

プログラミング言語

c言語

Pascal

FORTRAN77

c / Pascal / FORTRAN77の特徴

- ・ 高水準言語
- ・ 汎用型プログラミング言語
- ・ 構造化言語
(オブジェクト指向の上位言語をもつ)
- ・ 翻訳型(コンパイラ)で実装
(インタプリタの実装も可能)

c言語の特徴

- ・ 幅広い分野で利用されている
- ・ 低水準な処理が可能
- ・ 言語仕様が単純
- ・ ライブラリによる拡張が容易
- ・ 多彩なデータ構造
(ポインタ、構造体など)

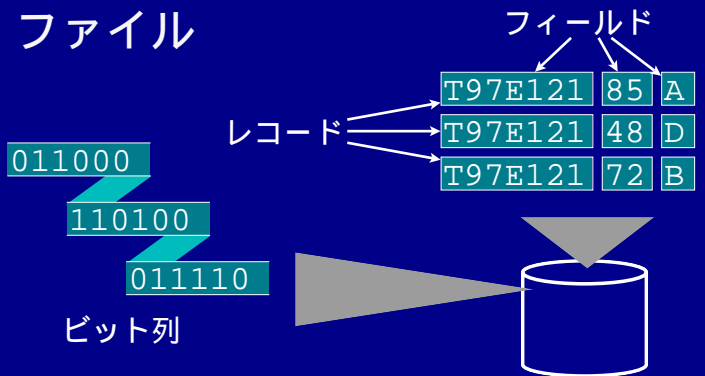
Pascal の特徴

- ・ プログラミングの教育に適する
- ・ 言語仕様が厳格で移植性が高い
- ・ 多彩な制御構造
アルゴリズムをそのままプログラムすることができる
- ・ 柔軟な入出力機構をもつ

FORTRAN (77) の特徴

- ・ 最古の高水準言語
 - ・ 数値計算に最適
複素数を扱うこともできる
 - ・ 過去の膨大な資産を利用できる
 - ・ 計算機科学以外の多数ユーザ
- 77 構造化言語
90 オブジェクト指向言語

プログラム(モジュール)の保存



プログラムのフレームワーク

[例題]

標準出力に

Hello, world

と表示するプログラムを作成せよ

プログラムのフレームワーク

Pascal

```
program hello(output);  
begin  
  writeln('Hello, world')  
end.
```

プログラムのフレームワーク

Pascal

```
program hello(output);
begin
  writeln('Hello, world')
end.
```

用意されているキーワード(予約語)
特別な意味を持つ記号

プログラムのフレームワーク

Pascal

```
program hello(output);
begin
  writeln('Hello, world')
end.
```

ユーザ定義名前付きオブジェクト
ユーザ定義リテラル

プログラムのフレームワーク

Pascal

```
program◇hello(output);begin◇
writeln('Hello, world')◇end.
```

「◇」(ホワイトスペース)「;」「begin」
「end」「end.」「{」「}」で区切られた
バイト列
「行・桁(レコード・フィールド)」の概念なし

プログラムのフレームワーク

FORTRAN77

```
PROGRAM HELLO
WRITE(6, *) 'HELLO, WORLD'
STOP
END
```

プログラムのフレームワーク

FORTRAN77

```
◇◇◇◇◇PROGRAM HELLO
◇◇◇◇◇WRITE(6, *) 'HELLO, WORLD'
◇◇◇◇◇STOP
◇◇◇◇◇END
```

ユーザ定義名前付きオブジェクト
ユーザ定義リテラル

プログラムのフレームワーク

FORTRAN77

```
◇◇◇◇◇PROGRAM HELLO
◇◇◇◇◇WRITE(6, *) 'HELLO, WORLD'
◇◇◇◇◇STOP
◇◇◇◇◇END
```

用意されているキーワード(予約語)
特別な意味を持つ記号

プログラムのフレームワーク

FORTRAN77

```
◇◇◇◇◇PROGRAM HELLO
◇◇◇◇◇WRITE(6, *) 'HELLO, WORLD'
◇◇◇◇◇STOP
◇◇◇◇◇END
```

システムに依存するリテラル

プログラムのフレームワーク

FORTRAN77 1行(1レコード)1文
コメントフィールド 実行文フィールド

```
◇◇◇◇◇PROGRAM HELLO
◇◇◇◇◇WRITE(6, *) 'HELLO, WORLD'
◇◇◇◇◇STOP
◇◇◇◇◇END
```

文番号フィールド
継続行フィールド

72桁まで
行・桁の概念あり

プログラムのフレームワーク

C言語

```
#include <stdio.h>
int main(void) {
    printf("Hello, world\n");
    return 0;
}
```

プログラムのフレームワーク

C言語

```
#include <stdio.h>
int main(void) {
    printf("Hello, world\n");
    return 0;
}
```

ユーザ定義名前付きオブジェクト
ユーザ定義リテラル

プログラムのフレームワーク

C言語

```
#include <stdio.h>
int main(void) {
    printf("Hello, world\n");
    return 0;
}
```

用意されているキーワード(予約語)
特別な意味を持つ記号

プログラムのフレームワーク

C言語

```
#include <stdio.h>
int main(void) {
    printf("Hello, world\n");
    return 0;
}
```

システムに依存するリテラル

プログラムのフレームワーク

C言語

```
#include<stdio.h>
```

```
int<main(void){printf("Hello,
world\n");return<0;}
```

プリプロセッサにより1レコード毎に
処理される
「#キーワード」のフィールドから始まる

プログラムのフレームワーク

C言語

```
#include<stdio.h>
```

```
int<main(void){printf("Hello,
world\n");return<0;}
```

コンパイラで処理される「<」「;」「{」「}」
「/」「*/」で区切られたバイト列
「レコード・フィールド」の概念なし

Pascalの基本構造

```
program プログラム名 (入出力);
begin
    実行すべき処理
end.
```

実行すべき処理が複数あるときは「;」で区切る

FORTRAN77の基本構造

```
PROGRAM プログラム名
実行すべき処理
STOP
END
```

処理は7桁目から72桁目までに記述する

c言語の基本構造

```
#プリプロセッサ命令
int main(引き数) {
    実行すべき処理;
    return 0;
}
```

キーワードと特殊記号

大文字と小文字を区別しない

Pascal

```
program, begin, end, end., writeln, ;,
', (, ) ...
```

FORTRAN77

```
PROGRAM, STOP, END, WRITE, ', (, ), *, ...
```

c言語

```
#, int, ;, (, ), {, }, main, ...
```

大文字と小文字を区別する

注釈 (コメント)

プログラム作成者が原始プログラム(ソースファイル)中に説明や覚書を書き込んだもの
使用や保守に必要な情報を簡潔にまとめておく
スペース(White space)と同じ扱いを受ける

c言語の注釈

「/*」と「*/」で囲まれた文字列

```
#include <stdio.h>
int main(void) {
    /* This is the program to write
    "Hello World" onto the standard
    output. */
    printf("Hello, world\n");
    return 0;
}
```

Pascalの注釈

「{」と「}」で囲まれた文字列

```
program hello(output);
{This program shows strings onto
the screen.}
begin
    writeln('Hello, world')
end.
```

FORTRANの注釈

第一桁に文字(通常「C」が使われることが多い)を記入した一行(レコード)

```
123456789.....
PROGRAM HELLO
C PRINT 'HELLO, WORLD'
WRITE(6, *) 'HELLO, WORLD'
STOP
END
```

入出力

Pascal

標準入出力が言語仕様として存在

FORTRAN

入出力装置を番号で指定

c言語

言語仕様としては存在しない
標準入出力ライブラリを使用

文字列リテラルと出力

Pascal

writeln('文字列')

出力後改行する

write('文字列')

出力後改行しない

FORTRAN

WRITE(6, *) '文字列'

出力後改行する

c言語

printf("文字列\n") 改行記号(\n)まで含めてライブラリ関数にわたす