

「情報処理」1年文I/IIクラス11-12 #7

久野 靖*

1995.11.27

0 本日の目標

予告した通り、今回は出席自由で「プログラミングについてまだ自信がない人のための補講」を行います。まず開始前に「理解度調査」に記入して頂き、続いてまったくの最初から演習を交えて説明をします。最後に再度、同じ調査に記入して頂きます。この調査票+アンケートをもって当日レポート **7A** とします。目標は次の通り。

- プログラムとはだいたいどんなものか納得する。
- 枝分かれまでの PAD 図が書けるようになる。
- その PAD 図を Pascal に直せるようになる。
- できればループについても多少は慣れる。
- (おまけ)xv を使って画面にあるものを何でも取り込み gif ファイルに保存する。

1 理解度調査

まず最初に、次の理解度調査に回答してください (回答用紙は別に配ります)。記入したものは、見えるように机の上に置いてください。

1. 「プログラムとは、計算機が実行する動作を逐一指示するものです。」という説明を読んで「その通りだ」と納得できますか?

納得できる 大よそは… 何のことかわからない その他:_____

2. 「変数とは、値を入れておくための『箱』である。」という説明を読んで「その通りだ」と納得できますか?

*筑波大学大学院経営システム科学専攻

納得できる 大よそは… 何のことかわからない その他:_____

3. 「外界から計算機に情報を与えることを『入力』と呼び、Pascalでは `readln` で表す」という説明を読んで「その通りだ」と納得できますか?

納得できる 大よそは… 何のことかわからない その他:_____

4. 「計算機から外界に情報を示すことを『出力』と呼び、Pascalでは `write` や `writeln` で表す」という説明を読んで「その通りだ」と納得できますか?

納得できる 大よそは… 何のことかわからない その他:_____

5. 「整数 a 、 b を読み込み、その和と差を出力する」という課題について、

自力で PAD 図は書けますか?

書ける 大よそは 書けない

Pascal に直せますか?

そらで OK 大よそは 資料を見れば 直せない

6. 「実数 a 、 b を読み込み、その最大値 (両者とも等しければその値) を出力する」という課題について、

自力で PAD 図は書けますか?

書ける 大よそは 書けない

Pascal に直せますか?

そらで OK 大よそは 資料を見れば 直せない

7. 「正の整数 x を読み込み、 $n^2 \leq x$ を満たす最大の整数 n を出力する」という課題について、

自力で PAD 図は書けますか?

書ける 大よそは 書けない

Pascal に直せますか?

そらで OK 大よそは 資料を見れば 直せない

2 最初から振り返ってみよう

ではさっそく、これまでに学んだプログラミング関係の内容を最初から振り返って見ることにする。

2.1 計算機の構造

そもそも、計算機とは何をする機械か？ それは、図1にあるように、外界から情報を取り込んで加工し、結果の情報を出力するというものだった。ここで「情報」というのはひらたくいえば「ビット列」つまり2進法の0や1の並び、である。ここで、人間が計算機に情報を渡すことを「入力」、計算機から人間に情報を返すことを「出力」と呼ぶ。

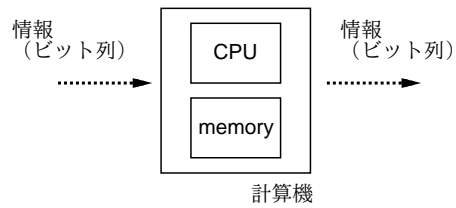


図 1: 計算機とは…

実際には計算機はユーザ(あなた)が使うわけだから、あなたが計算機に(キーボードやマウスを通じて)情報を与え(入力)、計算機があなたに向かって結果の情報を(画面表示などを通じて)返す(出力)。この様子を図2に示す。

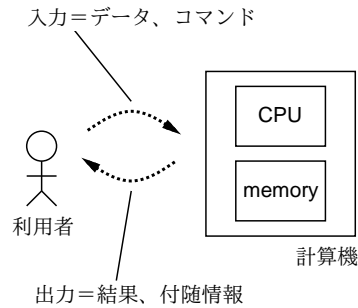


図 2: 計算機との対話

2.2 アルゴリズム (手順) とは

計算機に情報を処理 (ビット列を加工) してもらうには、その「やり方」をあいまさなく、きちんと指定しなければならない。この「やり方」または「手順」のことを計算機の世界では「アルゴリズム」という。

たとえば、「華氏の温度から摂氏の温度を求める」ためには、華氏の温度を f として、

$$c = \frac{5}{9}(f - 32)$$

により値 c を求めればよい。このように、処理の手順をきっちり示したものがアルゴリズムである。

2.3 PAD 図: アルゴリズムの表記法

実際には、計算機に情報を処理 (ビット列を加工) させる場合には、まず人間が計算機に情報を渡し (入力)、計算機の中で計算し、結果を人間に返す (出力) 必要がある。これらの事柄もきちんと書き表すために、いろいろな表記法が使われる。ここでは PAD (Problem Analysis Diagram) 図というのを使う。華氏摂氏変換プログラムの PAD は図 3 のようになる。その要点は次の通り。

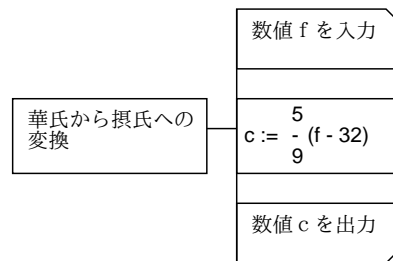


図 3: 華氏摂氏変換の PAD

- 左側の箱は「全体として何をやるか」を示し、右側に引き出して書いてある部分は「それは具体的にはこれこれ」を表す。
- 縦線で結んだ箱の並びは「上から順番に実行」を表す。
- 右上の欠けた箱は入力 (人間から計算機にデータを渡すこと) を表し、右下の欠けた箱は出力 (計算機から人間にデータを渡すこと) を表す。欠けていない箱は計算処理そのものを表す。
- c とか f とか書いてあるのは「処理するためのデータを入れておく箱 (変数)」を表す。入力ではその箱にデータを受け取り、計算処理

では箱からデータを取り出して加工して、結果を別の箱に入れ、出力では箱に入っているデータを画面に表示する。

- 「:=」というのは、左側の値を計算した後、その値を右側の変数にしまいますよ、という意味。

2.4 簡単なアルゴリズム

ではそろそろ演習を。

演習 1 次の手順を PAD 図で書け。

- a. 円の半径 r を読み込み、面積を出力する。
- b. 円錐の底面の半径 r と高さ h を読み込み、体積を出力する。
- c. 整数 n を読み込み、1 の位の数と 10 の位の数を出力する。

なお、最後の c については「切り捨て除算」や「剰余」が欲しい。ここでは Pascal に習って次の書き方を使おう。

- 「 $x \text{ div } y$ 」… x を y で割った整数の商。
- 「 $x \text{ mod } y$ 」… x を y で割った剰余。

たとえば $11 \text{ div } 3$ は 3、 $11 \text{ mod } 3$ は 2 である。

2.5 プログラムと変数

PAD やこの後でてくる Pascal での記述には、変数が重要な役割りを果たす。つまり計算で使う値や途中結果はすべて変数に入れて保持しなければならないから。ここで注意すべきなのは、

`a := b`

という文を実行して変数 b の値を変数 a にコピーしても、 b に入っている値はずっと元のままであり、なくなったりはしないということ。変数の値は一度入れたら、:= で別の値を入れるまではずっとそのまんまである。では先週やったような演習を再度やっておこう。

演習 2 演習 1 の c の PAD について、入力データとして「385」を与えた時のそれぞれの箇所での変数の箱の値を記入せよ (用紙を配布します)。

2.6 プログラム言語 Pascal

Pascal プログラム全体の形は次のようになっている。

```
program 名前 (input, output);  
var 変数定義 …  
begin [文] ; [文] ; … ; [文] end.
```

ここで、「変数定義」は先に述べたような変数の「箱」を用意してもらうためのものである(プログラムで使うすべての変数はまず定義しておかないといけない)。そして、「文」は具体的な個々の「動作」を指定する。名前や「:=」など2文字のくっついた記号の途中以外では、自由に空白や行かえを入れてよい。

2.7 変数定義

変数定義は次のような形をしている。

```
変数名, 変数名, … : 型指定 ;
```

ここで型指定は `integer`(整数) か `real`(実数) のどちらか。変数がいっぱいある場合には1つの変数定義にするより適宜分けた方が見やすい(`var`の後には変数定義がいくつあってもいい)。

2.8 Pascal の「式」

「式」というのは「どのように値を計算するか」の指定。変数と同様、式にも「整数の式」と「実数の式」があり、ちゃんと使い分ける。

- 定数 — その値を表す。例: 10、-3.14 など。
- 変数名 — その変数の箱に入っている値を表す。例: `x`、`max` など。
- 式 + 式 — 加算。
- 式 - 式 — 減算。
- 式 * 式 — 乗算。
- 式 / 式 — 実数の除算 (結果は必ず小数点つきになる)。
- 式 `div` 式 — 整数の (切り捨て) 除算。
- 式 `mod` 式 — 剰余。
- `trunc(e)` — e を切り捨てて整数にする。
- `round(e)` — e を四捨五入して整数にする。

乗除算は加減算より先に実行されるが(つまり $7 + 3 * 4$ は $3 * 4$ が先に計算されて、それから 7 と足される)、怪しいと思ったら適宜「(…)」を使う。式は 1 行に書かないといけないので、分数にしたい場合も適宜/とかっこを使う。たとえば

$$\frac{5}{9}(f - 32)$$

であれば Pascal では

```
(5.0/9.0) * (f - 32.0);
```

のように書く(ほかの書き方でもよいが)。計算が実数(小数点つき)なら定数も小数点つきに。

演習 3 次の数式を Pascal での書き方に直せ。

- a. πr^2
- b. $\frac{4}{3}\pi r^3$
- c. $\frac{(u+l)h}{2}$

このほかに if 文や while 文で使う条件式がある。条件式の値は「はい」「いいえ」のどちらかになる。

- $e=e$ — 等しい
- $e<>e$ — 等しくない
- $e>e$ — より大きい
- $e>=e$ — 以上である
- $e<e$ — より小さい
- $e<=e$ — 以下である
- (条件)and(条件) — 両方が成り立つ
- (条件)or(条件) — 少なくとも一方が成り立つ
- not 条件 — 条件の反転

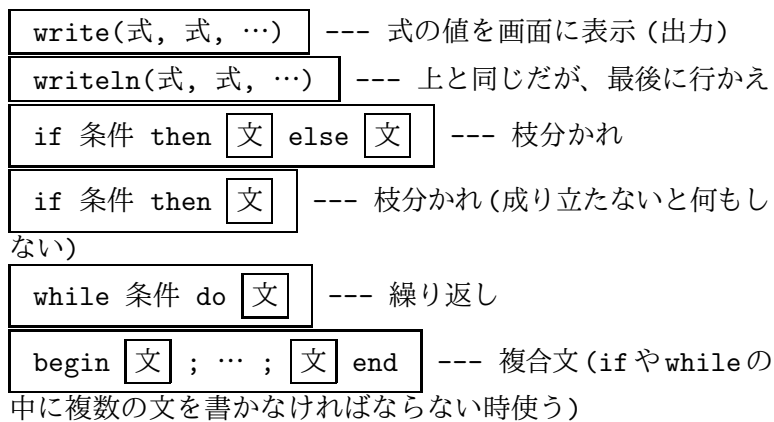
2.9 Pascal の「文」

Pascal の文についてもひとつおりとまとめておこう。

- | |
|----------|
| 変数名 := 式 |
|----------|

 --- 式の値を計算して変数に格納(代入ともいう)
- | |
|----------|
| read(変数) |
|----------|

 --- 変数に値を読み込む(入力)



華氏摂氏変換の PAD を Pascal に直したのもも再掲しておく。

```

program sam1(input, output); ←決まり文句 (名前は変えてよい)
var c, f: real; ←これは変数定義 1 個
begin
  write(' 華氏の温度 = '); ← write 文
  readln(f); ← read 文
  c := (5.0 / 9.0) * (f - 32.0); ←代入 (計算) 文
  writeln(' 摂氏の温度 = ', c:5:2) ← write 文
end.

```

演習 4 演習 1 の 3 つの PAD 図のうちどれか 1 つを Pascal に直して打ち込み、動かせ。

2.10 枝分かれのあるプログラム

実行が 1 直線に進むだけでは単純な計算しかできないので、アルゴリズムに枝分かれを導入する。PAD では枝分かれは右側がぎざぎざの箱を使う (図 4)。

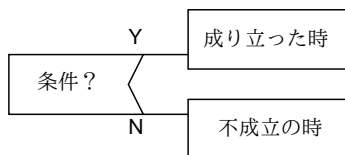


図 4: 枝分かれの PAD

「2 つの実数 a, b を入力し、大きい (小さくない) 順に打ち出す」という例題の PAD は図 5 の通り。

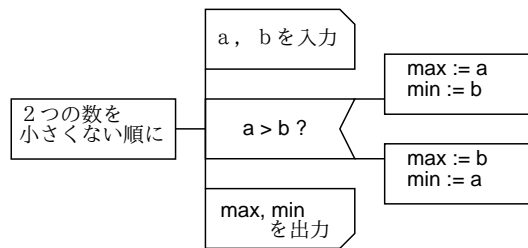


図 5: 2つの数を大小順に打ち出す PAD

これを Pascal にする場合には if 文を使う。if の枝の中に複数の文を入れるためには、それらの文を begin-end で囲む必要があるのに注意。

```

program sam6d1(input, output);
var a, b, max, min: integer;
begin
  write('a = '); readln(a);
  write('b = '); readln(b);
  if a > b then begin max := a; min := b end
  else begin max := b; min := a end;
  writeln('大: ', max:5);
  writeln('小: ', min:5)
end.
  
```

演習 5 次のプログラムの PAD を描け。

- 2つの実数 a 、 b を入力し、0 に近い方 (つまり絶対値の小さい方) を打ち出せ。¹
- 整数 x を入力し、それが奇数か偶数かを表示せよ。²
- 西暦の年号 y を入力し、その年に自分が何歳になるかを表示せよ。ただし自分が生まれていなければ「生まれていない」と表示すること。

演習 6 どれか 1 つ、Pascal に直して動かせ。

なお、基本的な if は 2 方向への分岐だけだが、もっとたくさん枝分かれする場合には

¹ヒント: そのまま比較すると符号が違うときに面倒くさい。だから、 a^2 と b^2 の大きさを比較するとよい。

²ヒント: もちろん、剰余 (さっき説明した) を使ってよい。

```

if 条件 1 then
  動作 1
else if 条件 2 then
  動作 2
else if 条件 3 then
  動作 3
else
  動作 X

```

という if-then-elseif-then の連鎖を使う。これは、「条件 1 を調べて、成り立っていれば動作 1 を、そうでなければ条件 2 を調べて、成り立っていれば動作 2 を、そうでなければ条件 3 を調べて、成り立っていれば動作 3 を、そうでなければ動作 X を実行する」と読める。これなら十分分かりやすい。なお、このパターンでは明らかに条件の数はいくつあってもよいし、また「それ以外」の場合の動作 X が「何もしない」なら最後の else 以下も省略してよい。PAD では図 6 のように書く。



図 6: 複数選択の PAD の箱

2.11 ループのあるプログラム

これは先週やったばかりだし次回もあるので、まあそれなりに。「ループ」というのはその部分を「繰り返して実行」するような部分をいう。PAD ではループは縦線入りの箱で繰り返し条件を表し、その右側に繰り返し実行される部分 (ループ本体) を書く (図 7)。

たとえば、「正の整数 n を正の整数 m で割った時の商と余り」を考える。これは結局、「 n から m が何回引けるか」「引けなくなった時の半端はいくつか」を求めるわけだから、実際に繰り返し引いてみればよい³。

PAD を 8 に示す。先週もくどく説明した通り、ループの中で「段々条件が成り立ちにくい方向に変数の値を変化させている」ことに注意。

³人間の場合、くりかえし引くなんて面倒だ! と思ってしまうが、計算機は面倒だなどと思ったりしないのでプログラムの場合は素直に繰り返し引いても全然構わない。ただし除算命令を使った方が速いので普段はやらないが、これは練習だから。

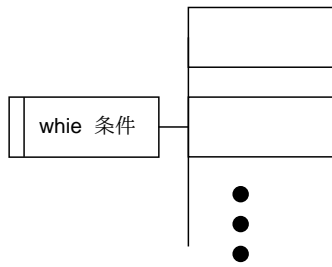


図 7: 繰り返しを表す PAD

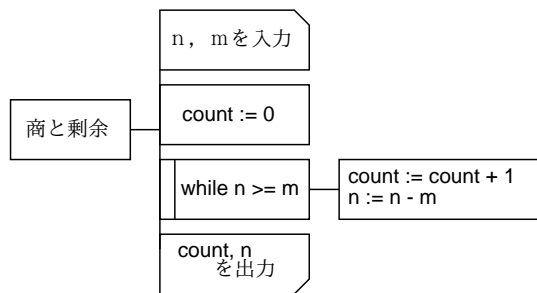


図 8: 商と剰余の PAD

Pascal を次に示す。ループ本体が2つ以上の文から成る時には begin-end で囲む必要があるのに注意。

```

program sam7a(input, output);
var m, n, count: integer;
begin
  write('n = '); readln(n);
  write('m = '); readln(m);
  count := 0;
  while n >= m do begin
    count := count + 1;
    n := n - m
  end;
  writeln('sho = ', count:3, ' amari = ', n:2)
end.

```

演習 7 次のプログラムの PAD を描け。

- a. 実数 n を入力し、金利 $n\%$ (つまり1年間経つと元金の $1 + \frac{n}{100}$ 倍に増えているということ) で、元利合計が元金の倍を越える

のは何年後かを出力せよ。⁴

- b. 1、2、4、8、…と数を倍々にしていったとき、はじめて6桁になったときの数はいくつか。また、それは1を何回「倍にした」値か。⁵

演習 8 どれか1つ、Pascal に直して動かせ。

3 おまけ: xv による画面の取り込み

プログラムばかりでは息がつかまると思うので、おまけとして「画面の任意の部分を取り込んで GIF 形式で保存する」というのを教えて差し上げよう。前回「idraw で描いたものを GIF にするにはどうしたらいい?」という質問を頂いたのだが、それはつまり idraw で絵を描いている画面を取り込んでしまえばよいのだ! なお、GIF ファイルは xpaint で直せるので、取り込んだ画面にさまざまな加工を施すことも自由である。

注意! ただし、Web ページをはじめ多くの「既にある絵」には著作権や肖像権が付随している。そのような絵を勝手に取り込んだりして使うのは違法行為である! 十分注意すること。

さて、まず取り込みのためには xv を引数指定なしで「xv &」で起動する。すると標準の絵が表示されるので、その上でマウス右ボタンをクリックして制御パネル (図 9) を出す。

取り込みを行うには、制御パネルの「Grab」というボタンをクリックすると、取り込みパネル (図 10) が現れるので、その「Grab」ボタンを押した後で (ちょうど hardcopy のように) 取り込みたい窓の上でマウスボタンをクリックする。すると、xv の絵の窓が取り込んだ窓に変わる。画面全体を取り込みたい場合には「背景」の部分でクリックすればよい。

このままでよい場合もあるが、たいていは取り込んだ画面全体ではなくその一部だけを使用したい。そこで、xv の絵の窓の上で「使用したい矩形領域」を指定する。具体的には、矩形領域の左上でマウス左ボタンを押し、押したままでマウスを右下の位置まで移動してボタンを放す。すると絵の上に矩形が表示される。場所や大きさがいまいちなら、OK になるまで繰り返してよい。OK になったら制御パネルの「Crop」ボタンを押すと矩形の外部が「切り落とされて」小さいイメージになる。

満足いく状態になったら、「Save」ボタンを押して保存パネルを出す。保存するファイル形式として「Format:」の選択バーをつついて GIF を撰

⁴ヒント: たとえば元金を 100 万円として、それに上の式を繰り返し掛けていって、200 万円超になるまで何回掛けたかを数える。

⁵ヒント: 6 桁ということは 100000 より小さい間は「まだだ」ということですね。ついでに、このプログラムは入力が必要!

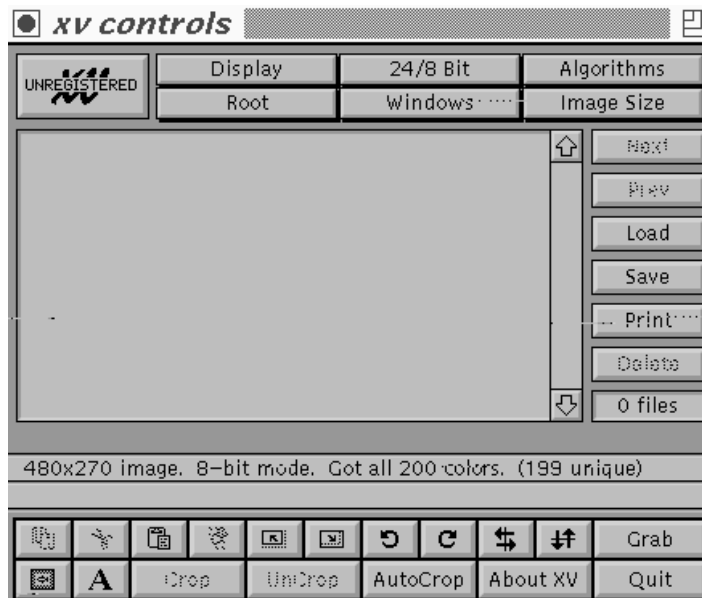


図 9: xv の制御パネル

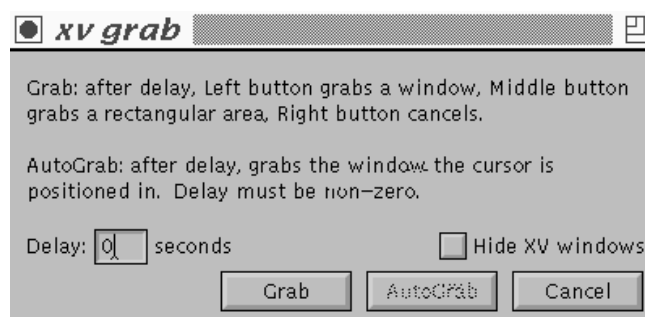


図 10: xv の取り込みパネル

び、「Save file:」の欄にファイル名(最後が「.gif」)を打ち込んで「OK」を押すと保存される。簡単でしょう？

なお、GIF ファイルは当然、xpaint で編集できる。それには、xpaint を起動した後パレット窓の「File」メニューで「Open」を選び、ファイルパネルが出たら「GIF Format」を選択してからファイル名を打ち込んで「OK」をつつけばよい。

演習 Mosaic で当たり障りのなさそうな画面を表示させ、xv で取り込んで必要そうな部分だけを切り出してみよ。それを GIF ファイルとして保存した後、xpaint で読み込んで落書きしてみよ。

A 本日の課題 **7A**

最後に、最初と同一の質問に(今度は裏側に)回答してください。その下にアンケートも記入すること。アンケートは次の通り。

- Q1. 今日の補講でちょっとは理解が進みましたか。進まないとすれば(あるいは進み方がいまいちだとすれば)どこに問題がありますか。
- Q2. 初めての人がプログラミングを学ぶときに、特に難しいことは何だと思えますか。
- Q3. その他、感想、要望、質問があればどうぞ。

B 次回までの課題

予告通り、本日は次回までの課題は出しません。先週提示した **6B** の期限が次回までとなっています。