

# 情報とコンピュータ

静岡大学工学部

安藤和敏

2004.10.18

# 1章 プログラミング入門

## 決定木のコーディング

- ・耳寄りな話
- ・決定木
- ・プログラミングを始めるには
- ・データの読み取りと記憶
- ・決定木のプログラミング

# メモリ

番地(メモリ内の記憶場所)

データ

0	175
1	
2	安藤
3	
4	浜松
5	
6	2004
7	

•  
•  
•

# 変数 (variable)

番地 (メモリ内の記憶場所)

データ

0	175
1	
<b>position1</b>	安藤
3	
4	浜松
5	
6	2004
7	

position1 にデータを記憶したり, position1にあるデータを取り出したりできる.

# Pascal における変数の宣言

```
var  
  変数名: 変数の型;
```

例:  
var  
 position1: string;

# readln (または, readLn)

**readLn(position1);**

1. 入力される文字列を受け取る。(Enterキーで終わる.)
2. その文字列を position1 という名前の場所に入れる.

**例:**

**var**

**position1: string;**

# プログラム例 1 (ReadData)

```
program ReadData(input,output);  
var  
    position1: string;  
begin  
    readLn(position1);  
    writeLn(position1);  
end.
```



# プログラム例 2 (ReadWriteDemo)

```
{ p.41 ReadWriteDemo }  
program ReadWriteDemo(input,output);  
var  
    position1: string;  
begin  
    writeln('Please type in some data.');    readln(position1);  
    writeln('The data have been stored.');    writeln('Next the data will be printed.');    writeln(position1);  
    writeln('This completes the run.');end.
```



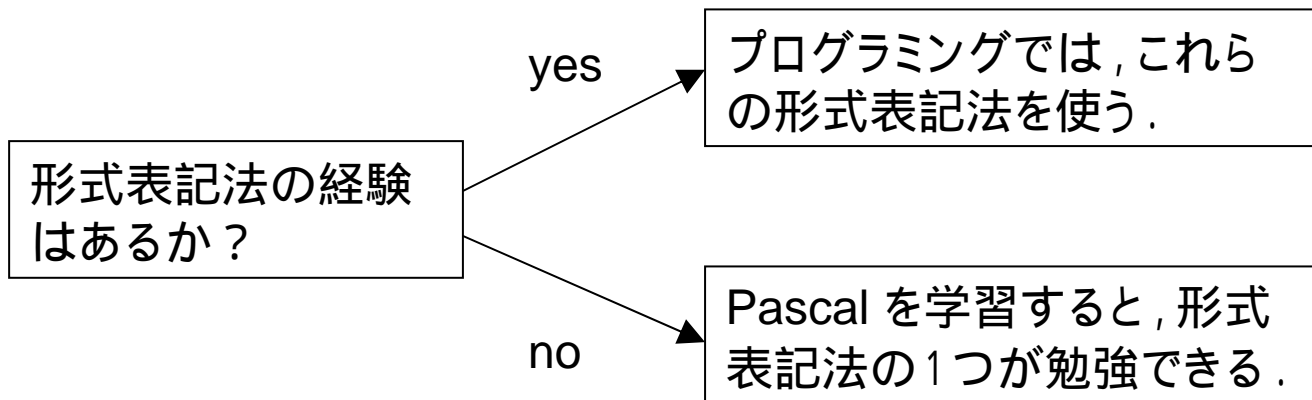
# プログラム例3 (GetTownName)

```
{ p.42 GetNameTown }  
program GetNameTown(input,output);  
var  
    PositionOfName, PositionOfTown: string;  
begin  
    writeln('Hi, tell me your name.');    readLn(PositionOfName);  
    writeln('What town do you live in?');    readLn(PositionOfTown);  
    writeln('Can you tell me something?');    writeln(PositionOfName);  
    writeln('How do you like living in');    writeln(PositionOfTown);  
end.
```

# プログラム例3' (GetTownNameJ)

```
{ p.42 GetNameTownJ }  
program GetNameTownJ(input,output);  
var  
    PositionOfName, PositionOfTown: string;  
begin  
    writeln('やあ, 君の名前を教えてよ. ');  
    readLn(PositionOfName);  
    writeln('どこに住んでるの?');  
    readLn(PositionOfTown);  
    writeln('何か話してくれる? ');  
    writeln(PositionOfName);  
    writeln(PositionOfTown);  
    writeln('ってどんな感じ? ');  
end.
```

# 簡単な決定木



# If-then-else 構文

Pascal の if-then-else文 は例えば, 以下のような書式を取る.

```
If answer = 'yes' then
    PascalコードA
else
    PascalコードB
```

「PascalコードA」と「PascalコードB」は, beginとendで挟まれた文の並びで, これらは, **複合文**と呼ばれる.

# 簡単な決定木のプログラム

```
program FirstTree(input, output);
```

```
var
```

```
    answer : string;
```

```
begin
```

```
    writeln('形式表記法の経験はあるか?');
```

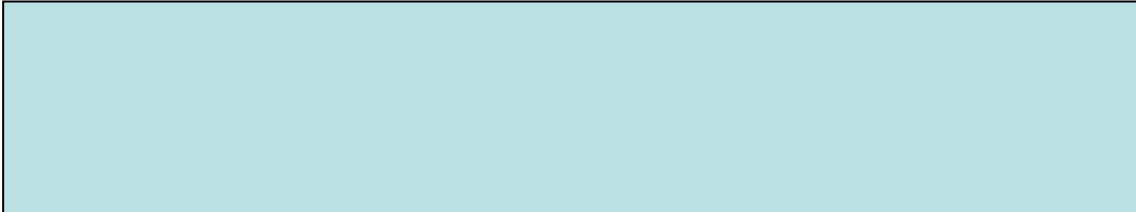
```
    readln(answer);
```

```
    if answer = 'yes' then
```



Pascal コードA

```
    else
```



PascalコードB

```
end.
```

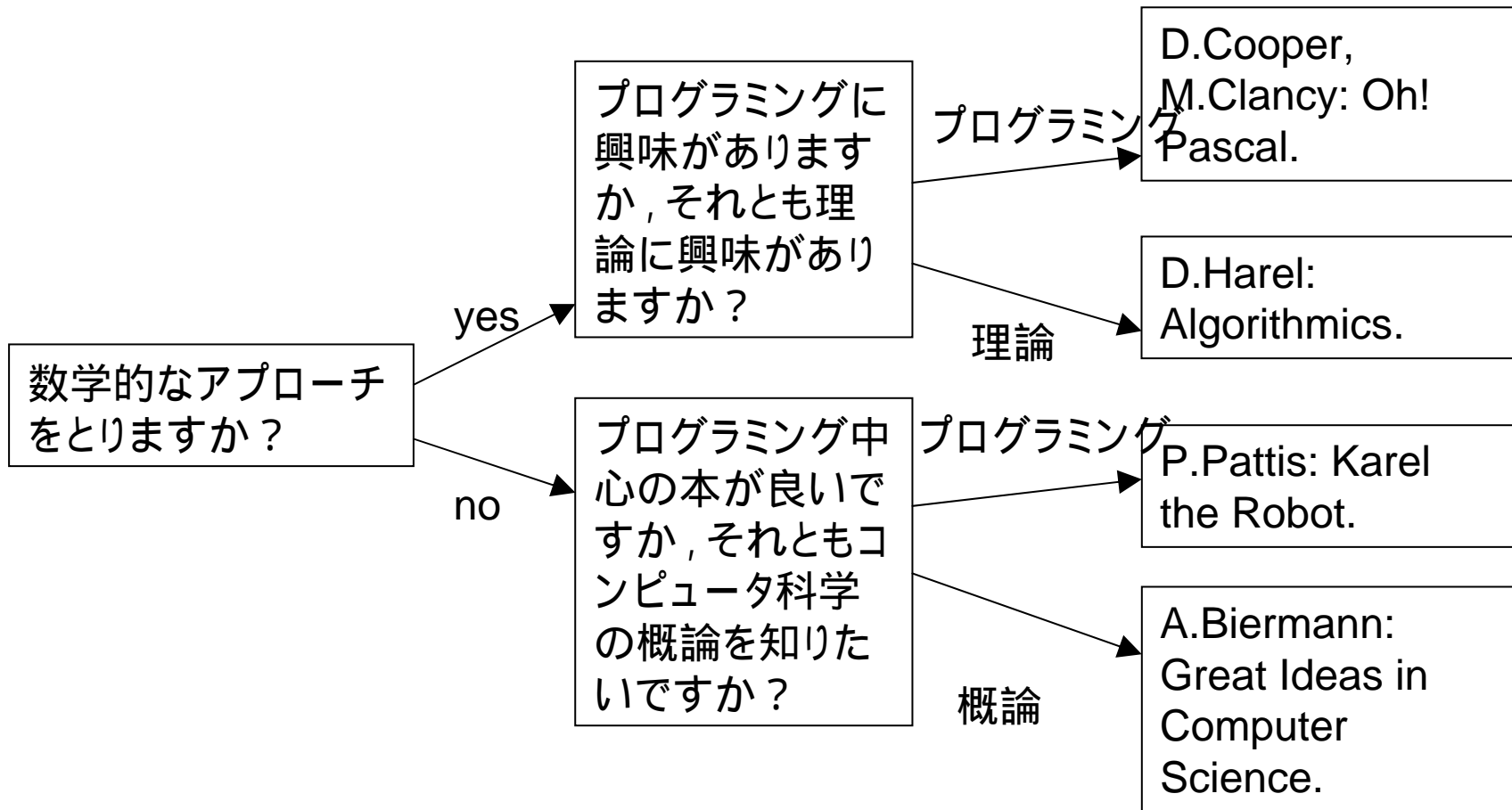
# いくつかの注意

- 質問には, yes か no で答えなければならない.
- 適当なインデント(字下げ)は, プログラムを読みやすくする.(文法的にはどうでもよい.)
- セミコロン(;)のつけ方:
  - ・ヘッダの終わりと宣言文の終わり.
  - ・(複合文中の)各文の終わり.

# セミコロンのつけ方

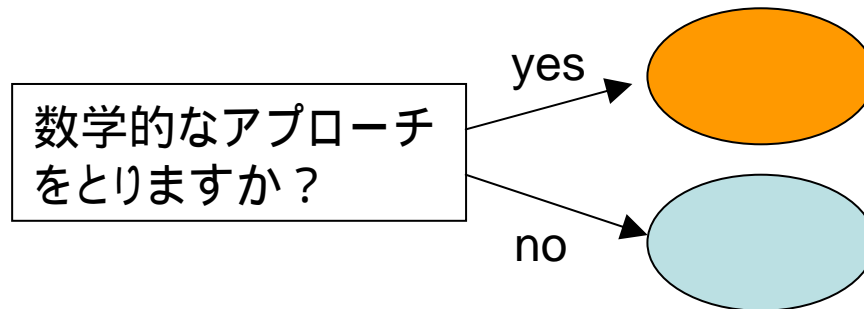
```
program FirstTree(input, output);
var
    answer : string;
{M} begin
{M1} writeLn('形式表記法の経験はあるか? '); {endM1}
{M2} readLn(answer); {endM2}
{M3} if answer='はい' then
    {N} begin
    {N1} writeLn('プログラミングではこれらの形式表記法を');
    {N2} writeLn('実際に使う. ');
    {N} end
else
    begin
    writeLn('Pascalを勉強すると, ');
    writeLn('表記法の1つが学習できる. ');
    end; {endM3}
{M} end.
```

# 図書推薦の決定木





# BookAdviceのコード(1)



```
program BookAdvice(input, output);  
var  
    answer1, answer2, answer3 : string;  
begin  
    writeln('数学的なアプローチをとりますか?');  
    readLn(answer1);  
    if answer1='yes' then  
        yes の分岐を扱うコードをここに入れる  
    else  
        no の分岐を扱うコードをここに入れる  
    end.
```

# BookAdviceのコード(2)

プログラミング

D.Cooper, M.Clancy: Oh! Pascal.

プログラミングに興味がありますか,  
それとも理論に興味がありますか?

理論

D.Harel: Algorithmics.

```
begin
writeLn('プログラミングに興味がありますか?');
writeLn('それとも理論に興味がありますか?');
readLn(answer2);
if answer2='programming' then
  begin
    writeLn('D. Cooper and M. Clancy著');
    writeLn('"Oh, Pascal!"を薦めます。');
  end
else
  begin
    writeLn('D. Harel著');
    writeLn('"Algorithmics"を薦めます。');
  end;
end
```

# BookAdviceのコード(3)

プログラミング中心の本が良いですか、それともコンピュータ科学の概論を知りたいですか？

プログラミング

P.Pattis: Karel the Robot.

概論

A.Biermann: Great Ideas in Computer Science.

```
begin
writeLn('プログラミング中心の本がよいですか,');
writeLn('それともコンピュータ科学の概論を知りたいですか?');
readLn(answer3);
if answer3='programming' then
  begin
    writeLn('R. Pattis著');
    writeLn('"Karel the Robot"がお薦めです. ');
  end
else
  begin
    writeLn('A. Biermann著');
    writeLn('"Great Ideas in Computer Science"がお薦めです. ');
  end;
end;
```

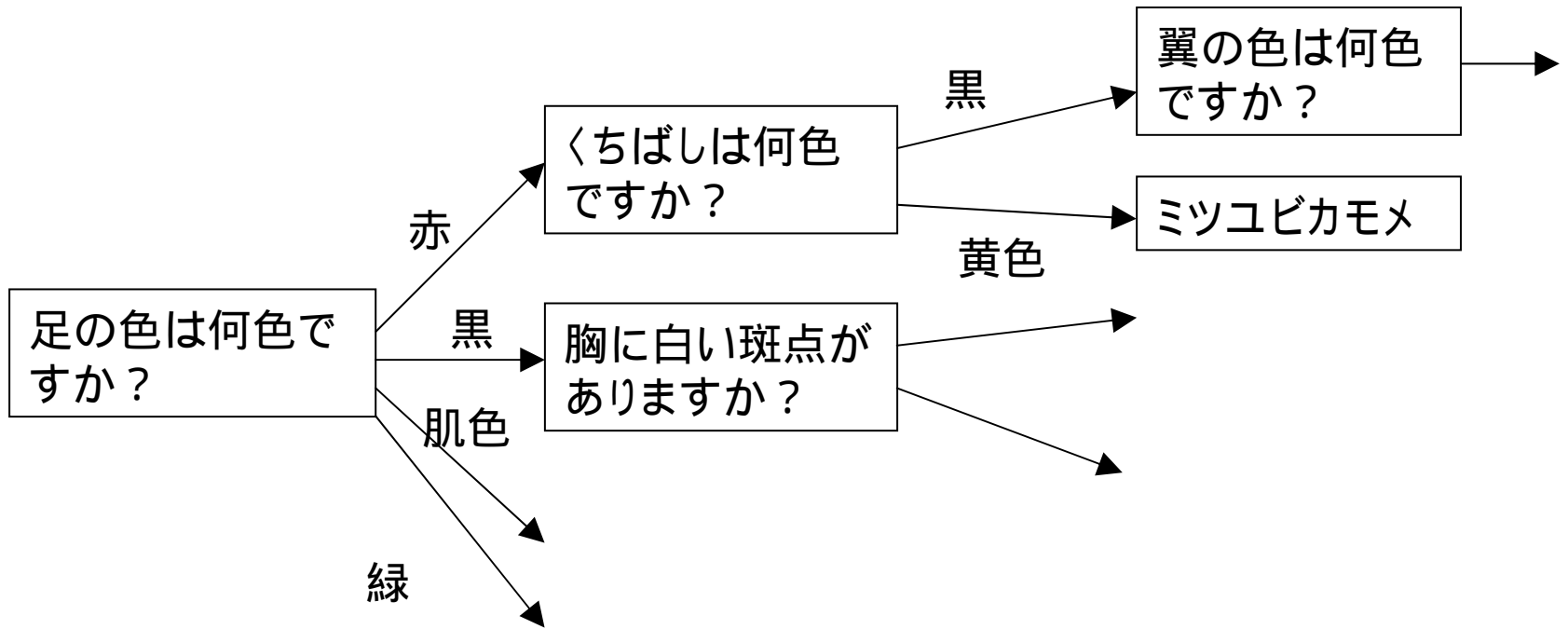
# If-then 構文

Pascal の if-then構文 は、以下のような書式を取る。

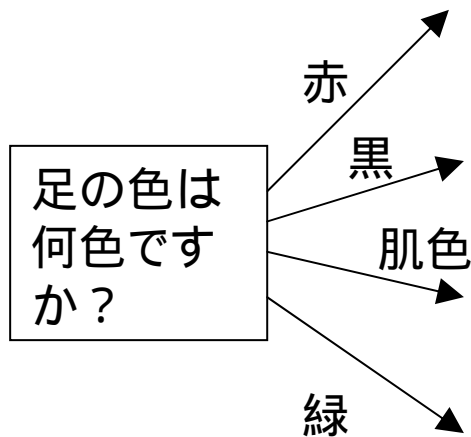
```
If 条件 then  
  複合文;
```

条件が満たされれば、それに続く複合文が実行され、条件が満たされなければ、この複合文は実行されずに、次の文の実行に移る。

# カモメの分類の決定木

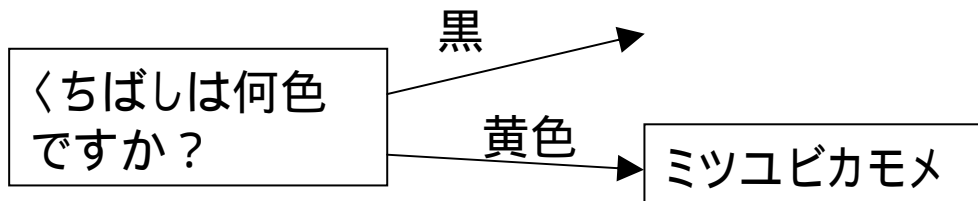


# SeagullClassのコード(1)



```
program SeagullClass(input, output);
var
    answer1: string;
begin
    writeln('足の色は何色ですか? ');
    readln(answer1);
    if answer1='red' then
        「赤」の場合のコード;
    if answer1='black' then
        「黒」の場合のコード;
    if answer1='flesh' then
        「肌色」の場合のコード;
    if answer1='green' then
        「緑」の場合のコード;
end.
```

# SeagullClassのコード(2)



```
begin
writeLn('くちばしは何色ですか ');
readLn(answer2);
If answer2 = 'black' then
  begin
    くちばしが黒い場合のコード
  end;
If answer2 = 'yellow' then
  begin
    writeLn('ミツユビカモメです .');
  end;
end;
```

# 第1章のまとめ

- 決定木のご概念と決定木を用いた情報処理の方法.
- 決定木のプログラミングに必要な概念:
  - 変数
  - Pascalの入出力 (readLn, WriteLn)
  - If-then-else 構文
  - If-then 構文