

「情報処理」1年文I/IIクラス9-10 #5

久野 靖*

1994.11.21

駒場祭おつかれ様でした。それで、WWWでも掲示した通りレポート4Bの提出は来週11/28の授業開始までとします。本日の内容はお約束通りお絵描きと、その前に「まだプログラムが書けない」人のための特訓、およびこれまたお約束の「;はどこにつけるかの解説」ということにさせていただきます。

1 プログラムとは結局何か

プログラムはむずかしい、よく分らない、という人は依然としているけれど、それらの人が「なぜわからない」かを究明するのは結構むずかしい。(ここでインタビューもしてみたいですね!)とまれ、同じことを色々な角度から説明するという方法を取ってみよう。

まず、プログラムというのは計算機が実行する「ビット列の加工」のやり方を指示したものだ、というのは前からさんざんやりましたね?そして、計算機というはおもにCPU(加工を行うところ)とメモリ(ビット列を保持しておくところ)から成るということも。これに対応して、プログラムの「材料」というのは結局次の2つしかない。

- (a) ビット列を入れる「箱」(変数)…メモリに対応。
- (b) ビット列を加工する動作の指示…CPUの動作に対応。

これらはそれぞれ、Pascalプログラムの場合「変数宣言」(var部)と「文」に対応していた。しかし「文」は沢山あって目立つけれど、変数の方は最初にちよつとあるだけだからあまり重要な気がしない。ところが、プログラムの各「文」を実行している時それぞれの変数がどんな値を保持しているか理解しなければ、プログラムは書けない。

その辺を理解して頂くために、変数の「箱」を描いた絵を使ってみよう。プログラムそのものは枝分れの無い時間差問題のプログラムを例に用いる(図1)。

3時22分40秒と5時18分25秒を入力とした場合に、各段階ごとに変数の「箱」の値が変化していく様子を図2に示した。このように、PADなりPascalなりで記述された「動作」が実行されるごとに、変数の値がどう変化していくかをイメージすることがプログラム理解のポイントとなる。もしよく分らなくなったら、自分でこれと同じような図を描いて考えることはよい方針である。

演習1 図3のPAD図およびそれを変換してできた以下のPascalプログラムについて、適当な入力を仮定して変数の箱の値の変化の様子を描き、これは要するに「何をしている」のか述べよ。

```
program sam6a(input, output);  
var a, b, c: integer;
```

*筑波大学大学院経営システム科学専攻

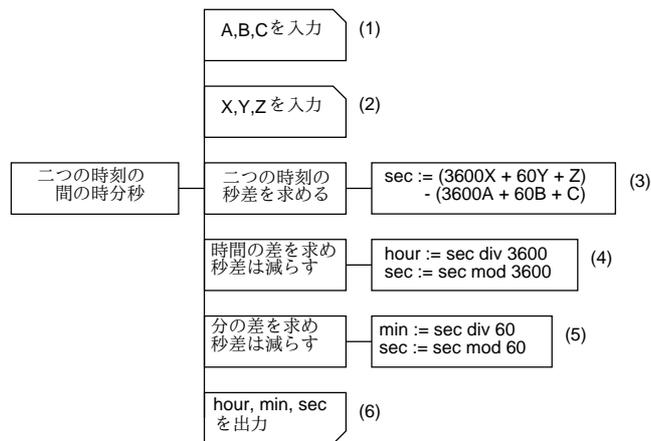


図 1: 枝分れのない時間差問題の PAD 図

(1)	A	3	X		sec	
	B	22	Y		hour	
	C	40	Z		min	
(2)	A	3	X	5	sec	
	B	22	Y	18	hour	
	C	40	Z	25	min	
(3)	A	3	X	5	sec	6945
	B	22	Y	18	hour	
	C	40	Z	25	min	
(4')	A	3	X	5	sec	6945
	B	22	Y	18	hour	1
	C	40	Z	25	min	
(4)	A	3	X	5	sec	3345
	B	22	Y	18	hour	1
	C	40	Z	25	min	
(5')	A	3	X	5	sec	3345
	B	22	Y	18	hour	1
	C	40	Z	25	min	55
(5)	A	3	X	5	sec	45
	B	22	Y	18	hour	1
	C	40	Z	25	min	55

図 2: 図 1 のプログラムにおける変数の値の変化

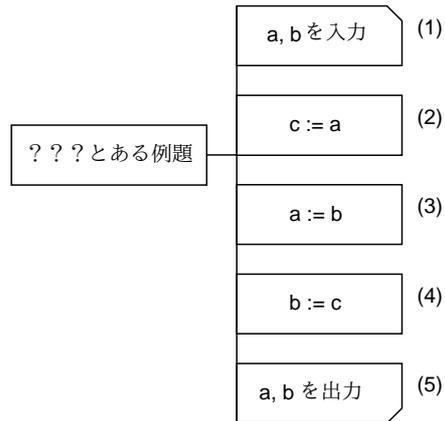


図 3: とある例題プログラムの PAD 図

```

begin
  write('a = '); readln(a);
  write('b = '); readln(b);
  c := a; a := b; b := c;
  writeln('a = ', a:2, ' b = ', b:2)
end.

```

2 どこに「;」を入れるべきか?

これまで Pascal のプログラムを書いて頂いて、いくらか書けるようになった人を一番悩ませるのは「どこに;を入れるべきで、どこには入れなくてもいいか」という問題のようです。ここではちょっと詳しくその説明を試してみませう。

まず、資料#2でも説明したように、Pascal では一番外側の `begin` から `end.` までのところは次のようになっている。

```
begin [文] ; [文] ; ... ; [文] end.
```

つまり、「;」は「文と文の区切りに入れる」ということである。それをはっきりさせるように、先の sam6a のプログラムについて、それぞれの文を四角で囲んだものを図 4 に示した。確かに文と文の切れ目に「;」が入っている。

```

program sam6a(input, output);
var a, b, c: integer;
begin
  write('a = '); readln(a);
  write('b = '); readln(b);
  c := a; a := b; b := c;
  writeln('a = ', a:2, ' b = ', b:2)
end.

```

図 4: とある例題プログラムの文を箱で囲む

では、ifによる枝分かれがあったらどうだろう？ if自体も1つの「文」であり、なおかつこの中に複数の「文」が含まれている、ということが問題をやっかいにしている。つまり、こうなっているわけだ。

```
if 条件 then 文 else 文
```

演習 2 これをヒントに、前回出てきた次のプログラムについて、手で書き写すかプリンタでプリントアウトした上で、それぞれの文を四角で囲め。if文は全体として文であり、なおかつ中に文が入っていることに注意。

```
program sam4c1(input, output);
var a, b, c: real;
begin
  write('a = '); readln(a);
  write('b = '); readln(b);
  write('c = '); readln(c);
  if a > b then
    if a > c then
      writeln('larger value = ', a:8:4)
    else
      writeln('larger value = ', c:8:4)
    else
      if b > c then
        writeln('larger value = ', b:8:4)
      else
        writeln('larger value = ', c:8:4)
  end.
```

ところで、if-then-else if-then の連鎖は本当は多数の if文が組み合わさっているのだが、ここでは簡単のため図5左のように各動作として文を複数含んだ1個の文であると考えておこう。そうしてみると、符号の問題のプログラムについて文を箱で囲んだものは図5右のようになり、確かに「文と文の切れ目」にのみ「;」があることが分かる。

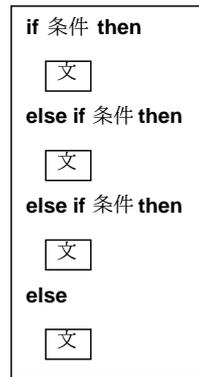
最後の難関として、「ifの枝に複数の文を書きたい時にはbeginとendで囲む」という問題を考える。beginとendで囲んだものは「複合文」といい、それ全体として1つの文だが、中に複数の文を含んでいる。つまり

```
begin 文 ; 文 ; ... ; 文 end
```

のようになる。

演習 3 次の例題プログラムについて、文を箱で囲み、確かに文と文の切れ目にのみ「;」があることを確認せよ。

```
program dam6b(input, output);
var a, b, c: integer;
begin
  write('a = '); readln(a);
  write('b = '); readln(b);
```



```

program sam4b(input, output);
var a, b, c: integer;
begin
  write('x = '); readln(x);
  if x < 0 then
    sign := -1
  else if x > 0 then
    sign := 1
  else
    sign := 0;
  writeln('a = ', a:2, ' b = ', b:2)
end.

```

図 5: 符合問題プログラムの文を箱で囲む

```

if a < b then begin
  c := a; a := b; b := c
end;
writeln('a = ', a:2, ' b = ', b:2)
end.

```

というわけで、これからも「;」の入れ方が分からなくなったら文を箱で囲んで見ることをお勧めする。

3 お絵描きについて

さて、前回お約束してお絵描きのお話に行きましょう。まず最初に説明しなければならないのは、計算機でお絵描きをする時には「ペイント系」と「ドロー系」の2つの系統があって、これらは全然違うということである。この名前の由来は、Apple Computer という会社が Macintosh を最初に世に出した時一緒に出した2つのお絵描きソフト MacPaint と MacDraw から来ている。

まずドロー系から説明しよう。ドロー系のお絵描きソフトというのは、いわば「無限に伸縮性のある針金でできた、直線や長方形や円や楕円の形をした枠」を紙の上に置いていくものと思えば良い(なお、それに無限に伸縮性のあるスクリーンを貼ることもできる)。それで何が嬉しいかと言うと、紙の上に置いてみてちょっと位置が悪いと思ったら(何しろ針金の枠だから)ちょっとずらして見たりでき、いくらでも直しが効くということである。

3.1 idraw 入門

ものは試し、ドロー系ソフトをとにかく使ってみよう。ここでは idraw という、上述の MacDraw の「そっくりさん」のフリーソフト(無料で入手できるもの)を使う。無料だから多少使いづらくても文句を言わないように。idraw を起動するには、Kterm の窓で

```
% idraw &
```

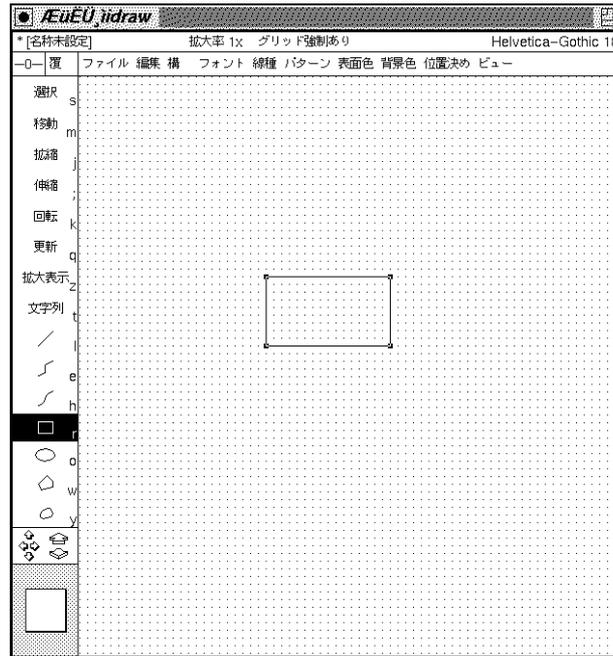


図 6: idraw の窓の様子

という。しばらく (1 分くらい!) 待つと図 6 のような新しい窓が現れる。

ここで、窓の左側に「選択」などと並んでいるのは「パレット」と呼び、各種操作や描く図形の種類を切替えるのに使う。パレットの上にマウスカーソルを持っていき、マウスボタンをどれでもいいから押してパレットの押した項目が反転することを確認してみよ。

一方、窓の上の方に「ファイル」などと並んでいるのは「プルダウンメニュー」と呼び、パレットにないような各種の操作を起動するのに使う。メニューの使い方は既によくご存じの背景メニューに似ているが、少しだけ違う。まず「ファイル」などと書かれた文字の上でボタンを押し下げ、そのまま保持する。するとメニューが現れるので、そうしたらボタンを押したままマウスを移動して選択したい項目を反転させ、その状態でボタンを離すとその項目が起動される。気が変わってやめたい場合にはマウスカーソルをメニューの外に出して (どれも反転してない状態で) ボタンを離せば何も起こらない。さっそく、各メニューにどんな項目があるかを一通り見物してみよ。

さて、ドロー系でお絵描きを行なう時美しくするコツは (あくまでも私の考えだが)、きちんと方眼紙を用いて、その目の上に合わせて線を引くことである。皆様の idraw の窓には方眼が現れていませんか? そこで、「ビュー」メニューの中の「グリッド表示/非表示」という項目を選んで目盛りを表示させる。ついで、同じメニューの「グリッド強制あり/なし」を選んで窓の一番上に「グリッド強制あり」と表示された状態にする。これで、あなたが描く直線や四角はすべて目盛りの上に乗るように idraw が面倒を見てくれる。

3.2 idraw による図形描き

ではいよいよ絵 (ドロー系の場合には「図」という方が正確かも知れない) を描こう。以下では特に指定しない限り、マウスボタンはすべて左ボタンを使うこと!

まず、パレットで四角を選び、その後窓の中で左ボタンを押し、押したままマウスを移動する。すると、移動につれて点線で四角が表示される。(その四角が必ず方眼の目盛りに揃っていることに注意。) 適当なところでボタンを離すと、その位置に四角が置かれる。さっそく、いくつか四角を描いてみよ。また、丸や直線も同様のやり方で描ける。描いてみよ。

残りの図形は、やり方がちよつと違う。例えば折れ線のところを選んでから、窓の中で左ボタンをクリックする(押して離す)。そのままマウスを移動すると、さっきと同様に直線が描かれ始める。次に適当な位置で再度クリックすると、そこまでの直線は固定され、そこから新しい直線が描かれ始める。これだといつまでたっても終わらないが、終わりにしたい時はマウスの中ボタンを押せばそこまで折れ線がおしまいになる。さっそく折れ線を描いてみよ。また、曲線(折れ線をなめらかにしたようなもの)、多角形、閉曲線も(終わりにすると最初の点と結ばれて閉じた図形になるだけで)ほぼ同様である。描いてみよ。

最後に、文字列が残った。文字列については、まずパレットで文字列を選び、次に窓の中で左ボタンをクリックすると縦棒が現れる。その状態でキーボードから打ち込むと文字が入る。漢字を入れたい場合には、Muleと同様「Control-\\」を打てばローマ字モードになり、あとは同様に使えるが、ただし!ローマ字モードから抜ける方法はなぜか「Shift-Space」である。文字列を入れ終わったら、次に入れたい場所をクリックするか、パレットで別の図形を選べばそれで固定される。

3.3 選択と移動

ところで、常に一番最後に描いたものの周囲に「小さい四角」が表示されていることに気がつきました?これは「ハンドル」といい、その図形が「選択されている」ことを意味する。

実は、マウスの右ボタンで図形の上をクリックすると、その図形が選択図形になる(やってみよ)。また、同時に複数の図形を選択することもできる。そのためには、次のどちらかを行なう。

1. 四角を描く要領で、選択したい図形を囲む領域の右上隅で右ボタンを押し、そのままマウスを移動して(長方形が表示される)選択したい図形を囲む四角ができたところでボタンを離す。
2. シフトボタンを押しながら右ボタンで図形の上をクリックすると、その図形がこれまでの選択に追加される。選択されているものの上でシフトボタンを押しながらクリックすると、選択から外される。

だいたい1をやった後2で微調整をする。さっそくやってみよ。

では、図形を選択は何の役に立つのか?まず、図形を移動するとき役に立つ。マウスの中ボタンは図形の移動機能に対応していて、選択された図形群のどれかの上で中ボタンを押し、そのままの状態でもマウスを移動すると図形群がマウスに呼応して移動する。(やってみよ。)実は、移動したい図形が1個だけの場合にはいちいちまず右ボタンで選択しなくても、いきなり中ボタンを押すことで選択しかつ移動開始できる。

3.4 様々な属性調整

さて、これまでのところ図形はいろいろできたが、線の太さも文字の形も1種類なのであまりすばらしくない。ところが!ここからがドロー系ソフトのすばらしいところで、後から色々属性を変化させられる。まず文字列をどれか1個右ボタンで選択し、その状態で「フォント」メニューを下ろしてどれか適当なフォントを選んでみよ。すると、選択した文字列はそのフォントに変わってしまう。

次に、直線または折れ線または曲線を右ボタンで選択し、「線種」メニューで線の種類を選ぶと線の形が変わる。下4項目は矢印のつけ方(なし、右矢印、左矢印、両矢印)で、線の太さなどは独立に変えられる。実は四角や丸や多角形や閉曲線などでも線の太さは変えられるが、矢印については意味を持たない。

次に、四角形や丸や多角形や閉曲線を選び、パターンメニューから適当なパターンを選んでみよ。実はこれまでは一番上の「覆」で、「透明」を意味していたが、それ以外を選ぶと図形は不透

明になり、(ちょうど窓がそうだったように)重なったところは上の図形が下の図形を隠すようになる。重なり順を変えたければ(窓とは違って移動しただけではだめで)、選択した上で「構」メニューの「最前面へ」「最後面へ」を使う。

次は色で、パターンを縞々などにしたままの状態ですべて「表面色」「背景色」をそれぞれ選んでみる。パターンの「模様」と「地」の色がそれぞれ選べるのがわかる。なお線や文字は「表面色」だけが意味を持つ。

3.5 その他の操作

パレットの残りの操作はいちいち説明しない。必要に応じて試してみればわかる(分からなければ質問メールをください)。また、細かいところを拡大して見たり全体像のため小さくして見たりしたい場合は窓の左下の矢印群や四角を使う。これもやってみればわかる。最初の見え方に戻したい場合には「ビュー」メニューの「標準サイズ表示」と「ページの中央」を使えばだいたい大丈夫。詳しく知りたい場合にはマニュアルを読む(Ktermの窓で「man idraw」を実行すると画面に表示される)。

そして最後に!できた絵を保存しなければならない。そのためには、「ファイル」メニューの「保存」と「保存(名前指定)」を使う。これらはちょうどMuleでの $\sim X^S$ と $\sim X^W$ に対応している。だから最初に保存する時は「保存(名前指定)」を使う。すると、新しい窓が現れてファイル名が入力できるようになるので、そこで「test.ps」のようにファイル名を入れて[RET]を打つ。

なお、名前の最後が「.ps」で終わるのはPostScript(PS)形式として扱えるファイルであることを示している。(厳密には、idrawが読み書きするのはidraw形式のファイルなのだが、idraw形式はPS形式としても使うことができるので便宜上「.ps」をつけている。)書き出した後idrawを終了するには「ファイル」メニューの「終了」を選ぶ。

一度書いた絵を編集するにはidrawを動かした状態で「ファイル」メニューの「開く」を使う。するとさっきと同様の窓ができてファイル名が打ち込めるので、そこで打ち込んだ後[RET]を打つ。絵が現れたら適宜修正し、最後に「保存」で元のファイルに書き出すか「保存(名前指定)」で別のファイルに書く。これらの意味をまとめておこう。なお、次にやるxpaintもまったく同様なので、一緒に含めた。

Mule	idraw	xpaint	機能
$\sim X^W$	「保存(名前指定)」	Save As	ファイル名を指定して書く
$\sim X^F$	「読み込み」	Open	ファイル名を指定して修正を開始
$\sim X^S$	「保存」	Save	もとのファイルに書き戻す
$\sim X^C$	「終了」	Quit	Muleやidrawを終わる

idraw形式を含め、PS形式のファイルをプリンタに打ち出す時にはlwpではなくlprというコマンドを使用すること。例えば次の通り。

```
% lpr -Plw25 test.ps
```

プリンタ指定は例によってなくてもよい(端末に張り付けたラベルに書かれたプリンタに出る)。

演習 4 これまでに自分が描いたPAD図または資料に載っていたのどれでもいいから、idrawで清書してみよ。なお、同じ形を沢山使う時には1つ描いた後それを選択した上「編集」メニューの「複製」を使ってコピーを作り、適当な位置に移動して使うのがよい。完成したらファイルに保存し、プリンタに打ち出してみよ。

3.6 xpaint 入門

さて、こんどはペイント系のお絵描きツールで、こっちは概念的にはずっと簡単である。そもそも、皆様が見ている画面というのはブラウン管の上で「様々な色の点の集まり」でできている。だから、それぞれの点に色々な色を割り当てれば任意の絵が作れますよ、というのがペイント系ツールの基本姿勢なわけである。ここではまたまた無料のソフトである xpaint を使用する。xpaint を起動するには Kterm の窓で

```
% xpaint &
```

と打ち込む。しばらくすると、図7のような窓が現われる。

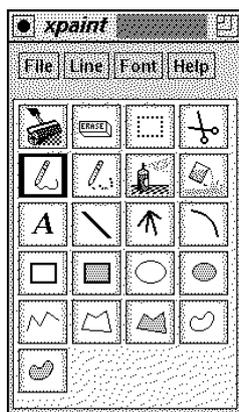


図 7: xpaint のツールパレット窓の様子

ちよつと、idraw のパレットに似ているでしょう？ ただし、xpaint では後で出てくる色パレットと区別するため「ツールパレット」と呼ぶことにしよう。xpaint の場合、絵の大きさを指定してから描き始めるので、まずツールパレットの窓が出て、ここからキャンバスの窓を開くようになっている。なお、ツールパレットの上の方にある「File」などは idraw と同じくプルダウンメニューになっている。

さて、以下では当面、特に述べない限りマウスの左ボタンを使用すること。キャンバスを開くには、「File」メニューの中から「New With Size」を選ぶ。すると、幅と高さと拡大率を打ち込む窓が出て来るので、ここでは練習用に幅と高さは 64、拡大率は 8 として (どれも既に値が入っているところをマウスでクリックして [DEL] キーでもとの値を消して新しい値を打ち込む)、最後に OK のボタンをつつくと図 8 のような四角い窓が現われる。

キャンバス窓の下には 2 つの「色パレット」があるが、これは左側が鉛筆や筆の色、右側がスクリーントーンの色と思えばよい。とりあえず両方のパレットとも模様ではない色 (で互いに違うもの) をクリックして選んでおくこと。

つぎに、ツールパレットから「鉛筆」マークを選び、キャンバスの上でどれかのボタンを押して動かす。すると、鉛筆の動いた後の点に色がついて残る。鉛筆の太さを変えたい時はパレットの「Line」メニューを使えばできる。

もっと太い線を描きたい場合にはパレットから「ローラ」を選んで使う。ローラの形はいく通りかに変えられるが、変えたい場合にはローラのパレットを右ボタンでつつくとメニューが出て、その中から「Select Brush」を選べばよい。ローラの隣にあるのは「消しゴム」で、これは間違っって書いてしまったところを白に戻すのに使う。消しゴムもローラと同様形が変えられる。「スプレー」はやってみれば分る通りスプレーペイントのような効果がある。「バケツ」は、閉じた形を作ってその中でバケツをこぼすと色が行き渡る。やってみて「失敗した!」と思ったときは、キャンバスの「Edit」メニューの「Undo」を選ぶと 1 回ぶんだけは操作を元に戻すことができる。

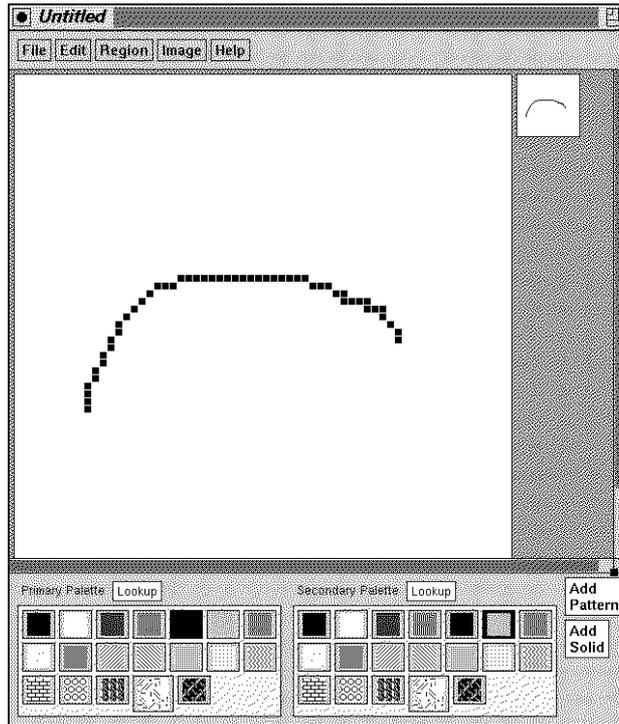


図 8: xpaint のキャンバス窓の様子

あと、直線や四角などの図形のパレットは idraw と同じように使えるが、ただしペイント系なので一度描いてしまったものは動かしたり消したりできない。なお、図形で中が灰色のものは、描いた後で中を色やパターンで塗る。その色やパターンはキャンバス窓の下右側のパレットで選ぶ。文字は文字が描けるが、ただし xpaint では漢字は使えない (どのみち文字はあまり重要でない)。

さて、キャンバス窓の色パレットについてだが、もっと別な色を使いたい場合には右下の「Add Solid」と書かれたボタンを押す。すると図 9 のような窓が現われるので、ここで円形のうち望む色のところを左ボタンでクリックするか、またはスライドラバーのところでもマウスの中ボタンを押して動かし、望む色が出た所で「OK」をクリックすればその色がパレットに追加されて利用可能になる。その他の機能については、「Help」ボタンなどを利用して調べて欲しい。

最後に、絵が完成して保存する時は例によって「File」メニューから「Save As」を並び、ファイル名を入力し [RET] する。ところで、絵を保存するとき様々なファイル形式が選べるが、ここでは「TIFF」(Tagged Image Format File) と呼ばれる形式を使っておく。標準では TIFF のボタンが選ばれているはず。もし別の形式にしたければファイル名を入れる前に使いたい形式のボタンをクリックして選んでおく。なお、TIFF のファイルは「test.tiff」のように最後を「.tiff」で終わらせておくとうい。

さて、これを何に使おうか? とりあえず、窓の背景模様にしても。それには

```
% xloadimage -onroot test.tiff
```

を実行すればよい。login したらいつでもこの絵を背景模様にした場合は、上の行 (%は不要!) を .xsession ファイルの適当な位置 (最後の行より上ならどこでも) に挿入すればよい。

演習 5 xpaint を使って、背景模様用の絵と、自分のトレードマークないしシンボルマークとを作成せよ。(面倒なら 1 つで両方兼ねてもよい。) そして、背景模様は .xsession を変更して login 時に自動設定させよ。なお、今回やった 64 × 64 ドットでは物足りないならもっと大きくしてもいいが、大きい絵ほど美しく作るのは大変なのでそのつもりで。大きくした場合拡大率は減らさないと画面に入り切らない。

A 本日の課題 **5A**

本日の課題は演習 1、演習 2、演習 3 を提出してください。あと演習 4 はできた人のみプリントアウトを提出してください。レポート番号は **5A** です。アンケートは次の通り。

- Q1. 変数の値の移り変わりについて、前から理解していましたか？ 今日ようやく理解しましたか？ まだだめですか？
- Q2. 文を四角で囲むのはできるようになりました？「;」を入れる場所についてはどうですか？
- Q3. idraw と xpaint についてそれぞれどう思いました？ また、両者の比較も述べてみてください。
- Q4. その他、感想、要望、質問があればどうぞ。

B 次回までの宿題

前回課したレポート **4B** が次回までの提出となっていますので、それをやってください。あと、演習 5 は特に提出を求めませんが、次回にその結果を使うのでやっておいてください。とって、今回練習で作ったものが自分のトレードマークにふさわしいと思う方はそれでも結構です。まだ提出してない方は **4B** の Q2. にその感想等も含めて頂ければ幸いです。

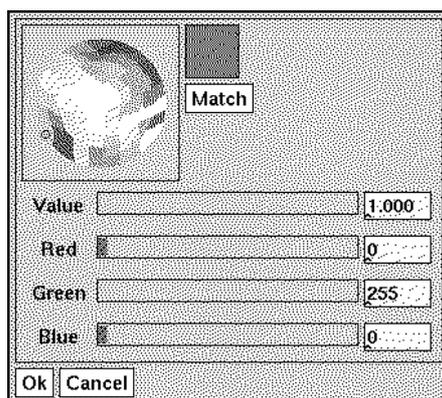


図 9: xpaint の色選択窓の様子