

# 情報とコンピュータ

静岡大学工学部

安藤和敏

2004.10.25

# 3章 数値計算と関数の学習

- 数値計算をしてみよう
- 単純な計算
- 関数
- ループの作成と関数の学習
- 最適値の探索
- 情報を配列に格納する
- 総和, 極小, 極大を求める
- プログラミングのパターン

# 数値計算をしてみよう(1)

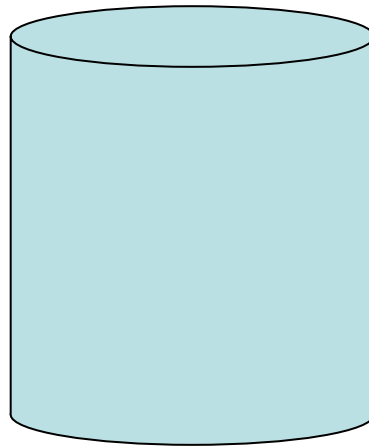
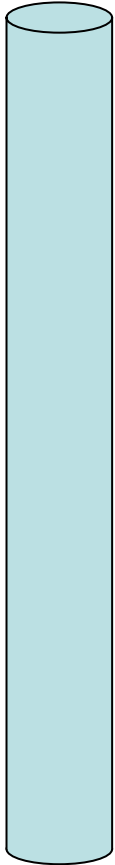
20歳の青年が60歳までに1億円貯めるには、毎月いくら貯金しなければならないか？

# 数値計算をしてみよう(2)

1000m<sup>2</sup>の錫(すず)の板で円柱をつくる  
とき、その円柱の体積が最大になるの  
は正確にどの寸法のとときか？

**最適化**: ある状況におけるパラメータの最適値  
を求める作業.

# 数値計算をしてみよう(2)



# 変数の型

変数は、その変数がどのような値を取ることができるか、つまり、変数の**型**を指定しなければならない。

`var`

`変数名: 変数の型;`

例:

`var`

`position1: string;`

Pascal で用いられる変数の型

- integer: 整数型 (例: 1, 2, 3, ..., -5, -6)
- string: 文字列型 (例: '安藤', 'kazu');
- real: **実数型**

# 実数 (real) 型の変数

実数 (real) 型の変数の値は、仮数部と指数部の二つの部分からなる。

例：

$$177 = 1.77 \times 10^2$$

仮数部

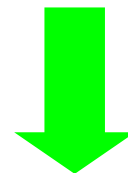
指数部

現在のPCで表現できる実数の絶対値の範囲は、 $10^{-45} \sim 10^{38}$ 、仮数部は7～8桁。

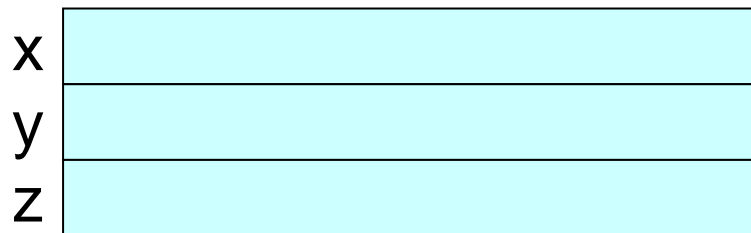
# 実数 (real) 型の変数の宣言

```
var  
  変数名の並び: real;
```

例:  
var  
 x, y, z: real;



メモリ上に, x, y, z という名前の付いた記憶場所が確保される. この記憶場所には, 実数型のデータが記憶される.





# 実数 (real) 型の変数

```
var
  x, y, z: real;
begin
  x := 12.0;
  y := 13.3 + x;
  z := (x + 17.2) * (121 - (y / x));
end.
```

x	$1.2 \times 10^1$
y	$2.53 \times 10^1$
z	$3.471637 \times 10^3$

# 四則演算

- 加算(足し算) (+)
- 減算(引き算) (-)
- 乗算(掛け算) (\*)
- 除算(割り算) (/)

# 円柱の体積の計算

を円周率,  $r$  を円柱の底面の半径,  $h$  を円柱の高さとする, この円柱の体積  $V$  は

$$V = \pi r^2 h$$

これをPascalで書くと,

```
V := 3.14159 * r * r * h;
```

# 円柱の体積の計算のアルゴリズム

1. 半径  $r$  を読み込む .
2. 高さ  $h$  を読み込む .
3.  $V = r^2h$  を計算する .
4. 体積  $V$  を書き出す .

# 円柱の体積の計算のプログラム

```
program CyllinderVolume(input, output);  
var  
  r, h, V: real;  
begin  
  writeln('円柱の半径を入力して下さい。');  
  readLn(r);  
  writeln('円柱の高さを入力してください。');  
  readLn(h);  
  V := 3.14159*r*r*h;  
  writeln('円柱の体積は',V,'です。');  
end.
```

# 円柱の体積の計算のプログラム

```
program CyllinderVolume2(input, output);  
var  
    r, h: real;  
begin  
    writeln('円柱の半径と高さを入力して下さい。');  
    readLn(r,h);  
    writeln('円柱の体積は',3.14159*r*r*h,'です。');  
end.
```

# 金利計算の公式

金利は少数で表されているとする。(12% = 0.12)

新預金残高 = 元の預金残高 + (元の預金算残高 \* 金利)

savings を預金残高, interestrate を金利とすると,  
Pascal の計算は,

```
savings := savings + (savings*interestrate);
```

# 預金残高計算のアルゴリズム

1. 預金の初期残高 `savings` を読み込む .
2. 一定期間の金利 `interestrate` を読み込む .
3. `savings := savings + (savings*interestrate)` を計算する .
4. `savings` を書き出す .



# 預金残高計算のプログラム

```
program FindSavings(input, output);
var
  savings, interestrate: real;
begin
  writeln('預金の初期残高を入力して下さい(単位:万円). ');
  readLn(savings);
  writeln('金利を入力して下さい. ');
  readLn(interestrate);
  savings := savings + (savings*interestrates);
  writeln('一期後の預金残高は , ', savings:6:2, '万円です. ');
end.
```

# 注意 (演算の優先順位)

$x = 2, y = 3, z = 4$  のとき, 次の計算結果はどうなるか?

`result := x + y * z;`

+ が先に実行されると  $\text{result} = 20$  だが,  
\* が先に実行されると  $\text{result} = 14$ .

# 注意 (演算の優先順位)

プログラミング言語では、曖昧な状況では、以下のような優先順位を持っている。

## 優先順位

1. 乗算・除算 (\*, /)
2. 加算・減算 (+, -)

$x = 2, y = 3, z = 4$  のとき,  $\text{result} := x + y * z;$   
を実行した結果は, 14.

# 注意 (演算の優先順位)

優先順位の等しい一連の計算では, 左から右へ計算が行われる.

$x = 6, y = 2, z = 3$  のとき, `result := x / y * z;`  
を実行した結果は,

$$(6/2)*3 = 9 \quad (6/(2*3)=1 \text{ ではない.})$$

括弧を使うことで, 優先順位を強制的に指定できる.

`result := (x + y) * z;` (result は 24).

`result := x + (y*z);` (result は 36);

# 注意 (誤差)

コンピュータのメモリには、有限の桁数の数しか記憶できない。

例えば、 $1/3$ を小数で表すと  $0.33333\dots$  であるが、コンピュータの中では、 $1/3$ の計算結果は  $0.333333$  として記憶される。

# ErrorDemo

```
program ErrorDemo(input, output);  
var  
  data, extra: real;  
begin  
  readLn(data);  
  readLn(extra);  
  data := data + extra;  
  data := data - extra;  
  writeLn(data:30:2);  
end.
```

- $data = 100$
- $Extra = 10000000000000000000000000000000$
- $data = data + extra$   
 $= 10000000000000000000000000000000100$
- $= 100000000000000000000000000000000$
- $data = data - extra$   
 $= 0$

# 関数 (言語による記述)

関数とは、入力を受け取り、出力を返す機能。

例) 人の名前を受け取るとその人の父親の名前を返す機能。(この関数を F と呼ぶ。)



例) 国の名前を入力として、首都名を返す関数、半径と高さを入力としてその円柱の体積を返す関数。



# 関数 (数式による記述)

数学的には、関数名の後の  $()$  の中に入力を書き、出力を  $=$  の右に書く。

例) 父親関数  $F$  の場合は、

$$F(\text{Issac}) = \text{Abraham}$$

$$F(\text{Abel}) = \text{Adam}$$

$$F(\text{小泉孝太郎}) = \text{小泉純一郎}$$

# 数値関数の例

入力データを2倍して返す関数を考え、この関数を  $d$  と呼ぶ。

$$d(3) = 6$$

$$d(17) = 34$$

より一般的には、 $d(x) = 2x$  と書ける。

円柱の半径  $r$  と円柱の高さ  $h$  を受け取って、その体積を返す関数  $v$  は、

$$v(r, h) = r^2h \text{ と書ける。}$$

# 数値関数の表による記述

2倍関数  $d(x) = 2x$  は以下のような表で表すこともできる。

入力(x)	出力(d(x))
1	2
2	4
3	6
4	8
5	10
6	12
7	14
8	16
9	18
10	20

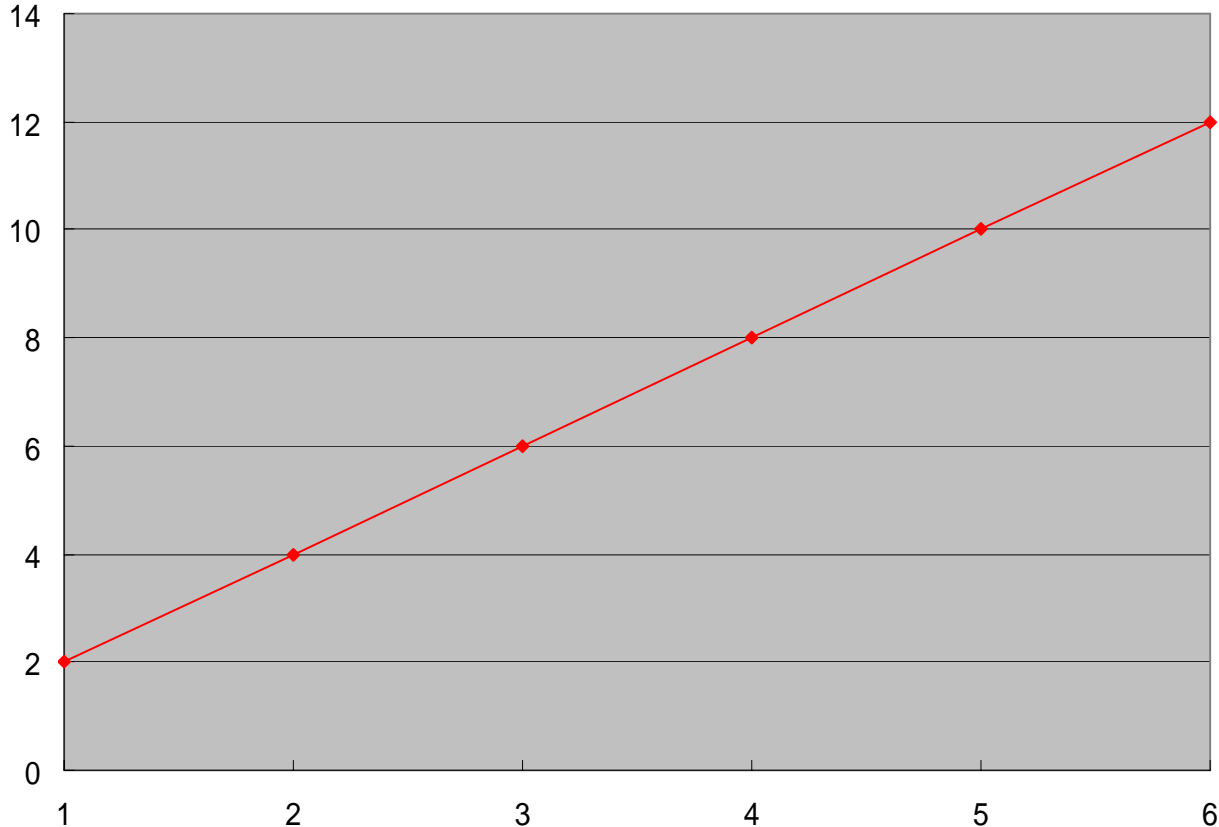
# 数値関数の表による記述

円柱体積関数  $v(r, h) = r^2h$  を表現するためには、2次元の表が必要になる。

		h					
	v(r, h)	1	2	3	4	5	6
	1	3.14	6.28	9.42	12.57	15.71	18.85
	2	12.57	25.13	37.70	50.27	62.83	75.40
	3	28.27	56.55	84.82	113.10	141.37	169.65
r	4	50.27	100.53	150.80	201.06	251.33	301.59
	5	78.54	157.08	235.62	314.16	392.70	471.24
	6	113.10	226.19	339.29	452.39	565.49	678.58
	7	153.94	307.88	461.81	615.75	769.69	923.63

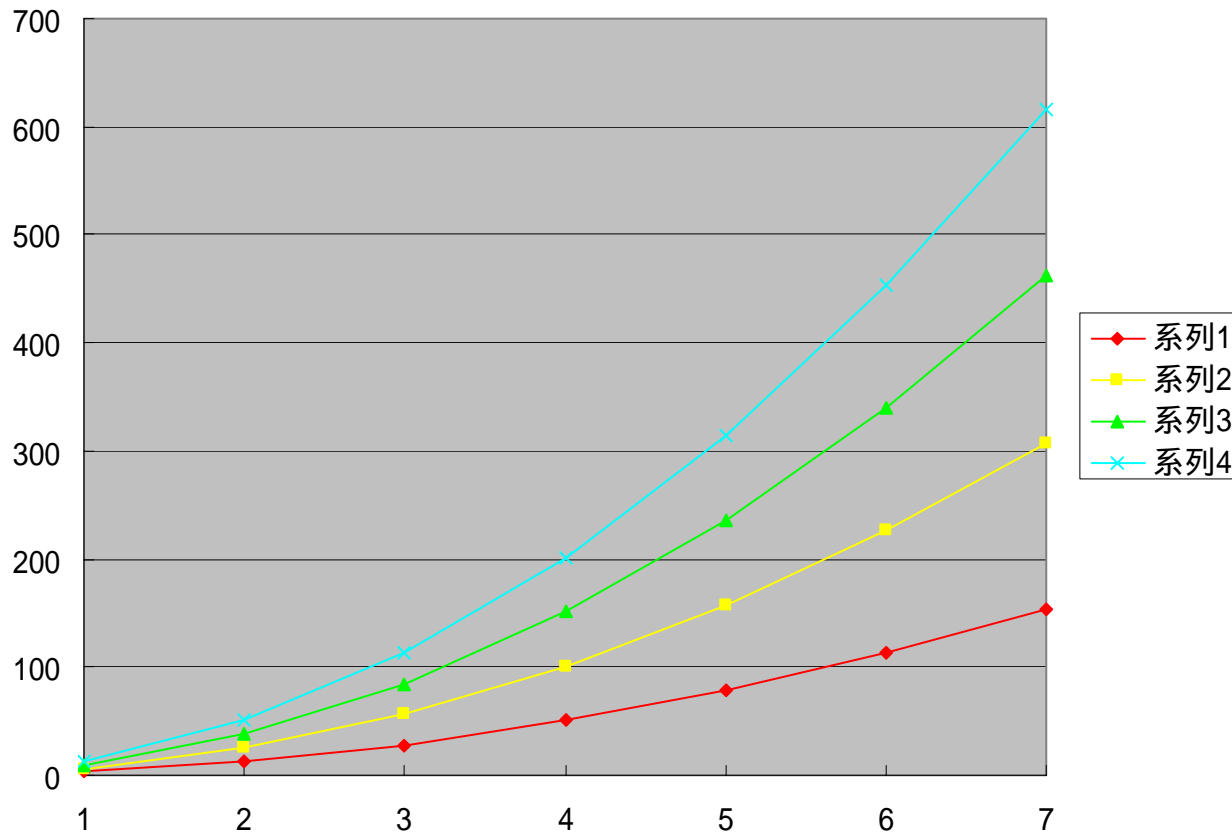
# 数値関数のグラフによる記述

2倍関数  $d(x) = 2x$  をグラフで表現すると以下のようになる。



# 数値関数のグラフによる記述

円柱体積関数  $v(r, h) = r^2h$  をグラフで表現すると以下のようなになる。



# レポートについて

- 課題: テキスト p.55 練習問題 2 「あなたが興味のある事柄に関して決定木を作り, それをプログラムに書け。」
- 提出方法: 決定木について記述した Microsoft Word で書いたファイル, 及び, Pascal プログラムをEメールに添付して送信しなさい.
- Eメールの宛先は, [ic@coconut.sys.eng.shizuoka.ac.jp](mailto:ic@coconut.sys.eng.shizuoka.ac.jp)
- Eメールの送信アドレスは, 大学から配布されたものを用いること.
- Eメールの件名には, 学籍番号と氏名を記載すること.
- 提出期限は, 10月31日(日) 17:00 である.