

今週の講義メモ：Pascal プログラム入門

プログラムって何？

プログラム = コンピュータに処理 (計算) 手順を示すための指示書。

コンピュータの中は'0' と'1' だけ (機械語) ↔ 人間が理解し記述するのは、とても大変

この間を取り持つものがあると、何かと便利

- 人間が理解しやすく、コンピュータの機械語に翻訳しやすい言語 = プログラミング言語
- 人間向けのプログラミング言語を、コンピュータの機械語に翻訳する仕組み = コンパイラ (翻訳機)

Pascal プログラムで使える文字

- 半角英大文字 A~Z, 半角英小文字 a~z, 半角数字 0~9
- 半角の空白と、一部の半角記号 ' () * + , - . / : ; < = > [] { }
- 文字列や注釈の中でのみ、全角文字が使えるが、一部の文字がコンパイルエラーを引き起こすので要注意。

Pascal プログラムの基本構成

0. 文の区切りにはセミコロン ; を書く。語句の区切りには空白または改行を用いる。
1. 必ず program 文で始まる program+プログラム名+引数並び
2. 次に宣言・定義部：ラベル宣言部, 定数宣言部, 型宣言部, 変数宣言部, 手続き関数宣言部
必要な部分だけ, 必ずこの順序で記述する
3. それから実行文部：プログラムの本文 begin で始まって end で終わる
4. 実行文部の終わりの end のすぐ後のピリオド . で、プログラム全体の終わり
5. 空白の間または行の終わりに、注釈 {...} を書くことができる。

名前付けのルール

- プログラム名や変数, 配列などの名前は作り方が決められている。
半角英数字だけが使える。ただし、頭文字は英字に限る。英字の大小は区別されない。
a, X, abc, y01, findingNimo などは OK ; 11b, 2x3, \$zz, M:i-2 などは NG
- Pascal 言語の特別なキーワード = 予約語は使えない
and, array, begin, case, const, div, do, downto, else, end, file, for, function, goto, if, in, label, mod,
nil, not, of, or, packed, procedure, program, record, repeat, set, then, to, type, until, var, while, with
- Pascal 標準で既定されている定数名, 型名, 手続き・関数名も避けた方が良い
定数名: maxint, false, true ファイル名: input, output
型名: integer, real, char, Boolean, text
手続き名: write, writeln, read, readln, reset, rewrite, get, put, page, new, dispose, pack, unpack
関数名: abs, sqr, sqrt, sin, cos, arctan, exp, ln, round, trunc, odd, chr, ord, pred, succ, eof, eoln
- Pascal 標準の既定名ではないが, FreePascal 独自に既定の定数名, 手続き・関数名もあるので注意
PI, assign, close, random, Randomize

データと変数

(1) データの型

- 基本のデータ型:
 - integer ... 16 ビット整数 ± 30,000 くらいの範囲
 - real ... 32 ビット実数 有効数字 7 桁くらい

- char ... 単一文字データ たぶん ASCII コード
- Boolean ... 論理値 true と false だけ
- 基本データ型ではないがよく使う
 - packed array[..] of char ... 文字データの詰込配列 文字列データ
 - text ... テキストファイル

(2) 変数はデータを保存する箱

- 計算に必要な値を保存し、必要なときに値を取り出す。
- 名前付けのルールに従う範囲で、自由に名前をつけてよい。
- var で始まる変数宣言部で、使う変数名とそのデータ型を宣言しておかないといけない。
- 演算などに使う前に、代入文や read 文で値を決めてあげないと使い物にならない。
- 用が済んだら、他の役割の変数として使っていい。

データの入出力

- データをコマンドプロンプトへ表示するには...
 - write(S) S を出力
 - writeln(S) S を出力して、改行
 ただし S は、文字列 (引用符 ' で括る) 'ABC', 'Hello' など、数値 100, 3.14, 3E+8 など、変数名 a, b, x, zz など、式 a+10, sqrt(x) など、あるいは、それらをコンマ、で区切って並べたもの。
- データをキーボードから読み込むには...
 - read(q1, q2, ...) 空白や改行で区切られた入力データを順に、それぞれデータ型に合わせて変換して、変数 q1, q2, ... へ保存

数値型と演算

- 整数型 integer ... 16 ビット表現
 - 整数 = 固定小数点数 右端を小数点の位置と決めた、符号付きの数表現
 - 整数型の定数の例 0, 16, 15000, -127 など
 - maxint 利用可能な整数値の最大値 (既定定数名)
- 実数型 real ... 32 ビット表現 有効数字 7 桁くらい
 - 実数 = 浮動小数点数 有効数字と指数で表現 $0.314 \times (10 \text{ の } +1 \text{ 乗})$
 - 実数型の定数の例 1.0, 0.125, -314.159, 0.271828E-02, 3.0E+8 など
 - 実数値の表現形式: $x.xxxE \pm yyy$ $x.xxx \times 10^{\pm YYY}$ を表す
-3.141590000000000E+002 -314.159 のこと
- 代入 変数 := 式
- 四則演算
 - 加・減・乗算 +, -, *
 - 整数型の除算と剰余 div, mod 注: 整数値になる $7 \text{ div } 3 = 2$
 - 実数型の除算 / 注: 実数値になる $7 / 3 = 2.333\dots$
 - カッコ () 注: 何重になっても丸カッコだけ!
 - ベキ乗の記号はない → 繰り返し計算で実装を
 - 実数型と整数型の演算結果は実数型
- 既定関数
 - 定数 (FreePascal 独自): PI (円周率)
 - 数学関数: abs (絶対値), sqr (二乗), sqrt (平方根), sin (正弦), cos (余弦), arctan (逆正接), exp (指数関数 e), ln (自然対数 \log_e)
 - 実数を整数化: round (四捨五入), trunc (ゼロ側へ切り捨て)

今週のサンプルプログラム

- (1a) [Renshula.pas] 表示のいくつかのパターン

```

program Renshula( input, output );
begin
  writeln( 'Hello, world.' ); { とりあえず普通に }

  writeln( 'Hello, ', 'world.' ); { 2つの文字列をつなぐ }

  write( 'Hello, ' ); { write 文を }
  write( 'world.' ); { 2つに分けて }
  writeln; { 最後に改行だけ }

  writeln( '日本語でこんにちは。' ); { 全角文字も使えるけど要注意 }
end.

```

- (1b) [Renshulb.pas] 数値型変数の宣言と代入

```

program Renshulb( input, output );
var n: integer; { 整数型変数の宣言 }
    x, y: real; { 実数型変数の宣言 }
begin
  n := 100; { 整数型データの代入 }
  writeln( 'Integer n = ', n ); { 文字列と整数型データの表示 }
  x := 3.1416; { 実数型データの代入 }
  writeln( 'Real x = ', x ); { 文字列と実数型データの表示 }
  y := x * n; { 掛け算 }
  writeln( 'x * n = ', y )
end.

```

- (1c) [Renshulc.pas] 数値の読み込み

```

program Renshulc( input, output );
var hankei, enshuu, menseki: real; { 実数型変数の宣言 }
begin
  writeln( '円周率 PI = ', PI ); { PI は円周率の定数 }
  write( '実数値を1つ入力してください:' ); { 入力を促す }
  read( hankei ); { 数値を読んで変数 hankei へ代入 }
  enshuu := 2 * PI * hankei;
  menseki := PI * sqr( hankei ); { sqr は二乗 }
  writeln( '半径 ', hankei, ' の円の ' ); { 文字列に空白を混ぜると読みやすくなる }
  writeln( '周の長さ = ', enshuu );
  writeln( '面積 = ', menseki )
end.

```

(1d) [Renshuld.pas] 整数のデータ型の変数と四則演算, 既定関数

```
program Renshuld( input, output );
var n, m, wa, sa, seki, shou, joyo, k: integer;
begin
  writeln( '整数 n,m の値を入力してください:' );
  read( n, m );
  writeln( 'n = ', n, ', m = ', m );
  wa := n + m;    writeln( '和 = ', wa );
  sa := n - m;    writeln( '差 = ', sa );
  seki := n * m;  writeln( '積 = ', seki );
  shou := n div m; writeln( '商 = ', shou ); { div : 整数の割り算 }
  joyo := n mod m; writeln( '余り = ', joyo ); { mod : 整数の余り算 }
  k := ( ( n + 3 ) div 7 - 1 ) * m; { カッコを使った計算 }
  writeln( 'n に 3 を足して 7 で割って 1 を引いて m を掛けると? ', k );
end.
```

(1e) [Renshule.pas] 実数のデータ型の変数と四則演算

```
program Renshule( input, output );
var a, b, wa, sa, seki, shou, c: real;
begin
  writeln( '実数 a,b の値を入力してください:' );
  read( a, b );
  writeln( 'a = ', a, ', b = ', b );
  wa := a + b;    writeln( '和 = ', wa );
  sa := a - b;    writeln( '差 = ', sa );
  seki := a * b;  writeln( '積 = ', seki );
  shou := a / b;  writeln( '商 = ', shou ); { / : 実数の割り算 }
  c := ( ( a + 3 ) / 7 - 1 ) * b; { カッコを使った計算 }
  writeln( 'a に 3 を足して 7 で割って 1 を引いて b を掛けると? ', c );
end.
```

(1f) [Renshulf.pas] 数学関数の利用

```
program Renshulf( input, output );
var a, b, c, d: real;
begin
  a := 5.23; b := -3.58;
  writeln( 'a = ', a, ', b = ', b );
  c := trunc( a ); d := trunc( b ); { trunc: 小数点以下を切り捨て }
  writeln( 'trunc: ', c, ', ', d ); { 空白文字で区切って数字を読みやすく }
  c := round( a ); d := round( b ); { round: 小数点以下を四捨五入 }
  writeln( 'round: ', c, ', ', d );
  c := abs( a-b ); { abs: 絶対値 }
  writeln( 'abs( a-b ) = ', c );
  c := sqr( a-b ); { sqr: 二乗 }
  writeln( 'sqr( a-b ) = ', c );
  d := sqrt( a+b ); { sqrt: 平方根 }
  writeln( 'sqrt( a+b ) = ', d );
end.
```