(オブジェクト; 物体化)するという指向
  - 構造化プログラミング: 機能別に手続きとデータを分けるプログラミング手法
  - カプセル化: オブジェクトの内部構成を隠蔽し、公開メソッドからのみ情報にアクセスする手法
- プログラムの生産性向上が目的
- オブジェクトの中に予め機能(メソッド)や性質(プロパティ)を組み込む。メッセージを与えることでオブジェクト(プログラム)を実行
- SmallTalk, C++, Java

K●G●E●I コンピュータ応用学科

## Java

- 1995年に米国サン・マイクロシステムズ社によって公開。
- C++をベースに開発
- 通信ネットワークを利用するアプリケーションソフトウェアの制作用に開発
- 仮想コンピュータ(JVM)を前提に開発されているのでOS、マイクロプロセッサの違いを吸収しやすく、移植性が高い

K●G●E●I コンピュータ応用学科

## Java によるプログラム例

```

public class Sample_ito10
{
    public static void main(String args[] )
    {
        int i, f;

        f = 0;

        for(i = 1; i <= 10; i++)
        {
            f = f + i;
        }

        System.out.println("1から10の足し込み合計 = " + f);
    }
}
    
```

K●G●E●I コンピュータ応用学科

## その他の(高水準)プログラム言語

- LISP: リストを基本単位とする関数型の言語  
人工知能分野でよく使われ、AI用アセンブリ言語とも呼ばれる
- PL/1: IBMによって開発 (FORTRAN+COBOL)
- Ada: アメリカ国防総省が開発 (PASCALの拡張)
- Perl: インタープリタ型の言語。CGI(Common Gateway Interface)やシステム管理、テキスト処理などのプログラムの作成に向く。美しさより実用性がモットー。
- Prolog: 論理型、宣言型のプログラム言語
- ALGOL, APL, Smalltalk, Modula, C++

K●G●E●I コンピュータ応用学科

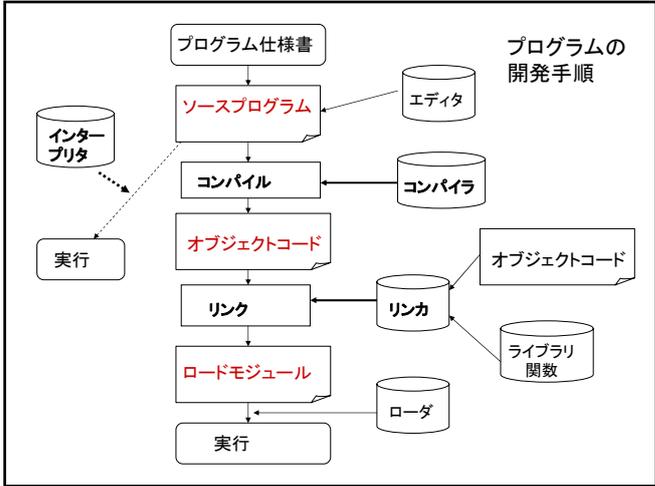
## プログラム開発の手順

- プログラム仕様書(内部設計の成果物)に基づいて実行
  1. プログラムの作成  
テキストエディタ(メモ帳、秀丸エディタ、vi, emacsなど)によりプログラムを記述  
**ソースプログラム(ソースコード)**と呼ばれる
  2. コンパイル
  3. リンク

K●G●E●I コンピュータ応用学科

## コンパイルとリンク

2. **コンパイル**  
ソースプログラムを機械語に翻訳する作業  
この作業を行うプログラムを**コンパイラ**という  
**オブジェクトプログラム(ネイティブコード)**が生成される
3. **リンク**  
別のプログラム(ライブラリ、サブルーチン)を利用するために、それを組み込んで実行可能なコード(**ロードモジュール**)を作成すること  
この作業を行うプログラムを**リンカ**という



## 言語プロセッサのまとめ

- 言語プロセッサ: ソースコードを機械語に翻訳するソフトウェア

言語プロセッサ	概要
アセンブラ	アセンブラ言語で書かれたソースコードを1対1で機械語に翻訳
インタプリタ	ソースコードを1行ずつその場で機械語に翻訳しながら実行するもの
コンパイラ	ソースコードをまとめて機械語に翻訳し、実行ファイルを作るもの

## オブジェクト指向の次は？

- アスペクト指向
  - 「横断要素」(関心事)と呼ばれるオブジェクトをまたがった処理(例えばロギング処理)を、オブジェクトに分散させるのではなく単一のモジュールにまとめて記述しておき、後からコンパイラなどのツールを用いてプログラムを結合しよう、というもの
- エージェント指向
  - データや手続き、さらにそれらを最適に活用できる知識を持ったモジュール。自律・協調・学習機能を持ったモジュールがエージェントと呼ばれる。それを用いてシステムを構築しようという考え方。

## 次回(第4回)は

- アルゴリズムについて